# International Journal on Advances in Internet Technology

2009 vol. 2 nr. 1

The *International Journal On Advances in Internet Technology* is Published by IARIA.
ISSN: 1942-2652
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

**International Journal On Advances in Internet Technology**
**Volume 2, Number 1, 2009**

### Communication Theory, QoS and Reliability

- Adrian Andronache, University of Luxembourg, Luxembourg
- Shingo Ata, Osaka City University, Japan
- Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania
- Michel Diaz, LAAS, France
- Michael Menth, University of Wuerzburg, Germany
- Michal Pioro, University of Warsaw, Poland
- Joel Rodriques, University of Beira Interior, Portugal
- Zary Segall, University of Maryland, USA

### Ubiquitous Systems and Technologies

- Sergey Balandin, Nokia, Finland
- Matthias Bohmer, Munster University of Applied Sciences, Germany
- David Esteban Ines, Nara Institute of Science and Technology, Japan
- Dominic Greenwood, Whitestein Technologies AG, Switzerland
- Arthur Herzog, Technische Universitat Darmstadt, Germany
- Malohat Ibrohimova, Delft University of Technology, The Netherlands
- Reinhard Klemm, Avaya Labs Research-Basking Ridge, USA
- Joseph A. Meloche, University of Wollongong, Australia
- Ali Miri, University of Ottawa, Canada
- Said Tazi, LAAS-CNRS, Universite Toulouse 1, France

### Systems and Network Communications

- Eugen Borcoci, University ‘Politechncia’ Bucharest, Romania
- Anne-Marie Bosneag, Ericsson Ireland Research Centre, Ireland
- Jan de Meer, smartspace®lab.eu GmbH, Germany
- Michel Diaz, LAAS, France
- Tarek El-Bawab, Jackson State University, USA
- Mario Freire, University of Beria Interior, Portugal / IEEE Portugal Chapter
- Sorin Georgescu, Ericsson Research - Montreal, Canada
- Huaqun Guo, Institute for Infocomm Research, A*STAR, Singapore
- Jong-Hyouk Lee, Sungkyunkwan University, Korea
- Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
- Zoubir Mammeri, IRIT - Paul Sabatier University - Toulouse, France
- Sjouke Mauw, University of Luxembourg, Luxembourg
- Reijo Savola, VTT, Finland

### Future Internet

- Thomas Michal Bohnert, SAP Research, Switzerland
- Fernando Boronat, Integrated Management Coastal Research Institute, Spain
- Chin-Chen Chang, Feng Chia University - Chiayi, Taiwan

- Bill Grosky, University of Michigan-Dearborn, USA
- Sethuraman (Panch) Panchanathan, Arizona State University - Tempe, USA
- Wei Qu, Siemens Medical Solutions - Hoffman Estates, USA
- Thomas C. Schmidt, University of Applied Sciences – Hamburg, Germany

**Challenges in Internet**

- Olivier Audouin, Alcatel-Lucent Bell Labs - Nozay, France
- Eugen Borcoci, University “Politehnica” Bucharest, Romania
- Evangelos Kranakis, Carleton University, Canada
- Shawn McKee, University of Michigan, USA
- Yong Man Ro, Information and Communication University - Daejon, South Korea
- Francis Rousseaux, IRCAM, France
- Zhichen Xu, Yahoo! Inc., USA

**Advanced P2P Systems**

- Nikos Antonopoulos, University of Surrey, UK
- Filip De Turck, Ghent University – IBBT, Belgium
- Anders Fongen, Norwegian Defence Research Establishment, Norway
- Stephen Jarvis, University of Warwick, UK
- Yevgeni Koucheryavy, Tampere University of Technology, Finland
- Maozhen Li, Brunel University, UK
- Jorge Sa Silva, University of Coimbra, Portugal
- Lisandro Zambenedetti Granville, Federal University of Rio Grande do Sul, Brazil

**International Journal On Advances in Internet Technology**
**Volume 2, Number 1, 2009**

## Foreword

The first 2009 number of the International Journal On Advances in Internet Technology compiles a set of papers with major enhancements based on previously awarded publications. It brings together a set of articles that provide a good balance between experimental and more theoretical aspects of various issues related to internet technology. For this issue, seventeen contributions have been selected.

The first contribution by Thomas Dreibholz and Erwin P. Rathgeb go through an overview and evaluation of server redundancy and failover mechanisms in the Reliable Server Pooling recently defined by the IETF. The evaluation of the approach is done with respect to load balancing strategies.

In the second article, Colin Atkinson et al. touch on the issue of context-sensitive service discovery which at the same time ensures privacy. Such a system presents challenges for both users and developers. With the same focus on the user and the principal actor, Netzahualcoyotl Ornelas et al. present in the following article an event based knowledge inference for user centric information systems. The proposal is to improve user centricity through a persistent data model.

The next two contributions deal with queuing aspects. Joris Walraevens et al. provide a performance analysis of a priority queue with session-based arrivals and its application to an e-commerce server. Yassine Ariba et al offer a framework for congestion control at a single router using Active Queue management.

The fifth article by Ashok-Kumar Chandra-Sekaran et al. deals with patient localization and air temperature monitoring using ZigBee. A location aware wireless sensor network can be used in cases of mass emergency in order to manage victims overflow, triage, etc.

Michael Collins et al. introduce a lightweight secure architecture for wireless sensor networks. By their nature, wireless sensor nodes can end up in situations where malicious users have an interest in getting to the data. With the proposed method, security can be achieved and aberrant sensor nodes can be isolated.

Mathilde Benveniste proposes an improvement to the MAC layer protocol in wireless mesh to deal specifically with inherent QoS degradation when traditional MAC design is used. The proposals reduce latency for mesh traffic and improve co-existence with neighboring WLANs.

Additional research at the MAC layer is presented by Radosveta Sokullu et al. GTS attacks are simulated under different scenarios. Results are collected and interpreted both from the attacker’s perspective and the victim’s point of view.

In the ninth article, Jyrki T.J. Penttinen describes the SFN gain related items as a part of the detailed radio DVB-H network planning. The emphasis is put to the effect of DVB-H parameter settings on the error levels caused by the over-sized Single Frequency Network (SFN) area.

Frank Bohdanowicz et al. investigate metric-based routing with topology investigation. As the network state change, routing algorithms have to adapt themselves to the new situation. The end objective is improvement in convergence, stability, and scalability.

Bart Braem et al. present and offer improvements for multi-hop body sensor networks. While current research has mostly focused on single hop networks, multi-hop networks offer advantages.

Alex Vallejo et al. propose the use of a genetic hybrid algorithm for traffic engineering in NGNs. As NGN architecture specifies support for QoS, it is a good idea to tie traffic engineering to this concept. Implementation and results obtained through the proposed solution are presented.

In the following article, Radim Burget et al. describe the integration and aggregation of IPTV mobile devices. Complexity grows as the content consumers are offered the option to interact with the content providers. Hierarchical aggregation with internet coordinate systems is proposed.

The last three articles of the journal deal with aspect of mobility and the effects on network state. Elin Sundby Boysen and Torleiv Maseng propose a handover scheme by extending SIP using a back-to-back user agent. Yao H. Ho et al. look at vehicular networks and expand the 3CE proposal meant at punishing misbehaving users making use of such a network. In this context, misbehaving is limited to the collaboration avoidance among mobile users. The final article dealing with mobile networks by Tom Leclerc et al. deals with stabilizing cluster structures in mobile networks. To this end, two protocols are compared: OLSR and WCPD. The conclusion is that OLSR is a better choice.

We thank the reviewers who spent significant time in reading and providing constructive comments. We hope that you will enjoy the contents of this journal find it useful for discovering more about the challenges and approaches for internet technology, and that you will be inspired to contribute to IARIA's conferences that include topics relevant to this journal.

*Andreas J Kassler, Editor-in-Chief*

**International Journal On Advances in Internet Technology**
**Volume 2, Number 1, 2009**

## CONTENTS

# Overview and Evaluation of the Server Redundancy and Session Failover Mechanisms in the Reliable Server Pooling Framework*

Thomas Dreibholz, Erwin P. Rathgeb
University of Duisburg-Essen, Institute for Experimental Mathematics
Ellernstrasse 29, 45326 Essen, Germany
{dreibh,rathgeb}@iem.uni-due.de

## Abstract

*The number of availability-critical Internet applications is steadily increasing. To support the development and operation of such applications, the IETF has recently defined a new standard for a common server redundancy and session failover framework: Reliable Server Pooling (RSerPool). The basic ideas of the RSerPool framework are not entirely new, but their combination into a single, resource-efficient and unified architecture is. Service availability achieved by the redundancy of servers directly leads to the issues of load distribution and load balancing, which are both important for the performance of RSerPool systems. Therefore, it is crucial to evaluate the performance of such systems with respect to the load balancing strategy required by the service application.*

*In this article – which is an extended version of our paper [1] – we first present an overview of the RSerPool architecture with a focus on the component failure detection and handling mechanisms. We will also shortly introduce the underlying SCTP protocol and its link redundancy features. After that, we will present a quantitative, application-independent performance analysis of the failure detection and session failover mechanisms provided by RSerPool, with respect to important adaptive and non-adaptive load balancing strategies.*

***Keywords:*** *Reliable Server Pooling, Service Availability, Redundancy, Session Failover, Server Selection*

## 1 Introduction and Related Work

Service availability is becoming increasingly important in today's Internet. But – in contrast to the telecommunications world, where availability is ensured by redundant links and devices [2] – there had not been any generic, standardized approaches for the availability of Internet-based services. Each application had to realize its own solution and therefore to re-invent the wheel. This deficiency – once more arisen for the availability of SS7 (Signalling System No. 7) services over IP networks – had been the initial motivation for the IETF RSerPool WG to define the Reliable Server Pooling (RSerPool) framework. The basic ideas of RSerPool are not entirely new (see [3, 4]), but their combination into one application-independent framework is.

Server redundancy [5] leads to the issues of load distribution and load balancing [6], which are also covered by RSerPool [7, 8]. But unlike solutions in the area of GRID and high-performance computing [9], the RSerPool architecture is intended to be lightweight. That is, RSerPool may only introduce a small computation and memory overhead for the management of pools and sessions [8, 10–12]. In particular, this means the limitation to a single administrative domain and only taking care of pool and session management – but not for higher-level tasks like data synchronization, locking and user management. These tasks are considered to be application-specific. On the other hand, these restrictions allow for RSerPool components to be situated on low-end embedded devices like routers or telecommunications equipment.

While there have already been a number of publications on applicability and performance of RSerPool (see e.g. [7, 13–17]), a generic, application-independent performance analysis of its failover handling capabilities was still missing. In particular, it is necessary to evaluate the different RSerPool mechanisms for session monitoring, server maintenance and failover support – as well as the corresponding system parameters – in order to show how to achieve a good system performance at a reasonably low maintenance overhead. The goal of our work is

*Parts of this work have been funded by the German Research Foundation (Deutsche Forschungsgemeinschaft).

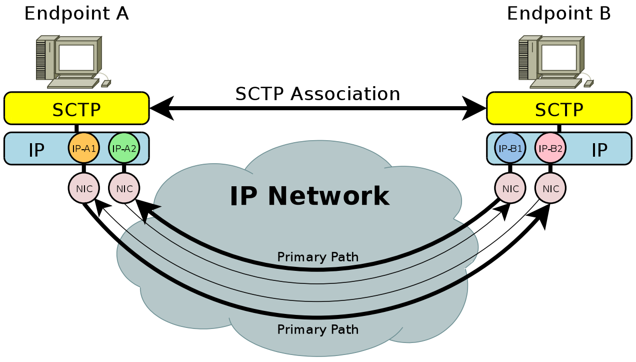


**Figure 1. A Multi-Homed SCTP Association**

an application-independent quantitative characterization of RSerPool systems, as well as a generic sensitivity analysis on changes of workload and system parameters. Especially, we intend to identify the critical parameter ranges in order to provide guidelines for design and configuration of efficient RSerPool-based services.

This article is an extended version of our conference paper [1]. It contains a broader overview of the redundancy mechanisms provided by RSerPool and the underlying SCTP protocol as well as a more detailed analysis of the session failover mechanisms.

The document is structured as follows: in Section 2, we present an overview of RSerPool and the underlying SCTP protocol. A generic quantification of RSerPool systems is introduced in Section 3. Using the RSPSIM simulation model and setup described in Section 4, we evaluate the server failure detection and handling features of RSerPool in Section 5.

## 2 The RSerPool Architecture

RSerPool is based on the SCTP transport protocol. Therefore, it is necessary to shortly introduce this protocol and its link failure handling features first.

### 2.1 Data Transport and Motivation

The Stream Control Transmission Protocol (SCTP, see [18,19]) is a connection-oriented, general-purpose, unicast transport protocol which provides reliable transport of user messages. An SCTP connection is denoted as *association*. Each SCTP endpoint can use multiple IPv4 and/or IPv6 addresses to provide network fault tolerance. The addresses used by the endpoints are negotiated during association setup, a later update is possible using dynamic address reconfiguration (Add-IP, see [20]). Add-IP can also be applied to allow for endpoint mobility. This link redundancy feature is called *multi-homing* [21,22] and illustrated in Figure 1. An endpoint sees each remote address as unidirectional *path*. In each direction, one of the paths is selected as so-called *primary path*. This path is used for the transport of user data. The other paths are backup paths, which are used for retransmissions only. Upon failure of the primary path, SCTP selects a new primary path from the set of possible backup paths. That is, as long as there is at least one usable path in each direction, the association remains usable despite of link failures. However, multi-homing cannot protect a service against endpoint failures. To cope with this problem, the IETF has defined the RSerPool architecture on top of SCTP.

### 2.2 Architecture

Figure 2 illustrates the RSerPool architecture, as defined in [23]. It consists of three major component classes: servers of a pool are called *pool elements* (PE). Each pool is identified by a unique *pool handle* (PH) in the handlespace, i.e. the set of all pools. The handlespace is managed by *pool registrars* (PR), which are also shortly denoted as *registrars*. PRs of an operation scope synchronize their view of the handlespace using the Endpoint haNdlespace Redundancy Protocol (ENRP [24]), transported via SCTP. An operation scope has a limited range, e.g. a company or organization; RSerPool does not intend to scale to the whole Internet. This restriction results in a very small pool management overhead (see also [8,10,25]), which allows to host a PR service on routers or embedded systems. Nevertheless, it is assumed that PEs can be distributed worldwide, for their service to survive localized disasters [26,27].

A client is called *pool user* (PU) in RSerPool terminology. To use the service of a pool given by its PH, a PU requests a PE selection from an arbitrary PR of the operation scope, using the Aggregate Server Access Protocol (ASAP [28,29]). The PR selects the requested list of PE identities using a pool-specific selection rule, called *pool policy*. Adaptive and non-adaptive pool policies are defined in [30]; relevant for this article are the non-adaptive policies Round Robin and Random and the adaptive policy Least Used. Least Used selects the least-used PE, according to up-to-date load information. The actual definition of *load* is application-specific: for each pool the corresponding application has to specify the actual meaning of *load* (e.g. CPU utilization or storage space usage) and present it to RSerPool in form of a numeric value. Among multiple least-loaded PEs, Least Used applies Round Robin selection (see also [8]). Some more policies are evaluated in [26,27,31].

A PE can register into a pool at an arbitrary PR of the operation scope, again using ASAP transported via SCTP. The

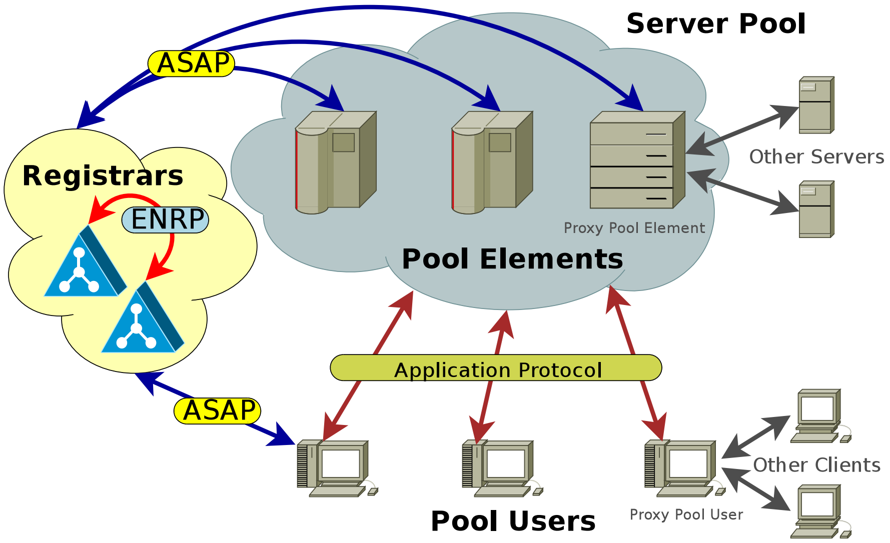


**Figure 2. The RSerPool Architecture**

chosen PR becomes the *Home PR* (PR-H) of the PE and is also responsible for monitoring the PE's health by *endpoint keep-alive* messages. If not acknowledged, the PE is assumed to be dead and removed from the handlespace. Furthermore, PUs may report unreachable PEs; if the threshold MAX-BAD-PE-REPORT of such reports is reached, a PR may also remove the corresponding PE. The PE failure detection mechanism of a PU is application-specific.

Proxy Pool Elements (PPE) allow for the usage of non-RSerPool servers in RSerPool-based environments. Respectively, non-RSerPool clients can use Proxy Pool Users (PPU) to access RSerPool-based services.

## 2.3 Protocol Stack

The protocol stack of the three RSerPool components is illustrated in Figure 3: a PR provides ENRP and ASAP services to PRs and PEs/PUs respectively. But between PU and PE, ASAP provides a Session Layer protocol in the OSI model[1]. From the perspective of the Application Layer, the PU side establishes a *session* with a pool. ASAP takes care of selecting a PE of the pool, initiating and maintaining the underlying transport connection and triggering a failover procedure when the PE becomes unavailable.

## 2.4 Session Failover Handling

While RSerPool allows the usage of arbitrary mechanisms to realize the application-specific resumption of an interrupted session on a new server, it contains only one built-in mechanism: Client-Based State Sharing. This mechanism has been proposed by us in our paper [32] and it is now part of the ASAP standard [28]. Using this feature, which is illustrated in Figure 4, a PE can send its current session state to the PU in form of a state cookie. The PU stores the latest state cookie and provides it to a new PE upon failover. Then, the new PE simply restores the state described by the cookie. For RSerPool itself, the cookie is opaque, i.e. only the PE-side application has to know about its structure and contents. The PU can simply handle it as a vector of bytes (However, as we will describe in Subsubsection 5.2.2, a more complex handling concept may improve application efficiency). Cryptographic methods can ensure the integrity, authenticity and confidentiality of the state information. In the usual case, this can be realized easily by using a pool key which is known by all PEs (i.e. a "shared secret").

## 2.5 Applications

The initial motivation of RSerPool has been to ensure the availability of SS7 (Signalling System No. 7, see [33]) services over IP networks. However, since component availability is a very common problem for computer network applications, RSerPool has been designed for application independence. The current research on applicability and performance of RSerPool includes application scenarios (described in detail by [7, Section 3.6]) like VoIP with SIP [17], SCTP-based mobility [34], web server pools [7, Section 3.6], e-commerce systems [32], video on demand [15], battlefield networks [16], IP Flow

[1] ASAP [28] is the first IETF standard for a Session Layer protocol.

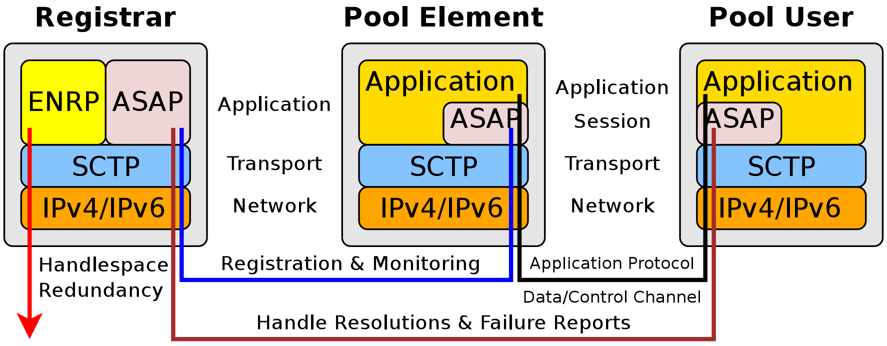


**Figure 3. The RSerPool Protocol Stack**

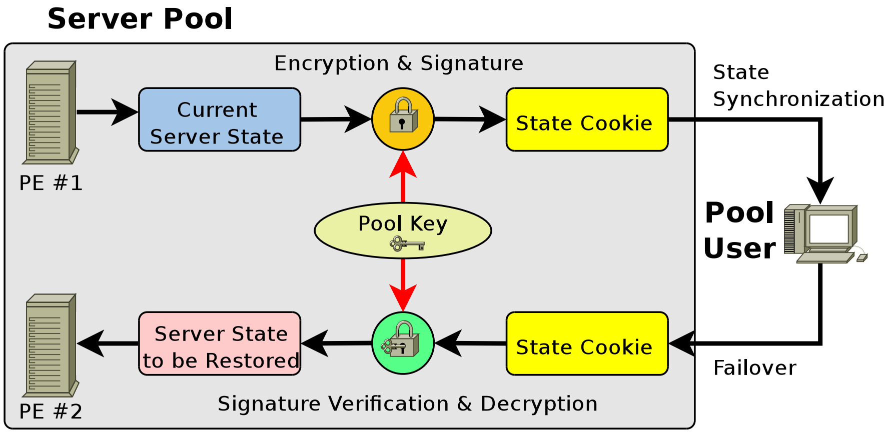


**Figure 4. Client-Based State Sharing**

Information Export (IPFIX) [35] and workload distribution [7, 13, 14, 26, 27, 31, 36–39].

Since RSerPool has just reached a major milestone by the publication of its basic protocol documents as RFCs in September 2008, a wide deployment of RSerPool-based applications is expected for the future [40].

## 3 Quantifying a RSerPool System

The service provider side of a RSerPool-based service consists of a pool of PEs, using a certain server selection policy. Each PE has a request handling *capacity*, which we define in the abstract unit of calculations per second. Depending on the application, an arbitrary view of capacity can be mapped to this definition, e.g. CPU cycles or memory usage. Each request consumes a certain number of calculations, we call this number the *request size*. A PE can handle multiple requests simultaneously, in a processor sharing mode (multi-tasking principle).

On the service user side, there is a set of PUs. The number of PUs can be given by the ratio between PUs and PEs (PU:PE ratio), which defines the parallelism of the request handling. Each PU generates a new request in an interval denoted as *request interval*. The requests are queued and sequentially assigned to PEs.

Clearly, the user-side performance metric is the handling speed – which should be as fast as possible. The total delay for handling a request $d_{\text{handling}}$ is defined as the sum of queuing delay, startup delay (dequeuing until reception of acceptance acknowledgement) and processing time (acceptance until finish) as illustrated in Figure 5. The *handling speed* (in calculations/s) is defined as:

$$\text{handlingSpeed} = \frac{\text{requestSize}}{d_{\text{handling}}}.$$

For convenience reasons, the handling speed can be represented in % of the average PE capacity. Clearly, in case of a PE failure, all work between the last checkpoint and the failure is lost and has to be re-processed later. A failure has to be detected by an application-specific mechanism (e.g.

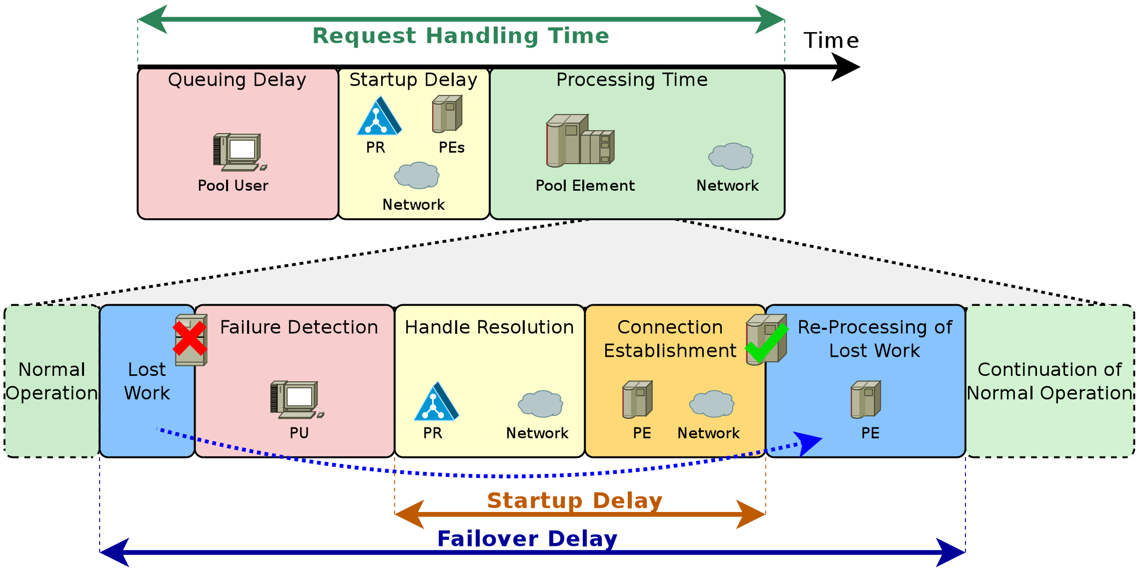


**Figure 5. Request Handling Delays**

keep-alives) and a new PE has to be chosen and contacted for session resumption.

Using the definitions above, the system utilization – which is the provider-side performance metric – can be calculated:

$$\text{systemUtilization} = \text{puToPERatio} * \frac{\frac{\text{requestSize}}{\text{requestInterval}}}{\text{peCapacity}}$$

In practice, a well-designed RSerPool system is dimensioned for a certain *target system utilization*. [7, 14] provide a detailed discussion of this subject.

## 4 The Simulation Scenario Setup

For our performance analysis, we have developed our simulation model RSPSIM [7, 13] using OMNeT++ [41] and the SimProcTC [36] tool-chain, containing full implementations of the protocols ASAP [28, 29] and ENRP [24], a PR module and PE and PU modules modelling the request handling scenario defined in Section 3. The scenario setup is shown in Figure 6: all components are interconnected by a switch. Network delay is introduced by link latency only. Component latencies are neglected, since they are not significant (as shown in [8]). We further assume sufficient network bandwidth for pool management and applications. Since an operation scope is limited to a single administrative domain, QoS mechanisms may be applied.

Unless otherwise specified, the used target system utilization is 60%, i.e. there is sufficient over-capacity to cope with PE failures. For the Least Used policy, we define *load* as the current number of simultaneously handled requests. The capacity of a PE is $10^6$ calculations/s, the average request size is $10^7$ calculations. Both parameters use negative exponential distribution – for a generic parameter sensitivity analysis being independent of a particular application [14]. We use 10 PEs and 100 PUs, i.e. the PU:PE ratio is 10. This is a non-critical setting for the examined policies, as shown in [14].

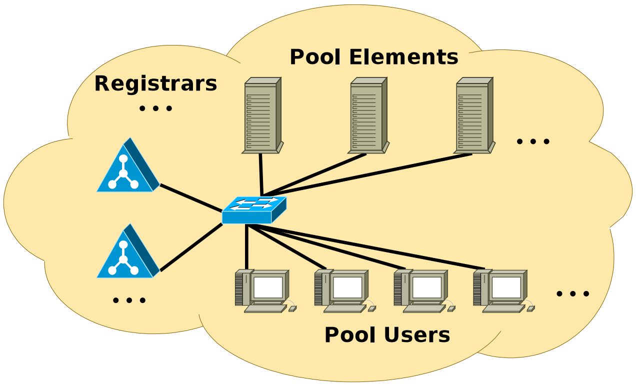


**Figure 6. The Simulation Setup**

Session health monitoring is performed by the PUs using keep-alive messages in a *session keep-alive interval* of 1s, to be acknowledged by the PE within a *session keep-alive timeout* of 1s (parameters evaluated in Subsubsection 5.1.3). Upon a simulated failure, the PE simply disappears and reappears immediately under a new transport address, i.e. the overall pool capacity remains constant. Client-Based State Sharing is applied for failovers; the default cookie interval is 10% of the request size (i.e. $10^6$ calculations; parameter is evaluated in Subsubsection 5.2.2). Work not being protected by a checkpoint has to be re-processed on a new PE upon session failover.

In this article, we neglect PR failures and therefore use a single PR only (for details on PR failures, see [11, 12]). All failure reports by PUs are ignored (i.e. MAX-BAD-PE-REPORT=$\infty$) and the endpoint keep-alive interval and timeout are 1s (parameters evaluated in Subsubsection 5.1.4). The inter-component network delay is 10ms (realistic for connections within a limited geographical area, see [26]). The simulated real-time is 60 minutes; each simulation is repeated 24 times with different seeds to achieve statistical accuracy. The post-processing of results, i.e. computation of 95% confidence intervals and plotting, has been performed using GNU R [42].

# 5 Results

As shown in Figure 5, two components contribute to the failure handling time:

1. The failure detection delay and
2. The failure handling delay (i.e. re-processing effort for lost work).

Therefore, we will examine the failure detection procedures in the following Subsection 5.1 and failover handling procedures in Subsection 5.2.

## 5.1 Failure Detection

### 5.1.1 Dynamic Pools

In the ideal case, a PE informs its PU of an oncoming shutdown, sets a checkpoint for the session state (e.g. by a state cookie [1, 32]) and performs a de-registration at a PR. Then, no re-processing effort is necessary. This situation, as shown for the handling speed on the right-hand side of Figure 7, becomes critical only for a very low PE MTBF (Mean Time Between Failure; here: given in average request handling times) in combination with network delay (i.e. the failover to a new PE is not for free). As being observable for failure-free scenarios (see [14]), the best performance is again provided by the adaptive Least Used policy, due to PE load state knowledge. However, the Round Robin performance converges to the Random result for a low MTBF: in this case, there is no stable list of PEs to select from in turn – the selection choices become more and more random.

The results for the system utilization, shown on the left-hand side of Figure 7, are very similar to the handling speed behaviour: except for extremely low MTBF settings, the utilization remains at the expected target system utilization of 60%.

An approach to utilize the failover capabilities of RSerPool-based systems for improving the server selection performance is described by [38, 39]: “Reject and Retry”. When a PE is highly loaded due to non-optimal server selection (e.g. due to temporary capacity changes), it may simply reject a new request. The failover capabilities of RSerPool then retry at another PE. Since the failover handling of RSerPool is quite efficient, this can lead – despite of the failure handling – to a significant performance improvement in certain scenarios.

### 5.1.2 De-Registrations and Failures

In real scenarios, PEs may fail without warning. That is, a PU has to detect the failure of its PE in order to trigger a failover. For the simulated application, this detection mechanism has been realized by keep-alive messages. The general effects of a decreasing amount of “clean” shutdowns (i.e. the PE simply disappears) are presented in Figure 8. Clearly, the less “clean” shutdowns, the higher the re-processing effort for lost work: this leads to a higher utilization and lower handling speed. As expected, this effect is smallest for Least Used (due to superior load balancing) and highest for Random. There is almost no reaction of the utilization to an increased session keep-alive interval (given in average request handling times): a PU does not utilize resources while it waits for a timeout. However, the impact on the handling speed is significant: waiting increases the failover handling time and leads to a lower handling speed. For that reason, a tight session health monitoring interval is crucial for the system performance.

### 5.1.3 Session Health Monitoring

To emphasize the impact of the session health monitoring granularity, Figure 9 shows the utilization (left-hand side) and handling speed (right-hand side) results for varying this parameter in combination with the endpoint keep-alive interval, for a target utilization of 40% (higher settings become critical too quickly). The utilization results have been omitted, since they are obvious. Again, the performance results for varying the policy and session keep-alive interval reflect the importance of a quick failure detection for the handling speed – regardless of the policy used. However, the policy has a significant impact on the utilization: as shown in [14], the selection quality of Least Used is better than Round Robin, and Round Robin is better than Random. A better selection quality results in better handling

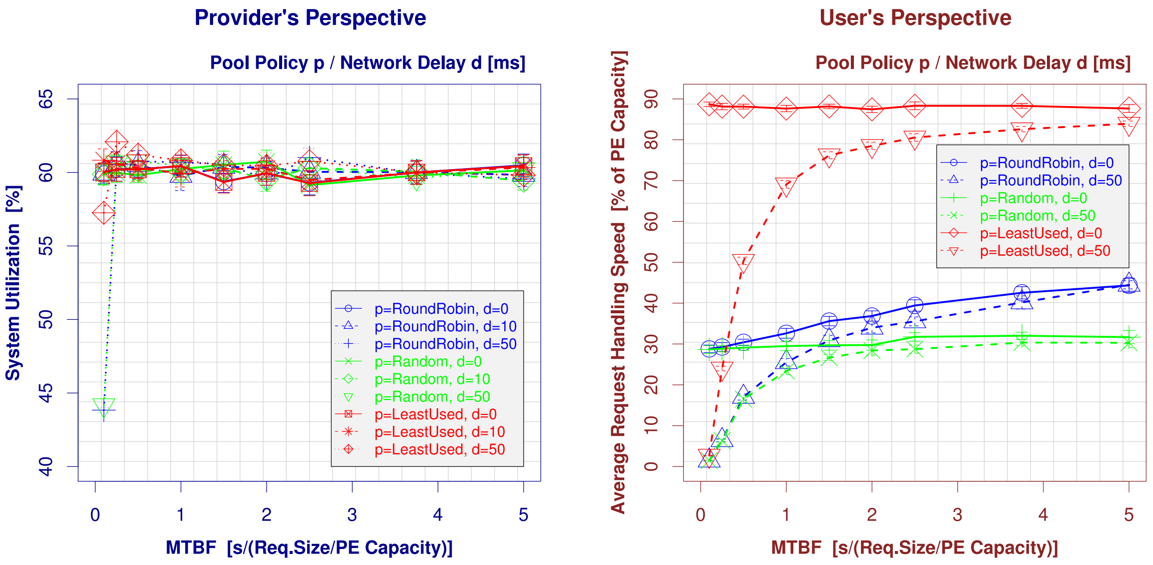


**Figure 7. The Performance for Dynamic Pools**

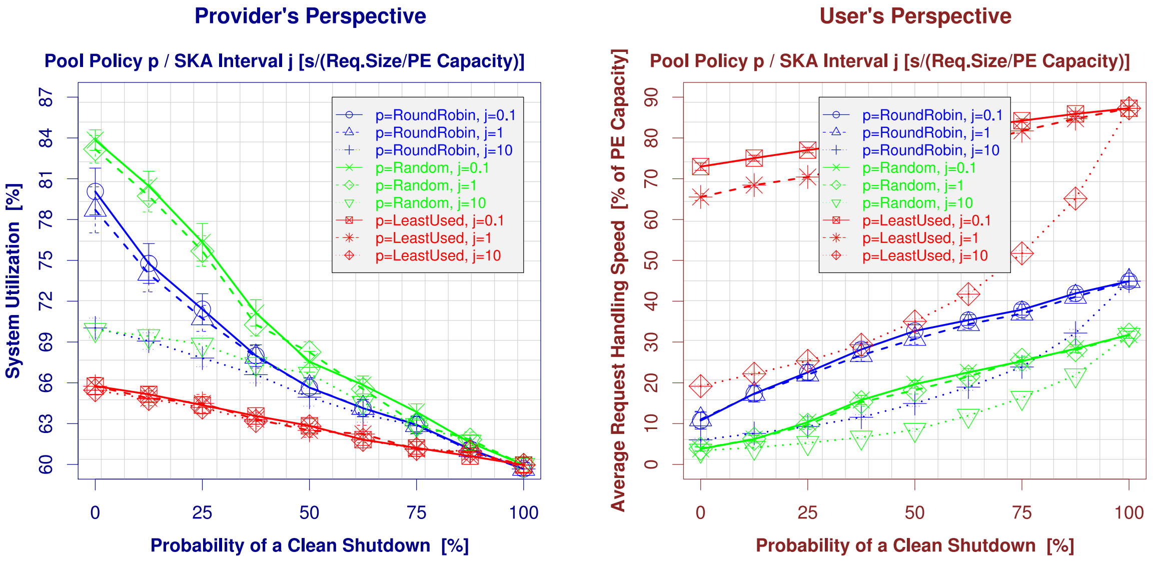


**Figure 8. The Impact of Clean Shutdowns**

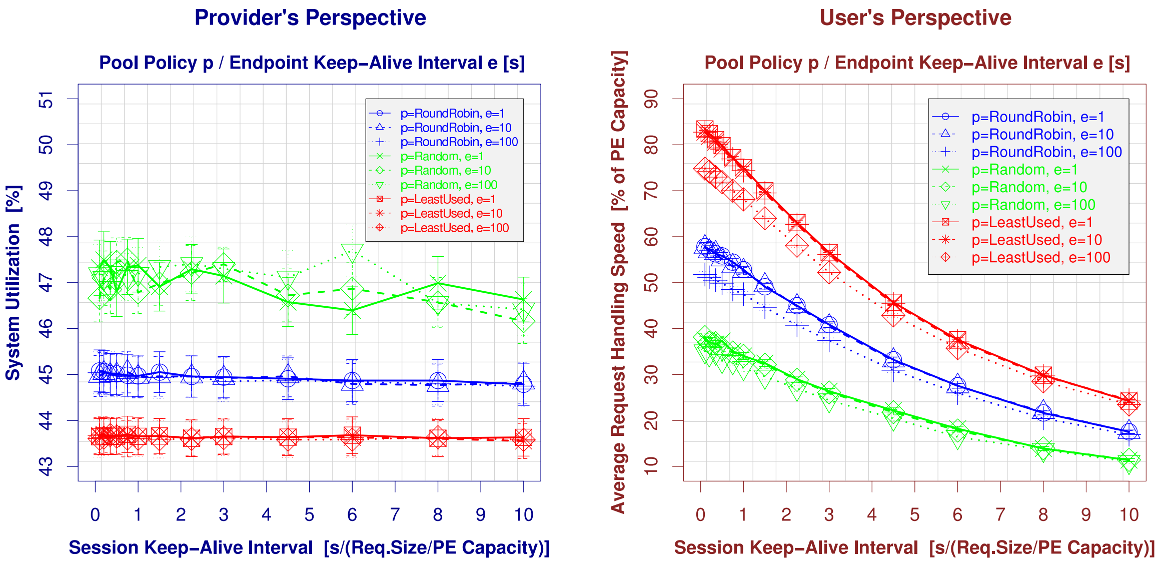


**Figure 9. The Impact of Session Monitoring**

speeds. Since the handling speed for Random is lowest, the amount of lost work is highest here. This results in an increased amount of re-processing, which can be observed by the higher utilization: about 47% for Random, about 45% for Round Robin but only about 43.5% for Least Used at a target system utilization of 40%.

It has to be noted that a small monitoring granularity does not necessarily increase overhead: e.g. a PU requesting transactions by a PE could simply set a transaction timeout. In this case, session monitoring even comes for free. Unlike the session monitoring, the impact of the PR's endpoint keep-alive interval is quite small here: even a difference of two orders of magnitude only results in at most a performance difference of 10%.

### 5.1.4 Server Health Monitoring

The endpoint keep-alive interval gains increasing importance when the request size becomes small. Then, the startup delay becomes significant, as illustrated in Figure 5. In order to show the general effects of the PE health monitoring based on endpoint keep-alives, Figure 10 presents the utilization (left-hand side) and handling speed results (right-hand side) for a request size:PE capacity ratio of 1 and a target system utilization of 25% (otherwise, the system becomes unstable too quickly).

While the policy ranking remains as expected, it is clearly observable that the higher the endpoint keep-alive interval and the smaller the MTBF, the more probable is the selection of an already failed PE. That is, the PU has to detect the failure of its PE by itself (by session monitoring, see Subsubsection 5.1.3) and trigger a new PE selection. The result is a significantly reduced request handling speed. Furthermore, for a very low MTBF, the utilization decreases: instead of letting PEs re-process lost work, the PUs spend more and more time on waiting for request timeouts. For these reasons, a PR-based PE health monitoring becomes crucial for such scenarios. But this monitoring results in network overhead for the keep-alives and acknowledgements as well as for the SCTP transport. So, is there a possibility to reduce this overhead?

The mechanism for overhead reduction is to utilize the session health monitoring (which is necessary anyway, as shown in Subsubsection 5.1.3) for PE monitoring by letting PUs report the failure of PEs. If MAX-BAD-PE-REPORT failure reports have been received, the PE is removed from the handlespace. The effectiveness of this mechanism is demonstrated by the results in Figure 11 (for the same parameters as above): even if the endpoint keep-alive overhead is reduced to $\frac{1}{30}$th, there is only a handling speed decrease of about 4% for MAX-BAD-PE-REPORT=1. The higher MAX-BAD-PE-REPORT, the more important the endpoint keep-alive granularity.

However, while the failure report mechanism is highly effective for all three policies, care has to be taken for security: trusting in failure reports gives PUs the power to impeach PEs! Approaches for improving the security of RSerPool systems against Denial of Service attacks are presented by [43–46].

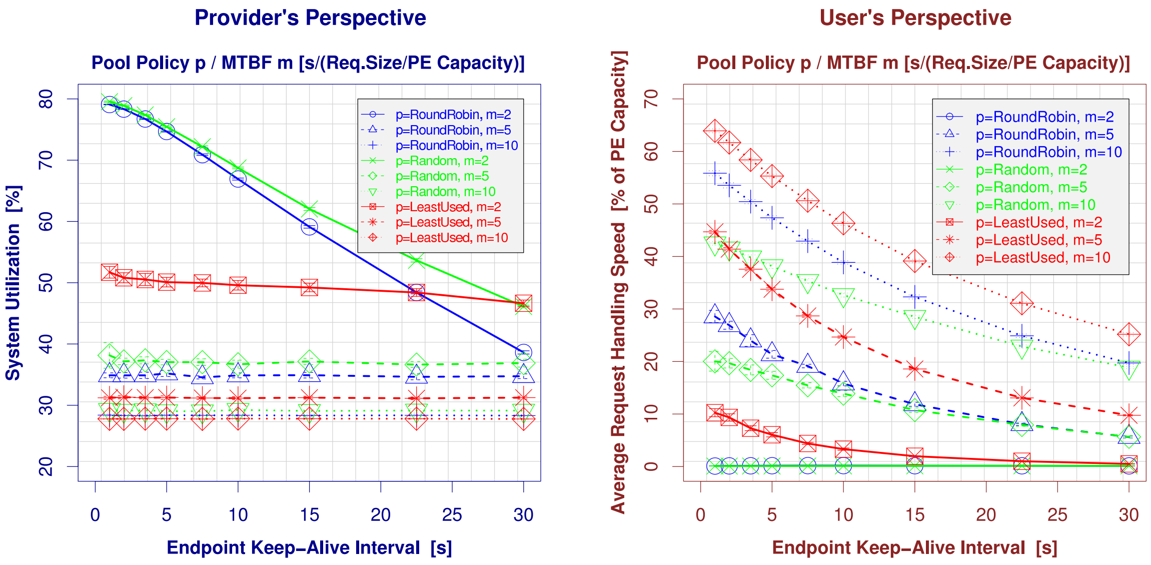


**Figure 10. The Impact of Pool Element Health Monitoring**

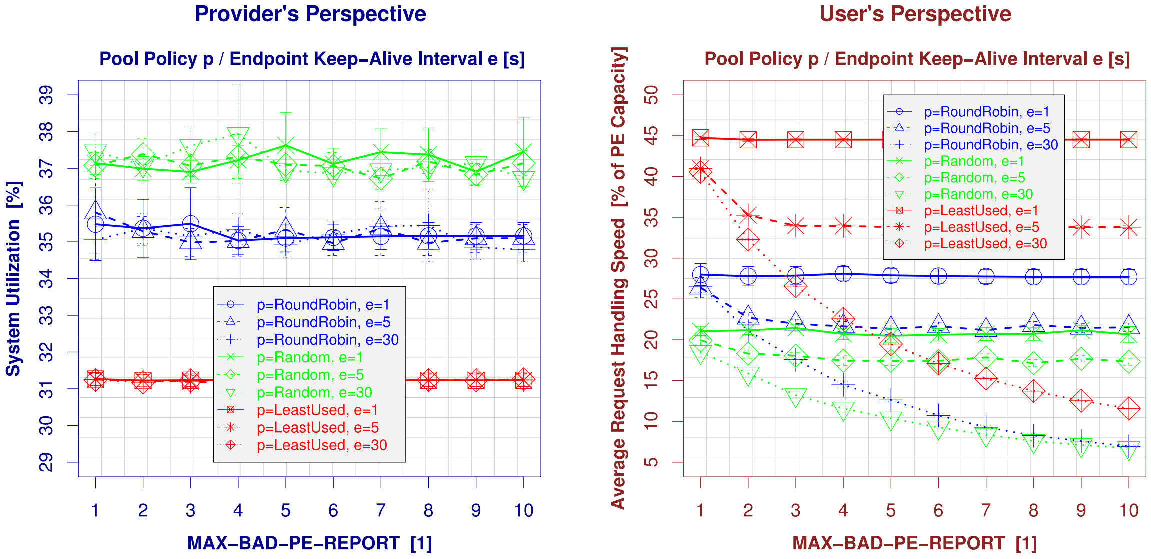


**Figure 11. Utilizing Failure Reports**

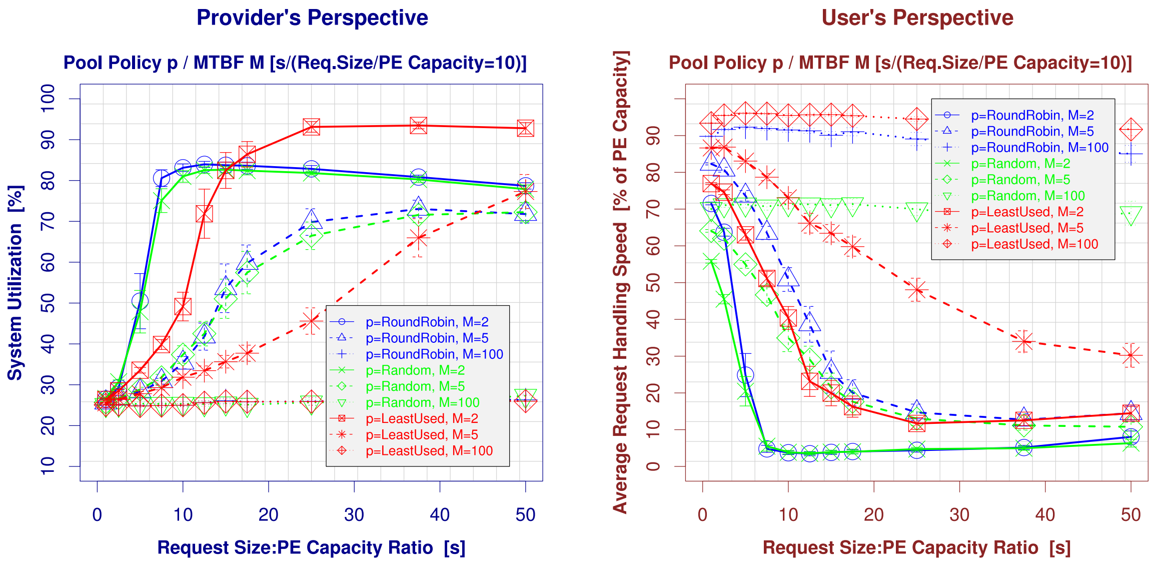


**Figure 12. Using “Abort and Restart”**

## 5.2 Failover Mechanisms

### 5.2.1 “Abort and Restart”

After detecting a PE failure and contacting a new server, the session state has to be restored for the re-processing of lost work and the application resumption. The simplest mechanism is “Abort and Restart” [7]: the session is restarted from scratch. The essential effects of applying this mechanism are presented in Figure 12: As expected, the impact of using “Abort and Restart” on the average system utilization (left-hand side of Figure 12) is small, as long as the PEs remain sufficiently available: for a MTBF of 100 times the time required to exclusively process a request having a request size:PE capacity of 10, the utilization increment is almost invisible. Furthermore, the decrement of the handling speed (right-hand side of Figure 12) is also small. Clearly, the rare necessity to restart a session has no significant performance impact.

However, for a sufficiently small MTBF, the results change: at a MTBF of 5, a significant utilization rise – as well as a handling speed decrease – can be observed for Round Robin and Random if the request size:PE capacity ratio $s$ is increased. The effect on Least Used is smaller: as expected from the dynamic pool performance results of Subsubsection 5.1.1, this policy is able to provide a better processing speed due to superior request load distribution. That is, the probability for a request of a fixed size to be affected by PE failures is smaller if using Least Used instead of Round Robin and Random. Note that the utilization of Least Used almost reaches 95% for a MTBF of 2 and larger request size:PE capacity ratios $s$, due to its better load distribution capabilities. In the same situation – i.e. high overload, of course – the Round Robin and Random policies only achieve an utilization of less than 85%.

In summary, “Abort and Restart” is fairly simple and useful in case of short transactions on sufficiently available PEs. But in all other cases, it is instead useful to define checkpoints to allow for session resumption from the latest checkpoint.

### 5.2.2 Client-Based State Sharing

Client-Based State Sharing (see Subsection 2.4) using state cookies offers a simple but effective solution for the state transfer. It is applicable as long as the state information remains sufficiently small[2]. To show the general effects of using this mechanism, Figure 13 presents the performance results for varying the cookie interval CookieMaxCalcs (given as the ratio between the number of calculations and the average request size) for different policy and PE MTBF settings.

The larger the cookie transmission interval and the smaller the PE MTBF, the lower the system performance: work (i.e. processed calculations) not being conserved by the cookie is lost. This results in an increased utilization, due to re-processing effort. Furthermore, this additional workload leads to a reduction of the request handling speed. Clearly, the better a policy’s load balancing capabilities, the better the system utilization and request handling speed (Least Used better than Round Robin better than Random, as for failure-free scenarios [7, 14]).

[2] The maximum state cookie size is less than 64 KiB [28].

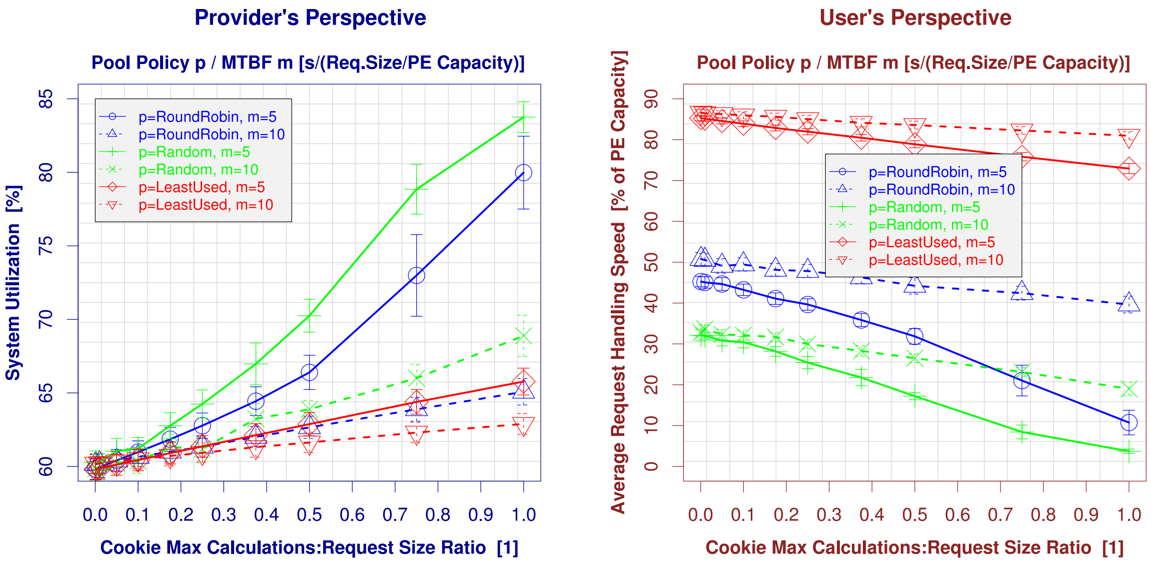


**Figure 13. Using Client-Based State Sharing for the Session Failover**

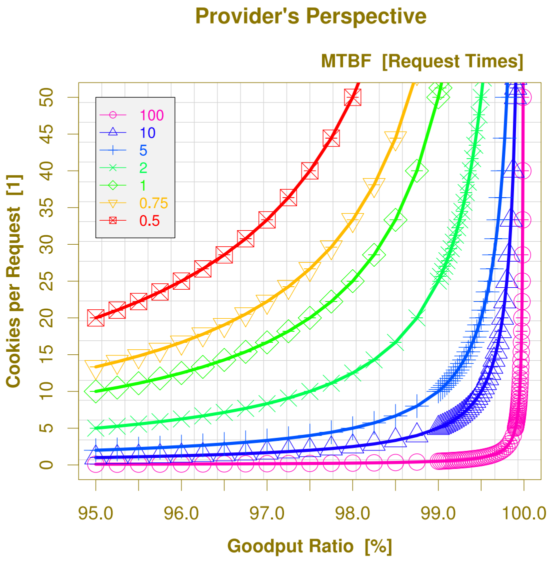


**Figure 14. The Number of Cookies**

In order to configure an appropriate cookie interval, the overhead of the state cookie transport has to be taken into account. The average loss of calculations per failure can be estimated as the half cookie interval $\mathrm{CookieMaxCalcs}$ (in calculations, as multiple of the average request size):

$$\mathrm{AvgLoss} = \frac{\mathrm{CookieMaxCalcs}}{2}.$$

Given an approximation of the PE MTBF (in average request handling times) and $\mathrm{AvgCap}$ the average PE capacity, the goodput ratio can be estimated as follows:

$$\mathrm{Goodput} = \frac{(\mathrm{MTBF} * \mathrm{AvgCap}) - \mathrm{AvgLoss}}{\mathrm{MTBF} * \mathrm{AvgCap}}.$$

Then, the cookie interval $\mathrm{CookieMaxCalcs}$ for a given goodput ratio is:

$$\mathrm{CookieMaxCalcs} = -2 * \mathrm{MTBF} * \mathrm{AvgCap} * (\mathrm{Goodput} - 1). \qquad (1)$$

Figure 14 illustrates the cookies per request (i.e. $\frac{1}{\mathrm{CookieMaxCalcs}}$) for varying the goodput ratio and MTBF in equation 1. As shown for realistic MTBF values (i.e. MTBF $\gg$ a request time), the number of cookies per request keeps small unless the required goodput ratio becomes extremely high: accepting a few lost calculations (e.g. a goodput ratio of 98% for a MTBF of 10 request times) – and the corresponding re-processing effort on a new PE – leads to reasonable overhead at good system performance.

The size of a state cookie is usually in the range of a few bytes up to multiple kilobytes (some examples are provided by [7, Subsubsection 9.4.2.2]). In order to further reduce the overhead, it is useful to examine the contents of a session state [32]. Usually, it consists of some *long-term sub-states* (which mostly remain constant) and some *short-term sub-states* (which change frequently). A useful strategy is to only transmit the changed fractions of the state as *partial state cookie*. Then, the PU can combine them with the already known long-term part sent in a *full state cookie*. Of course, the need to combine long-term and short-term parts in order to apply this so-called *state splitting* technique requires the PU to be aware of the cookie structure.

Knowing the state cookie structure allows for a further optimization: by using so-called *state derivation* [32], the

PU can derive parts of the session state from the application protocol data. Then, the PU itself is able to fill in parts of the cookie, avoiding transmission overhead. This mechanism requires further differentiation of the cookie parts:

- *public* parts may be read and modified,
- *immutable* parts may only be read (a PE can verify the integrity by signature; see Subsection 2.4) and
- *private* parts (which are encrypted and therefore understandable by the PEs only; see Subsection 2.4).

An application example is an audio on demand service: a state cookie could contain some authentication information, the name of the media file and the current playback position. While the authentication information is clearly private, the media name could be immutable (it is known by the PU-side application anyway) and the playback position could be public. Since the playback position is transmitted in each RTP frame [47], it can be filled in by the PU – there is no need to transmit it in form of a new (partial) state cookie.

## 6 Conclusions

RSerPool is the new IETF standard for server redundancy and session failover. In this article, we have introduced RSerPool – including the underlying SCTP protocol – with a focus on server failure handing. After that, we have provided a quantitative performance analysis of its server failure handling performance. This failover performance is influenced by two factors: (1) the failure detection speed and (2) the failover mechanism.

In any case, it is crucial to detect server failures as soon as possible (e.g. by session keep-alives or an application-specific mechanism). The PR-based server health monitoring is becoming important when the request size becomes small. Failure reports may be used to reduce its overhead significantly – if taking care of security. Using the "Abort and Restart" failover mechanism, a reasonable performance is already achieved with sufficiently reliable PEs at minimal costs. A more sophisticated but still very simple and quite effective failover strategy is Client-Based State Sharing. Configured appropriately, a good performance is achieved at small overhead.

The goal of our ongoing RSerPool research is to provide configuration and optimization guidelines for application developers and users of the new IETF RSerPool standard. As part of our future work, we are going to validate our results in real-life scenarios. That is, we are going to perform PLANETLAB experiments by using our RSerPool prototype implementation RSPLIB [7, 12, 26, 43]. Furthermore, we are going to analyse the failure handling features of the ENRP protocol in detail [11, 12].

## References

[1] T. Dreibholz and E. P. Rathgeb. Reliable Server Pooling – A Novel IETF Architecture for Availability-Sensitive Services. In *Proceedings of the 2nd IEEE International Conference on Digital Society (ICDS)*, pages 150–156, Sainte Luce/Martinique, February 2008. ISBN 978-0-7695-3087-1.

[2] E. P. Rathgeb. The MainStreetXpress 36190: a scalable and highly reliable ATM core services switch. *International Journal of Computer and Telecommunications Networking*, 31(6):583–601, March 1999.

[3] L. Alvisi, T. C. Bressoud, A. El-Khashab, K. Marzullo, and D. Zagorodnov. Wrapping Server-Side TCP to Mask Connection Failures. In *Proceedings of the IEEE Infocom 2001*, volume 1, pages 329–337, Anchorage, Alaska/U.S.A., April 2001. ISBN 0-7803-7016-3.

[4] F. Sultan, K. Srinivasan, D. Iyer, and L. Iftode. Migratory TCP: Highly available Internet services using connection migration. In *Proceedings of the ICDCS 2002*, pages 17–26, Vienna/Austria, July 2002.

[5] K. Echtle. *Fehlertoleranzverfahren*. Springer-Verlag, Heidelberg/Germany, 1990. ISBN 3-540526-80-3.

[6] D. Gupta and P. Bepari. Load Sharing in Distributed Systems. In *Proceedings of the National Workshop on Distributed Computing*, January 1999.

[7] T. Dreibholz. *Reliable Server Pooling – Evaluation, Optimization and Extension of a Novel IETF Architecture*. PhD thesis, University of Duisburg-Essen, Faculty of Economics, Institute for Computer Science and Business Information Systems, March 2007.

[8] T. Dreibholz and E. P. Rathgeb. An Evaluation of the Pool Maintenance Overhead in Reliable Server Pooling Systems. *SERSC International Journal on Hybrid Information Technology (IJHIT)*, 1(2):17–32, April 2008.

[9] Ian Foster. What is the Grid? A Three Point Checklist. *GRID Today*, July 2002.

[10] T. Dreibholz and E. P. Rathgeb. An Evaluation of the Pool Maintenance Overhead in Reliable Server Pooling Systems. In *Proceedings of the IEEE International Conference on Future Generation Communication and Networking (FGCN)*, volume 1, pages 136–143, Jeju Island/South Korea, December 2007. ISBN 0-7695-3048-6.

[11] X. Zhou, T. Dreibholz, F. Fa, W. Du, and E. P. Rathgeb. Evaluation and Optimization of the Registrar Redundancy Handling in Reliable Server Pooling Systems. In *Proceedings of the IEEE 23rd International Conference on Advanced Information Networking and Applications (AINA)*, Bradford/United Kingdom, May 2009.

[12] X. Zhou, T. Dreibholz, E. P. Rathgeb, and W. Du. Takeover Suggestion – A Registrar Redundancy Handling Optimization for Reliable Server Pooling Systems. In *Proceedings of the 10th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2009)*, Daegu, South Korea, May 2009.

[13] T. Dreibholz and E. P. Rathgeb. A Powerful Tool-Chain for Setup, Distributed Processing, Analysis and Debugging of OMNeT++ Simulations. In *Proceedings of the 1st ACM/ICST OMNeT++ Workshop*, Marseille/France, March 2008. ISBN 978-963-9799-20-2.

[14] T. Dreibholz and E. P. Rathgeb. On the Performance of Reliable Server Pooling Systems. In *Proceedings of the IEEE Conference on Local Computer Networks (LCN) 30th Anniversary*, pages 200–208, Sydney/Australia, November 2005. ISBN 0-7695-2421-4.

[15] A. Maharana and G. N. Rathna. Fault-tolerant Video on Demand in RSerPool Architecture. In *Proceedings of the International Conference on Advanced Computing and Communications (ADCOM)*, pages 534–539, Bangalore/India, December 2006. ISBN 1-4244-0716-8.

[16] Ü. Uyar, J. Zheng, M. A. Fecko, S. Samtani, and P. Conrad. Evaluation of Architectures for Reliable Server Pooling in Wired and Wireless Environments. *IEEE JSAC Special Issue on Recent Advances in Service Overlay Networks*, 22(1):164–175, 2004.

[17] M. Bozinovski, L. Gavrilovska, R. Prasad, and H.-P. Schwefel. Evaluation of a Fault-tolerant Call Control System. *Facta Universitatis Series: Electronics and Energetics*, 17(1):33–44, 2004.

[18] R. Stewart. Stream Control Transmission Protocol. Standards Track RFC 4960, IETF, September 2007.

[19] A. Jungmaier. *Das Transportprotokoll SCTP*. PhD thesis, Universität Duisburg-Essen, Institut für Experimentelle Mathematik, August 2005.

[20] R. Stewart, , Q. Xie, M. Tüxen, S. Maruyama, and M. Kozuka. Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration. Standards Track RFC 5061, IETF, September 2007.

[21] P. Conrad, A. Jungmaier, C. Ross, W.-C. Sim, and M. Tüxen. Reliable IP Telephony Applications with SIP using RSerPool. In *Proceedings of the State Coverage Initiatives, Mobile/Wireless Computing and Communication Systems II*, volume X, Orlando, Florida/U.S.A., July 2002. ISBN 980-07-8150-1.

[22] A. Jungmaier, E. P. Rathgeb, M. Schopp, and M. Tüxen. A multi-link end-to-end protocol for IP-based networks. *AEÜ - International Journal of Electronics and Communications*, 55(1):46–54, January 2001.

[23] P. Lei, L. Ong, M. Tüxen, and T. Dreibholz. An Overview of Reliable Server Pooling Protocols. Informational RFC 5351, IETF, September 2008.

[24] Q. Xie, R. Stewart, M. Stillman, M. Tüxen, and A. Silverton. Endpoint Handlespace Redundancy Protocol (ENRP). RFC 5353, IETF, September 2008.

[25] C. S. Chandrashekaran, W. L. Johnson, and A. Lele. Method using Modified Chord Algorithm to Balance Pool Element Ownership among Registrars in a Reliable Server Pooling Architecture. In *Proceedings of the 2nd International Conference on Communication Systems Software and Middleware (COMSWARE)*, pages 1–7, Bangalore/India, January 2007. ISBN 1-4244-0614-5.

[26] T. Dreibholz and E. P. Rathgeb. On Improving the Performance of Reliable Server Pooling Systems for Distance-Sensitive Distributed Applications. In *Proceedings of the 15. ITG/GI Fachtagung Kommunikation in Verteilten Systemen (KiVS)*, pages 39–50, Bern/Switzerland, February 2007. ISBN 978-3-540-69962-0.

[27] X. Zhou, T. Dreibholz, and E. P. Rathgeb. A New Server Selection Strategy for Reliable Server Pooling in Widely Distributed Environments. In *Proceedings of the 2nd IEEE International Conference on Digital Society (ICDS)*, pages 171–177, Sainte Luce/Martinique, February 2008. ISBN 978-0-7695-3087-1.

[28] R. Stewart, Q. Xie, M. Stillman, and M. Tüxen. Aggregate Server Access Protcol (ASAP). RFC 5352, IETF, September 2008.

[29] T. Dreibholz. Handle Resolution Option for ASAP. Internet-Draft Version 04, IETF, Individual Submission, January 2009. draft-dreibholz-rserpool-asap-hropt-04.txt, work in progress.

[30] T. Dreibholz and M. Tüxen. Reliable Server Pooling Policies. RFC 5356, IETF, September 2008.

[31] T. Dreibholz, X. Zhou, and E. P. Rathgeb. A Performance Evaluation of RSerPool Server Selection Policies in Varying Heterogeneous Capacity Scenarios. In *Proceedings of the 33rd IEEE EuroMirco Conference on Software Engineering and Advanced Applications*, pages 157–164, Lübeck/Germany, August 2007. ISBN 0-7695-2977-1.

[32] T. Dreibholz. An Efficient Approach for State Sharing in Server Pools. In *Proceedings of the 27th IEEE Local Computer Networks Conference (LCN)*, pages 348–352, Tampa, Florida/U.S.A., October 2002. ISBN 0-7695-1591-6.

[33] ITU-T. Introduction to CCITT Signalling System No. 7. Technical Report Recommendation Q.700, International Telecommunication Union, March 1993.

[34] T. Dreibholz, A. Jungmaier, and M. Tüxen. A new Scheme for IP-based Internet Mobility. In *Proceedings of the 28th IEEE Local Computer Networks Conference (LCN)*, pages 99–108, Königswinter/Germany, November 2003. ISBN 0-7695-2037-5.

[35] T. Dreibholz, L. Coene, and P. Conrad. Reliable Server Pooling Applicability for IP Flow Information Exchange. Internet-Draft Version 07, IETF, Individual Submission, January 2009. draft-coene-rserpool-applic-ipfix-07.txt, work in progress.

[36] T. Dreibholz, X. Zhou, and E. P. Rathgeb. SimProcTC – The Design and Realization of a Powerful Tool-Chain for OMNeT++ Simulations. In *Proceedings of the 2nd ACM/ICST OMNeT++ Workshop*, Rome/Italy, March 2009. ISBN 978-963-9799-45-5.

[37] T. Dreibholz and E. P. Rathgeb. The Performance of Reliable Server Pooling Systems in Different Server Capacity Scenarios. In *Proceedings of the IEEE TENCON '05*, Melbourne/Australia, November 2005. ISBN 0-7803-9312-0.

[38] X. Zhou, T. Dreibholz, and E. P. Rathgeb. A New Approach of Performance Improvement for Server Selection in Reliable Server Pooling Systems. In *Proceedings of the 15th IEEE International Conference on Advanced Computing and Communication (ADCOM)*, pages 117–121, Guwahati/India, December 2007. ISBN 0-7695-3059-1.

[39] X. Zhou, T. Dreibholz, and E. P. Rathgeb. Improving the Load Balancing Performance of Reliable Server Pooling in Heterogeneous Capacity Environments. In *Proceedings of the 3rd Asian Internet Engineering Conference (AINTEC)*, volume 4866 of *Lecture Notes in Computer Science*, pages 125–140. Springer, November 2007. ISBN 978-3-540-76808-1.

[40] T. Dreibholz and E. P. Rathgeb. Towards the Future Internet – An Overview of Challenges and Solutions in Research and Standardization. In *Proceedings of the 2nd GI/ITG KuVS Workshop on the Future Internet*, Karlsruhe/Germany, November 2008.

[41] A. Varga. *OMNeT++ Discrete Event Simulation System User Manual - Version 3.2*. Technical University of Budapest/Hungary, March 2005.

[42] R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna/Austria, 2005. ISBN 3-900051-07-0.

[43] T. Dreibholz and E. P. Rathgeb. A PlanetLab-Based Performance Analysis of RSerPool Security Mechanisms. In *Proceedings of the 10th IEEE International Conference on Telecommunications (ConTEL)*, Zagreb/Croatia, June 2009.

[44] P. Schöttle, T. Dreibholz, and E. P. Rathgeb. On the Application of Anomaly Detection in Reliable Server Pooling Systems for Improved Robustness against Denial of Service Attacks. In *Proceedings of the 33rd IEEE Conference on Local Computer Networks (LCN)*, pages 207–214, Montreal/Canada, October 2008. ISBN 978-1-4244-2413-9.

[45] X. Zhou, T. Dreibholz, W. Du, and E. P. Rathgeb. Evaluation of Attack Countermeasures to Improve the DoS Robustness of RSerPool Systems by Simulations and Measurements. In *Proceedings of the 16. ITG/GI Fachtagung Kommunikation in Verteilten Systemen (KiVS)*, pages 217–228, Kassel/Germany, March 2009. ISBN 978-3-540-92665-8.

[46] T. Dreibholz, E. P. Rathgeb, and X. Zhou. On Robustness and Countermeasures of Reliable Server Pooling Systems against Denial of Service Attacks. In *Proceedings of the IFIP Networking*, pages 586–598, Singapore, May 2008. ISBN 978-3-540-79548-3.

[47] T. Dreibholz. Management of Layered Variable Bitrate Multimedia Streams over DiffServ with Apriori Knowledge. Masters Thesis, University of Bonn, Institute for Computer Science, February 2001.

# A Privacy-Maintaining Framework for Context-Sensitive Service Discovery Services

Colin Atkinson, Philipp Bostan, Thomas Butter, Wolfgang Effelsberg
Institute of Computer Science
University of Mannheim
Mannheim, Germany
{atkinson, bostan, butter}@uni-mannheim.de, effelsberg@pi4.informatik.uni-mannheim.de

*Abstract*—**Despite the rapid growth in the number of mobile devices connected to the internet via UMTS or wireless 802.11 hotspots the market for location-based services has yet to take off as expected. Moreover, other kinds of context information are still not routinely supported by mobile services and even when they are, users are not aware of the services that are available to them at a particular time and place. We believe that the adoption of mobile services will be significantly increased by context-sensitive service discovery services that use context information to deliver precise, personalized search results in a changing environment and reduce human-device interaction. However, developing such applications is still a major challenge for software developers. In this paper we therefore present a framework for building context-sensitive service discovery services for mobile clients that ensures the privacy of the users' context while offering valuable search results.**

***Keywords: context-aware systems, service discovery, mobile applications.***

## I. INTRODUCTION

Although an increasing number of mobile devices are equipped with the ability to automatically sense and identify their location, the use of location based services has so far failed to reach the levels expected. To date, only a few mobile services are receiving widespread use, such as push-E-Mailing services and navigation services, and a so called "killer application" has yet to emerge. Moreover, other sensor technologies have rapidly evolved over the last few years and now support the automatic sensing of other kinds of context information on mobile devices. Together these provide the basis for the next generation of services for mobile devices and users – context-sensitive services that deliver value tailored to a user's context. However, supporting commercial context-sensitive services that are usable by a broad range of mobile users as well as services that offer high revenue for service and telecommunication providers is still a major challenge for developers.

In contrast to desktop applications, mobile applications must cope with additional problems arising from the influence of their environment. Key challenges include mobility, resource limitations, heterogeneity, personalization and stricter requirements on usability. In view of these constraints, it is even more important to provide mobile users with enhanced support to find suitable services effectively. Humans should not have to engage in long-winded interaction patterns in order to find services since mobile devices provide only limited input capabilities. Users are also usually unwilling to enter large search requests typical of browser-oriented search engines on desktop computers. Context-sensitive service discovery has the potential to deliver significant added value since it considerably reduces the level of interaction required from the user. Furthermore it delivers personalized, precise search results that are tailored to the user's current situation.

To provide better support for context-sensitive service discovery, the SALSA (**S**oftware **A**rchitectures for **l**ocation-**s**pecific Tr**a**nsactions in Mobile Commerce) project of the Mobile Business Research Group at the University of Mannheim has developed a generic service discovery platform for service brokers and service providers. This platform also offers a client framework that supports the development of generic mobile client applications for use with context-sensitive service discovery as well as the dynamic integration and execution of discovered services. An important strength of the SALSA approach is that it considers the user's privacy in the process of context-sensitive service discovery. Especially when sensitive context information like a device's current location is involved, many mobile users fear that context information about them will be misused. As will be presented in this paper, this issue is handled in the overall SALSA architecture by a mechanism that ensures privacy during service discovery.

The main goal of the SALSA project was to develop a generic framework and a reference architecture to support service discovery based on context information. The implementation of our prototype was therefore inspired by the following scenario. A mobile user is moving around in a possibly unknown area (e.g. a foreign city), and would like to use some of the real-world services that are available in that vicinity. The user might also be interested in various electronic services (e.g. a tourist guide) that are specialized for a certain domain and fit his current situation. This is mainly described by his context attributes that may include such things as his current location, the current time, the current weather, and his profile.

Since the number of potentially available services could be very large in this scenario, the system should use this context information to drive the discovery of suitable services. Key-word based searches of the kind applied in desktop browser-oriented search applications are not useful in such a scenario because browsing a large result set is not appropriate for users of small, tiny mobile devices with limited input capabilities. Thus, the approach to context-sensitive service discovery that we present in this paper enhances a mobile user's search request with a set of context attributes that drive the service discovery process. The search results that are returned upon a search request are tailored to the user's current situation, containing a choice of the most relevant services in a personalized, ranked order. This tremendously reduces the human-device interaction involved in searching for suitable services and thus the user's overall level of effort.

The remainder of this paper is structured as follows. Section II starts with a discussion of previous definitions of the term of "*context*". We analyze these definitions in detail and afterwards introduce our view of context. Then, we introduce a simple context model and the approach used to represent context in our framework. This is followed by a classification of services and our schema for service descriptions required for the matching of context and services. In Section III, the overall SALSA architecture is discussed in more detail. We first introduce the basic principles of a Service Discovery Service (SDS) that uses context for the process of service discovery. Following that, the server and the client framework are presented which both offer systematic support for the development of applications and services that support context-sensitive service discovery. In Section IV, we describe the underlying context framework that supports the handling of context on different layers. These layers include context sensing, context resolution and context aggregation and inference. A detailed consideration of our approach for context-sensitive service discovery, based on the matching of context and service descriptions, is presented in Section V. We also introduce our approach for transforming service descriptions as well as our approach for context matching. In Section VI, we present past research work that is related to context-sensitive service discovery and frameworks. Finally, Section VII closes the paper with a summary of our presented approach and technology.

## II. Context and Service Representation

In this section we introduce and discuss previous definitions of context and present the definition that we use in our approach. We then present a simple context model and a representation format suitable for context processing and matching within our framework. This is followed by the presentation of various definitions related to services and a schema that supports the matching of services against context for the purpose of service discovery.

### A. *The Definition of Context*

In past research, many definitions of the term "*context*" for context-aware computing have been introduced. In [2], Baldauf et al. present a survey on context-aware systems that contains various definitions of the term "context". Many of the definitions published in early research work have been redefined and extended over time. As the authors in [3] point out, most of the past definitions are based on concrete examples and categories, while other definitions use paraphrases or synonyms. This also indicates the problems involved in coming up with a clear definition of the notion of context.

From a natural langrage point of view, the term is defined in the Merriam Webster dictionary [4] as follows: *"the parts of a discourse that surround a word or passage and can throw light on its meaning"* and *"the interrelated conditions in which something exists or occurs: the Environment, the Setting"*. The first part of this definition can be interpreted to mean that context is something implicit that can be used as additional information to give something an enhanced meaning. The second part of the definition has a more general nature and defines synonyms for context. In the free online dictionary of computing [5] context is defined as: *"That which surrounds, and gives meaning to, something else"*. Both of these definitions leave a great deal of room for interpretation and cannot be transferred directly into a form that allows context-aware systems to determine whether or not different kinds of information can be regarded as context.

One of the first definitions was given by Schilit and Theimer (initiators of context-aware computing with the PARCTab project) in 1994 in [6]. They define software as context-aware if it uses *"location information to adapt according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time"*. In later work Schilit et al. define the three important aspects of context as: *"Where you are, who you are with and what resources are nearby"*. They also enumerate a few examples, like lightning, noise level and others.

In 1996, Brown defines context within the stick-e-document project as: *"Context can be a combination of elements of the environment that the user's computer knows about"* in [7] and also enumerates a few examples. This definition leads especially to the question – how does the computer know about the user's environment? This is also raised by Brown in [8] where he discusses the question of whether context is automatically detected or user-delivered information (implicit vs. explicit). In 1997, Ryan, Pascoe and Morse introduced their definition of context in a mix of paraphrases and examples as: *"Context-awareness describes the ability of the computer to sense and act upon information about its environment, such as location, time, temperature and user identity"* [9]. This definition has been refined by Pascoe into: *"Context-awareness is the ability of a program or device to sense various states of its environment and itself"*. It highlights the implicitness of context. In a later definition he adds that context is: "*the subset of physical and conceptual states of interest to a particular entity*".

This concept of defining context in relation to some entity was later elaborated by Dey et al in [10] leading to one of the most widely used and often cited definitions: *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications* *themselves"*. Many recently introduced definitions are based on the definition of Dey or extended versions of this definition. Other recently introduced definitions try to offer a more sophisticated view of context [3]. However, not only do they introduce additional complexity, most of these definitions fail to clarify exactly what kinds of information are to be regarded as context.

In summary, we can observe that defining context by enumerating examples is static and places limitations on what can be regarded as context. This can be improved by defining a taxonomy of context categories. However, none of the enumeration-based approaches provide a deterministic way of establishing whether or not a particular piece of information is context. All definitions based on synonyms are also vague since they leave a great deal of room for interpretation. Furthermore, they shift the problem to the detailed definition of the chosen synonym, most of which are too generic. Too much freedom for interpreting the notion of context essentially removes any semantics from the term since everything can then be regarded as context. Using more complex and formal definitions of context that subdivide context into different categories do not solve the basic problem either.

The basic problem with most of the definitions is that it is very difficult, and sometimes impossible, to determine whether or not a particular piece of information can be regarded as context. In order to avoid this, in our project we have develop a refined definition of context that conveys useful semantics (i.e. that distinguishes between information that is context and information that is not context). We think that the most important distinction between context and "pure" information is related to the question of whether context is implicit or explicit information. In [11] Pascoe, gives a definition of context that includes the notion that context is "sensed" information which does not have to be explicitly provided by the human. Lieberman et al. also identify the distinction between implicit and explicit context in [12] and Dey states that context may either be determined implicitly (e.g. from a personalized device) or explicitly by user-delivered data in a login dialogue. It is assumed that in both cases the identity of the user is the same and thus both should be regarded as context.

Considering the example of a weather service, a user could explicitly enter the name of the city for which he wants weather information or the city name could be implicitly determined by sensors that detect the mobile device's current location. In the second case, it is clear that the implicitly determined information (i.e. the city name) should be viewed as context, but what about the first case where the name of a city is explicitly input by the user? What if the city name entered by the user does not correspond to his current location? How can the weather service determine if the entered city really belongs to the current context of the user or not? In general, how can a system determine whether it should treat a particular piece of information as context or not? It is clear that the boundary between implicit and explicit information is somehow fuzzy, but in fact implicitly delivered information, and information derived from it, is less prone to errors and thus more reliable and valuable for applications and services. Furthermore, the decision about whether information should be treated as context within the boundaries of a certain system or application must be made at design-time, since context is relative to the system. For the SALSA project and the work presented in this paper we use the following definition: *"Context is any relevant information that can be gathered implicitly or derived based on implicitly gathered information within the boundaries of a system or application and is used to determine the behavior of a system or service"*.

### B. Context Representation in SALSA

The overall SALSA framework consists of a client and a server framework, which both provide general support for the development of context-sensitive mobile applications. Both parts of the framework need to collect and process context, but most of the context is collected on the mobile client and enhanced through additional context processing in the server framework.

For these purposes a common representation of context is needed that is shared between the involved participants. While many complex models for representing context have been proposed [13,14], we have chosen a simple, flexible representation that is easy for service providers to apply and manage. In SALSA, a context set is represented by an XML document that contains multiple context attributes. Each of the contained context attributes is in principle an instance of a certain context data type. To support run-time validation of the context set and to provide a template for the definition of context attributes, we have defined a few standard context data types as an XML schema. Context data types are similar to data types of typical programming languages (e.g. *String*, *Integer*, etc.), but they are extended by a few specific data types (e.g. *PositionCircle* represents geographical areas). An example context set is shown below.

```
<Context>

  <PositionCircle name="SALSA.Position.GeoPosition">
    <Center>
      <Latitude>49.48739</Latitude>
      <Longitude>8.4705</Longitude>
      <Altitude>88</Altitude>
    <Center>
  </PositionCircle>

  <Time name="SALSA.Time.CurrentTime”>
    <Hours>12</Hours>
    <Minutes>32</Minutes>
    <Seconds>21</Seconds>
  </Time>

  <String name="SALSA.Time.Weekday">
    <Value>Saturday</Value>
  </String>

  <Integer name="SALSA.Time.FreeTime">
    <Value>30</Value>
  </Integer>

</Context>
```

This example represents a context set that contains four different context attributes using different context data types using our XML representation schema. It contains the geographical position of a mobile user in GPS coordinates, the time and weekday on which the search request was issued, and the user's available free time inferred from personal data in the mobile client's calendar.

A context attribute is characterized by a specification of its data type, its name and its respective values. The name is constructed using namespaces to provide a globally unique identifier which is important for the context matching process as will be explained in Section V. If an application or service needs data types that are not supported in the standard XML schema, the SALSA framework allows arbitrary new context data types to be added in a simple way. This can be achieved by the extension of the XML schema to include new context data types as needed.

### C. *Service Types and Descriptions*

The approach for context-sensitive service discovery presented in this paper allows various different kinds of services to be discovered. In general, we define a *service* as anything that delivers some kind of value to someone or something. The main distinction we make is between *electronic services* and *non-electronic services*. We define *electronic services* as services that deliver value to the mobile user by electronic means in the form of information. In the scenario introduced in Section I, examples of electronic services are an electronic gastronomy guide, an event-guide, a bargain hunter or a tourist guide. Electronic services can be offered in various different forms, for example as traditional web sites or as Web services using SOAP and WSDL.

A *non-electronic real-world (business) service* is defined as a service that does something to improve the state of a user in a certain way. Examples related to our scenario are a restaurant, a bar, a tourist site or a shop. The Venn diagram in Figure 1 presents the relationship between these service categories. It basically shows that Web services are electronic services and electronic services are services in general.

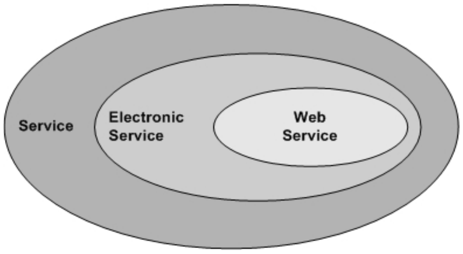


Figure 1. Service Taxonomy.

The following four types of services can be described and discovered using our approach for context-sensitive service discovery:

- real-world (business) services (non-electronic),
- Web services (electronic),
- web sites (electronic),
- SALSA services (electronic).

Real-world (business) services are represented as services in our Venn diagram. The fourth kind of service – the SALSA service – is specific to SALSA and consists of a Web service (representing the service interface) and optionally downloadable software components for dynamic installation and execution in the SALSA client framework. These may be graphical user interface components for interaction with the service, business logic components or security components for example.

To describe these different service types we have developed a generic XML schema that consists of two separate parts: a core part and a domain-specific part. Like OWL-S [15], the core description part, as depicted in Figure 2, is divided into three categories - *ServiceProfile*, *ServiceProperties* and the *ServiceGrounding*. We have extended and tailored the profiles of OWL-S for the purpose of context-sensitive service discovery. The domain-specific part is used to extend service descriptions in cases where additional attributes are needed to describe services of a domain whose properties are not covered by the generic schema. The description of gastronomy places, for example, requires additional attributes to describe their special characteristics. The separation of the core description of the common properties from the domain-specific extensions adds more flexibility and allows service providers to tailor service descriptions to the service's domain.

The **ServiceProfile** mainly contains general information about the service and its provider. First there is a textual description of the service along with its classification according to one of the following schemas: UNSPSC (United Nations Standard Products and Service Codes) [16], the NAICS (North American Industry Classification System) [17] or an arbitrary, self-defined categorization schema. Furthermore, contact information about various key individuals that have some responsibility related to the service is included. The domain-specific part of service descriptions can optionally be added to the *ServiceProfile*. For example, a context-sensitive event guide service that needs to add specific properties to event description can include its own schema in the *ServiceProfile*.

The **ServiceProperties** section of our service description schema introduces the properties of a service, including the spatial and temporal availability of the service as well as aspects like payment and security restrictions. While a real-world (business) service clearly has spatial and temporal restrictions, at first sight it is not so obvious that electronic services may have such properties as well. However consider an electronic context-sensitive shopping guide for the city of Berlin. By providing temporal availability information about the general opening times of stores and malls, the system is able to avoid returning services that are only available during the day in response to service requests issued at night when stores are closed. Also spatially restricting this specialized shopping guide to the Berlin area allows the system to avoid returning this service in response to search requests issued in a different city. While the value of temporal availability attributes for non-electronic (business) services is self-explanatory, spatial restriction information can be used for example to define an area of service delivery. For example, a pizza service might only deliver its service in a circular area with a radius of 10 miles. Finally payment and security restrictions are described in the *ServiceProperties* category. These are mainly used to filter out services during the service discovery process which do not fulfill requirements

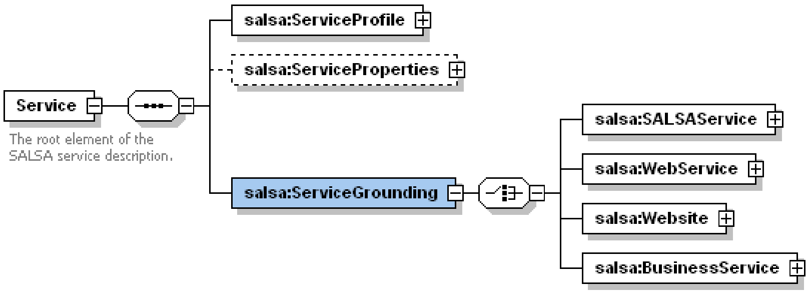


Figure 2. Service Description Core.

concerned with payment methods and security restrictions defined by the user or the service itself.

Finally, the **ServiceGrounding**, the last category of our service description approach, delivers access information for the different services depending on their type. A non-electronic service, for example requires the description of its physical location while an electronic service requires access information like web or server addresses. The difference between the four different types of services – Web Services, Business Services, Websites and SALSA services – is mainly reflected in their different groundings as indicated in Figure 2. Each type of service requires its own description schema for the service grounding. A Web service defines its WSDL grounding, a web site defines its internet address and a real-world (business) service defines its physical and geographical location. While the first three types obviously provide the necessary access information, the latter one, SALSA Service, needs to be considered in more detail. This type defines Web service port types for its service interface and optionally multiple internet addresses for downloadable software components to support dynamic reconfiguration of the mobile client.

## III. The SALSA Architecture

In this section, we introduce the architecture that was developed within the SALSA project to support context-sensitive service discovery. We first introduce the basic ideas behind our *Service Discovery Service (SDS)* based on the principles of service-oriented architectures. We then present the architecture of the prototype we implemented to verify the scenario introduced in Section I. This is followed by a description of the SALSA server framework that supports the implementation of context-sensitive SDSs. Finally, we consider the SALSA client framework that offers a simplified way to implement arbitrary context-sensitive mobile client applications as well as client applications that interact with *Service Discovery Services*.

### *A. Service Discovery Services*

At the heart of our overall context-sensitive service discovery approach is the concept of a *Service Discovery Service* (SDS) that acts as a service broker in the SALSA architecture. Following the principles of the famous SOA triangle shown in Figure 3 an SDS has the role of the service broker (registry). The SDS allows service providers to register their services using a service description that follows the schema introduced in Section II. A mobile client corresponds to the service requestor and uses the SDS as a service broker to find suitable services. The mobile user finally uses or consumes a service, which is either an electronic service or a real-world (business) service.

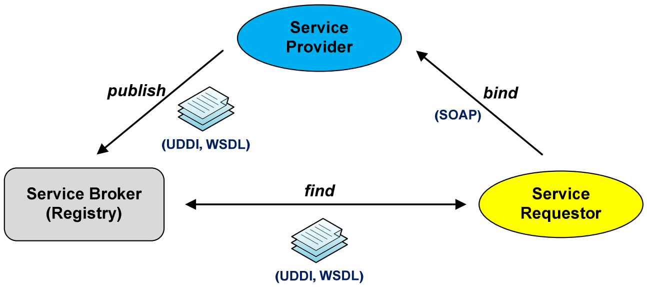


Figure 3. The SOA triangle.

In another paper [18], we have identified and analyzed several configurations that could theoretically be applied for the scenario that we introduced in Section I. For various reasons (evaluated also in [18]) we adopted the *User-Managed Linear Configuration* depicted in Figure 4 for our prototype.

This configuration is characterized as “user-managed” and “linear” because the user is involved in the selection of lower-level service brokers that are also SDSs. The main advantages of this configuration are enhanced privacy, transparency in pricing of SDSs and much lower bandwidth requirements since communication between the mobile client and the SDSs is minimized. Also, the client application consists of less complex software. The user-managed, linear configuration presented in Figure 4 involves the *User*, a *Mobile Client*, the *Universal SDS (USDS)*, multiple lower-levels *SDSs* and multiple *real-world (business) service providers*.

According to the scenario introduced in Section I, the mobile user is interested in immediately getting value from real-world (business) services that are relevant to his current context. Thus, the mobile client first sends an initial search request to the Universal SDS (USDS) that acts as a first-level service broker. This initial search request consists of implicit context collected on the mobile client. The USDS uses our context-sensitive service discovery technology to return a list of service descriptions of lower-level SDSs. These are specialized service brokers that have registered themselves as service providers at the USDS. Examples of lower-level SDSs are tourist guides, event guides and gastronomy guides. The mobile user selects one of these specialized service brokers from the returned list of suitable services (e.g. based on costs or user rating, etc.) and the mobile client then connects to the service broker. A new search request, enhanced by new explicit parameters (e.g. domain-specific parameters input via the service's graphical user interface) and the determined context set is then sent to the chosen lower-level service broker (SDS) which returns a list of suitable service providers. These service providers (e.g. a restaurant, a bar, a café, etc.) in turn have registered their services with the specialized second-level or lower-level SDSs.

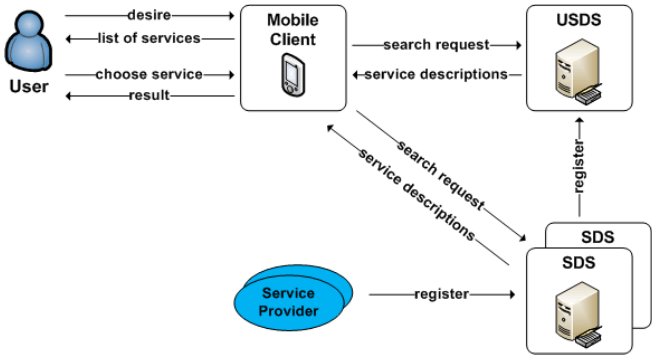


Figure 4. User-Managed Linear Configuration.

In the case where the lower-level broker is a gastronomy guide, for example, the returned list consists of an ordered set of service descriptions representing gastronomy places. On the other hand, in the case of a bargain hunter service, a list of descriptions of shops with special offers (e.g. coupons, etc.) is returned. In all cases, the returned services are ordered by the degree to which their properties matches the context of the request, as will be explained in Section V. Finally, the mobile user can choose a service provider from the list and consume the service (e.g. by going to eat in a restaurant, visiting a tourist site, using coupons in a shop, etc.). In the case where the lower-level broker has registered electronic services that are even lower-level brokers, this interaction can be performed over multiple levels.

In Figure 5, we present an alternative configuration which is called the *Server-Managed Hierarchical Configuration*. Using this configuration, each SDS aggregates search results from specialized lower-level SDSs in a hierarchical way. This minimizes communication between the mobile clients and SDSs, but it reduces the flexibility in user-managed service selection and especially leads to problems when payment is involved. In this configuration the user has no control over which SDS is contacted. Furthermore, as we have already discussed in [18], multiple alternative configurations are possible. For example, SDSs can be federated by location or category. Moreover, mobile clients can aggregate search results from lower-level specialized SDSs returned by the USDS, similar to the presented configuration in Figure 5.

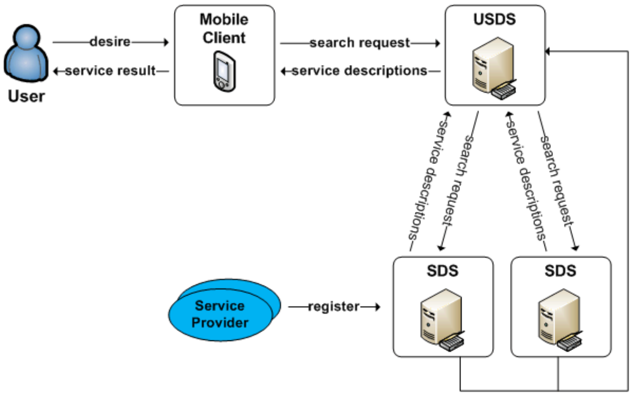


Figure 5. Server-Managed Hierarchical Configuration.

## B. Server Framework

Our SDS technology takes the form of a generic server framework that implements all the functionality needed to realize services offering context-sensitive service discovery. In this subsection we introduce the basic components of the SALSA server framework. Since our architecture follows a component-based design, all of the system components can be exchanged with other supporting technologies as long as the interfaces remain the same. For example, our service registry realized using an XML database could be replaced by a relational database. Figure 6, presents an overview of the server framework and its various components.

In principle, each SDS consists of a *Service Registry*, a *Service Search Engine* and multiple components that are responsible for context processing. Service registration can be supported in various ways. The service broker may provide web-based forms on a web site portal to support domain-specific service registration and the management of the registered service description. Another possibility is the automated transformation of existing service descriptions or database schemas to the XML document representation and automated registration via the API of the *Service Registry*. In our prototype the registry is realized using the Natix XML database [19] developed at the University of Mannheim. Natix stores the XML-based service descriptions directly and uses an optimized query mechanism based on the XPath query language [20] to retrieve XML documents. Every incoming search request, enhanced by a context set, is handled first by the *Service Search Engine* component which coordinates the following processing steps:

- pre-processing of the received context set,
- querying the registry for service descriptions with explicitly defined query parameters,
- transformation of the service descriptions into a contextual representation for matching,
- matching the context set against the transformed service descriptions.

The pre-processing of the received context set is handled by the *Context Manager*. It invokes the *Context Resolution Engine (CRE)* to enhance context using external *Context Provisioning Services (CPS)* and the *Context Aggregation* component to infer new context attributes. The *Service*

*Search Engine* is responsible for querying the registry for service descriptions using XPath queries based on the explicitly defined search parameters. The explicitly defined search parameters in a search request mostly relate to the domain-specific extensions of service descriptions (e.g. a user might specify "*Italian*" kitchen within a gastronomy guide service).

The *Service Transformation* component is used to transform pre-filtered service descriptions into a contextual representation for context matching. Finally the *Context Matcher* is used to match the pre-processed context set against the transformed list of service descriptions so that the *Service Search Engine* can return a list of suitable services to the mobile client. In Section V we will consider this process in more detail when we refer to context-sensitive service discovery.

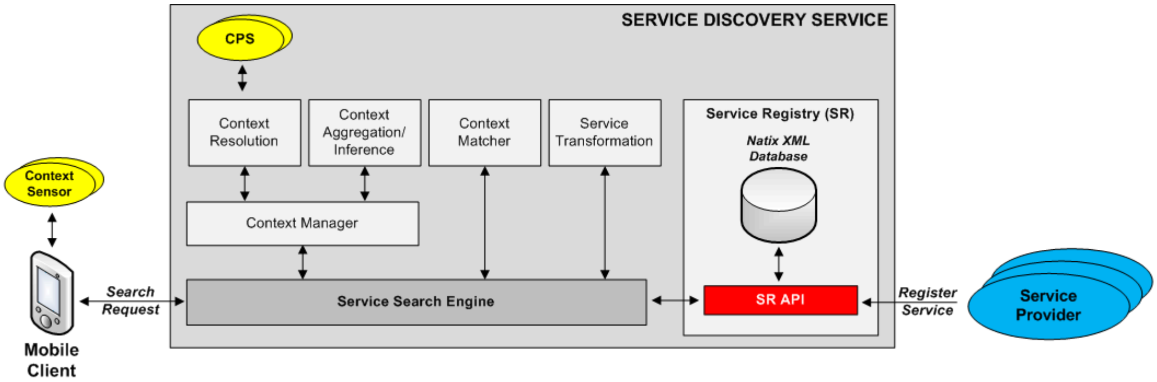


Figure 6. The SALSA Server-Framework.

### C. Client Framework

This subsection introduces the SALSA client framework that provides the basic support for the development of mobile client applications that are able to interact with SDSs for context-sensitive service discovery. The client framework has been developed in a generic and component-oriented way in order to support the implementation of independent, self-contained context-sensitive applications for other purposes as well. An overview of the client framework and its components (described in another paper in more detail [21]) is presented in Figure 7. In the following paragraphs we will briefly explain each of the framework's components.

The *Context Manager* component is responsible for the management of context sources and the updating and delivery of context attributes. The *Context Manager* can be subscribed to Context Sources that use either a "*pull*" or "*push*" mechanism to obtain current context information. A further responsibility of the *Context Manager* is the administration of context attributes that can be declared by the user as "*public*", "*private*" or "*blurred*" as will be explained in more detail in Section V. If search requests are initiated by the mobile user, the *Context Manager* is responsible for creating a context set that contains all available context attributes and their respective values. This context set is serialized in an XML representation and embedded in every search request that is sent to a service provider as an implicit parameter. Finally, the *Context Manager* is used for local context matching when search results are received by the client and need to be matched with "*private*" context attributes as will be elaborated in Section V.

The *Component Manager* of the client framework is responsible for the life-cycle management of components used to reconfigure the mobile client application. An important capability of the SALSA framework is that service providers can provide downloadable components to enhance the client-side graphical user interfaces, business logic or security services. Furthermore context sensors and sources can also be added to a client application by the *Component Manager*.

The generic *Communication Framework* implemented in SALSA offers support for well-established Enterprise Computing communication protocols such as SOAP or IIOP from the Common Object Request Broker Architecture (CORBA) [22]. It therefore offers an API on top of the communication layer so that different communication protocols can be supported. The current version implements the SOAP protocol [23].

The *Security Manager* is responsible for ensuring secure communication. It is therefore connected to the previously introduced *Communication Framework*. Furthermore it optionally offers anonymous communication using a TOR (The Onion Router) anonymity network approach.

The *GUI framework* [24] implemented in the SALSA framework is based on the XML User Interface Language (XUL) and offers support for the implementation of adaptable user interfaces to cope with several issues. The XUL approach separates the presentation and application logic whilst offering portability for different Java ME platform configurations. This furthermore eases the porting of graphical user interfaces to different client devices with different configurations. The reconfiguration and adaptation of user interfaces is especially useful if the current context changes (e.g. the mobile user is driving at high speed in a car and the content presentation is adapted accordingly). The implemented GUI framework avoids extensive programming effort for developers of mobile applications.

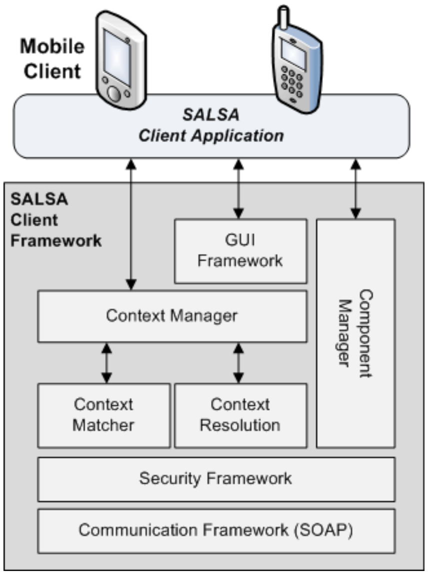


Figure 7. The SALSA Client-Framework.

## IV. CONTEXT HANDLING IN SALSA

Before we introduce our approach for context-sensitive service discovery in Section V, in this section we describe how context is handled in the SALSA framework. This essentially involves different context and service description post-processing steps which are needed to match the context to transformed service descriptions. To this end, we first introduce a layered model of context processing followed by a detailed consideration of each layer.

### A. *Context Processing Layers*

The basic goal of context processing is to enhance the client-delivered context information to the highest possible level to support higher precision service retrieval in the process of context-sensitive service discovery. Figure 8 introduces the different context processing layers which are in general supported by the framework. Each of these layers is realized through different components and mechanisms. We will first discuss the layers in an abstract way, and then introduce details of the mechanisms and examples for each layer in the following subsections.

In the first layer, which is called *context sensing*, raw sensor data received from context sensors is delivered to the context sources. These convert raw data into SALSA context attributes using the specified context data types. Context sources are the primary providers of context attributes for the upper context layers. In the second layer, named *context resolution*, a mechanism is applied which enhances the low-level context into new, higher-level context attributes. These new context attributes either represent a low-level context attribute with another meaning or a new, independent context attribute which has been derived from low-level context attributes. The delivery of new context attributes in this layer is realized by so called *Context Provisioning Services* (CPS) which will be explained in part C of this section in more detail. Finally, the third layer, called the *context aggregation and inference* layer, introduces another kind of high-level context that is generated from multiple context attributes based on pre-defined rules.

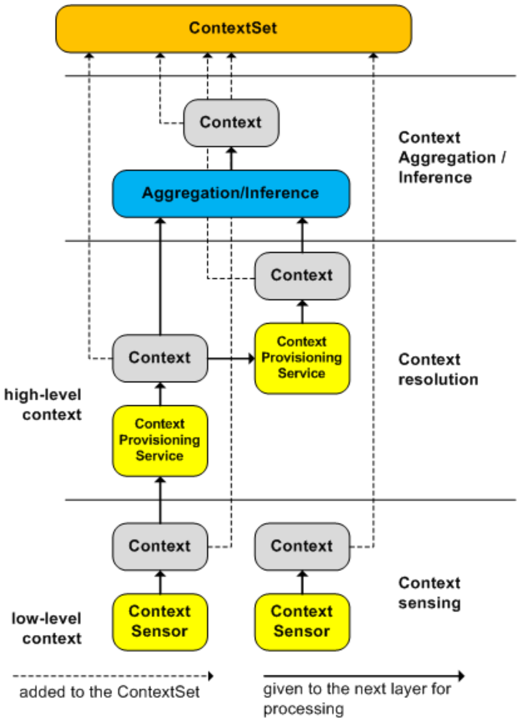


Figure 8. Context Processing Layers

All context attributes together build the context set that is used in the process of context-sensitive service discovery. As indicated in Figure 8, context attributes need not necessarily be processed to higher-level context and can be placed directly from the first or second layer into the final context set used for subsequent context matching. In SALSA, both the client and the server framework support context processing. While the server framework supports all layers presented in Figure 8, only the first two layers are supported in the client framework due to the restricted resources available. The second layer, context resolution, is only supported in a limited way in the client framework for the same reason.

### B. *Context Sensing*

The essential ingredients for high precision service retrieval, based on context-sensitive service discovery, are context sensors and context sources which deliver the initial context attributes. Context processing in general starts on the mobile client with the delivery of data detected by context sensors in the form of raw sensor data (e.g. the current location in GPS coordinates) or components on the mobile client that act as context sources and deliver implicit context attributes like the current time or the user's free time. Theoretically, different context sensors and sources can be integrated into the *Context Manager* in the client as well as in the server framework, but in practice they mostly reside on the mobile client. Context sources and sensors work together either in a push or pull model. Context sources are registered at the local *Context Manager* which automatically pulls context attributes from registered context sources when a search request is issued by the mobile user. Alternatively, the context sources push the context attributes based on a local event or at regular time intervals.

To verify our approach and build our first prototype, we implemented a few context sensors and context sources which deliver different kinds of context attributes collected on mobile clients. The main focus was on the development of high-precision positioning technologies. This was realized using a sensor fusion approach that supports algorithms for indoor and outdoor positioning based on GPS, wireless network and Bluetooth technologies. As well as the positioning of mobile users, we have developed algorithms that use a digital compass to determine a mobile device's alignment [25]. Using the context sensors for positioning we have implemented several context sources that offer context attributes like the geographical position, alignment or speed, which is estimated using collected positioning data over time. Other context sources implemented for our prototype deliver context attributes like the mobile devices screen resolution, color depth, network speed, connection type, current time, personal free time and other information derived from user profiles.

### C. *Context Resolution*

Moving to the next layer in our context processing architecture, the low-level context delivered by primary context sources is resolved to create additional context attributes for the final context set. As explained previously and shown in Figure 9, Context Provisioning Services (CPS) are the services that deliver this additional capability. In principle, there are two different kinds of CPSs available. The simple form of CPS acts only locally and is mostly based on simple resolution algorithms that, for example, resolve the current date into a week day or the current time to the time of day. The other kind is more complex and uses external services to resolve context attributes - for example, turning the current location from GPS coordinates into a city name (e.g. using an external GeoService) or resolving a city name into the weather information for that city. Both of these examples use external services to obtain the required context attributes. Due to their complexity and the fact that they use external services that require network and power resources which are expensive in mobile networks, complex CPSs are generally only used within the SALSA server framework. In contrast, simple CPSs may be used in mobile clients as well.

Once the context manager of a mobile client has pulled all context attributes from the registered context sources when a search request is initiated, the resulting context set is processed in the *context resolution* layer by the *Context Resolution Engine*. This processing step is executed on the client framework with a simplified engine and on the server framework with full functionality using the implicit context set in an incoming search request. The architecture of the *Context Resolution Engine* is based on SOA principles as presented in Figure 9.

CPSs are registered at the CPS registry that stores all descriptions. The *Context Resolution Engine (CRE)* receives a context set and uses the CPS registry to search for suitable CPSs. It then invokes these directly to resolve the context attributes that are contained in the context set. The description of a CPS contains a unique service ID, a name and a link to a WSDL description. All CPSs on the server framework are implemented as Web services while those on the client framework are implemented as simple classes. The most important part of the description provides information about the context data type and attributes that the service is able to resolve and the data type and attributes it finally delivers as a result.

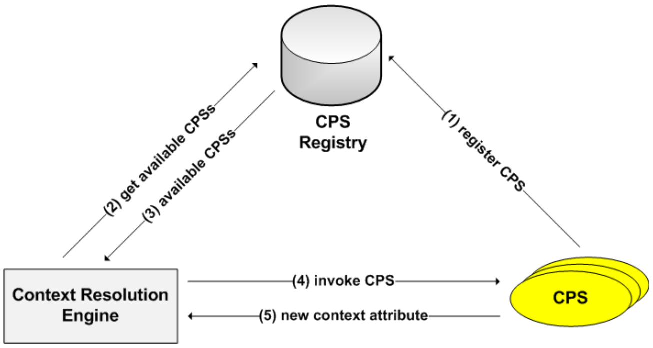


Figure 9. Context Resolution

The API of the CRE in general offers three possible usage modes. In the first mode, a request for context resolution specifies the context attribute of the context set that should be resolved and the CRE tries to find a suitable CPS in the registry that is able to do this resolution. In the second mode, a request specifies the context attribute that should be delivered to the CRE. The CRE then looks in the CPS registry for CPSs that can resolve to this context attribute. With the required input specified in the CPS description, the CRE checks if the provided context set contains the required context input. In the third mode, which is the most prevalent mode, the resolution request specifies the context set as a parameter. The CRE then takes the context set and goes through each context attribute, checking whether there is a CPS in the registry that can resolve a context attribute to new context. If this is the case the CRE sends a call to the CPS, receives the resolved context and adds it to the beginning of the context set description. When all context attributes that are contained in the context set have been processed, the described procedure starts again going through all context attributes of the list since it could be possible to further resolve one of the newly added context attributes. This recursive process continues until the CRE finds no more context attributes to add. Finally all possible resolutions of the client-delivered context set have been applied and the CRE finally returns the enhanced context set to the *Context Manager* for further processing within the SALSA framework.

### D. Context Aggregation and Inference

The last step in context processing takes place in the third layer named *context aggregation and inference*. This is appropriate for applications that need enhanced context attributes. These mechanisms mostly map the meaning of multiple context attributes describing a situation into a new context attribute based on certain assumptions and conditions. We use a simple but practical approach for formalizing these in our framework based on a rule engine. The advantages of a rule engine are efficient evaluation of rules and support for the dynamic definition, refinement and extension of rules without the need to re-compile or re-install the service implementation. Since the implementation is still in a rudimentary form we will give only a simple example in this paper.

```
if (temperature > 25.9 AND skyconditions == sunny){
  weatherConditions=goodWeather
} else {
  weatherConditions=badWeather
}
```

The above rule is represented in a common programming language style and needs to be adapted to the rule engine's language. It takes two weather attributes and aggregates them to infer a new context attribute representing the weather condition at a higher-level.

## V. Context Matching

The final step in our approach for context-sensitive service discovery is the matching of the pre-processed context set against services represented by their service descriptions. In this section we first present the basic ideas behind our context matching approach, and then we provide a description of the service transformation mechanism and the intrinsic step of context matching.

### A. Service Discovery

After the mobile client has issued a search request and the pre-processing of the initial context set has been finished, the next step in our context-sensitive service discovery process is the matching of context information with potential service descriptions. This is handled by a two-step matching process. As previously mentioned in this paper, a search request in SALSA always consists of explicit and implicit parameters. In the first step of the matching process, the explicit parameters are used in an XPath query to pre-filter service descriptions that contain the specified properties. The second step uses the implicitly delivered context set and the pre-filtered service descriptions in a process that we refer to as context matching in the SALSA framework.

Our analysis of user requirements indicated that one of the main concerns of users related to mobile applications is privacy [26]. Therefore, we developed a context matching approach that provides an option to maintain the privacy of the user's context. As introduced in Section III, in the SALSA framework the mobile user is able to label context attributes as "*private*", so that they will not be sent to the server in search requests. The user can also set certain context attributes to be "*blurred*". For example, the geographical position may be set to be artificially blurred to make it less accurate. Not all context attributes can be blurred, such as the languages spoken by a mobile user as inferred from the user profile. The standard option for context attributes is "*public*" where the context attribute is submitted in a search request with exactly the value that has been determined.

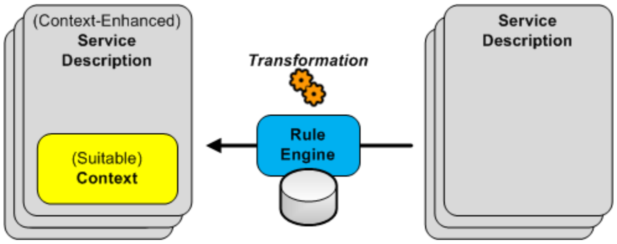


Figure 10. Service Transformation

Our approach to context matching is based on a transformation mechanism as illustrated in Figure 10. It uses rules to transform each service description that matched in the first step into a contextual representation. This context set represents the optimal context that is most suitable for the service under consideration. After the transformation, the context set is embedded within the service description.

Given the requirements for privacy, the main advantage of this approach is that context attributes that have been configured as private can still be used locally with the same context matching approach on the mobile client. This filtering and personalization is performed using the list of service descriptions returned by the SDS. Since each description contains its own transformed context set, the *Context Manager* in the client framework is able to apply context matching locally. Another major advantage of this approach is its flexibility with respect to rule definition, where rules for transformation can be added, changed and deleted dynamically at anytime and in an easy manner. Using this mechanism, the privacy of context attributes can be preserved since the service provider who receives the search request is not aware of the exact context of the mobile user, but is still able to deliver valuable search results [26].

### B. Transformation of Service Descriptios

In this subsection we present our approach to service transformation. Before going into detail, we start by illustrating our approach using the previously introduced context-sensitive gastronomy guide as an example service. The gastronomy guide is a specialized service that allows gastronomy places to register their real-world (business) service with information that is mapped to a description defined and supplied by the gastronomy guide service provider. This description schema is based on the core service schema presented previously in Section II and a domain-specific extension to describe the special properties of gastronomy places.

For example, in a service description, the domain specific fact "*outdoorSeating*" indicates that a gastronomy place offers seating outdoors. This fact may be captured by the following rule (presented in a simplified notation).

```
if (outdoorSeating) {
   weatherconditions=goodWeather
}
```

The above rule is applied for each service description that is returned after matching the explicitly defined properties of the mobile user's search request with the service descriptions in the gastronomy guide SDS registry. If the fact "*outdoorSeating*" is contained in the service description, the rule is applied and a context attribute "*weatherConditions*" is set to the defined value.

As a second example, we may define a rule based on the type of gastronomy place.

```
if (FastFoodRestaurant) {
      freeTimeMin=20
  }
else if (Restaurant) {
      freeTimeMin=60
}
else if (Café) {
      freeTimeMin=30 {
}
```

Depending on the type, we set the context attribute "*freetime*" to a certain value depending on the previous rule specification (e.g. a fast food restaurant or a café requires less free time than a regular restaurant). Applying all specified rules to the service descriptions, the transformed facts which are represented as context attributes are added to the context set that is embedded into the service description and used for later context matching.

In the server framework we realized this approach using Drools [27] which is a Java-based rule engine that supports the description of rules using XML. Following the principles of Drools, or of rule engines in general, each rule contains a condition (left hand side) and a conclusion (right hand side). When a certain condition is true (e.g. a certain fact has been discovered in the service description), the rule is triggered and the associated action is applied. Drools allows the conditions and conclusions within the left and right hand parts of a rule to be defined in two ways, either using Java statements or XSL style sheets.

We have identified two different kinds of rules, static rules and dynamic rules, which can be applied in our transformation approach. The static rules are triggered by facts that are elements of the core service description. We have therefore defined a set of standard rules and routines for the transformation process using the Drools Java style, which could also be replaced at any time by the XSLT templates style. Dynamic rules, on the other hand, are triggered by facts contained in the domain-specific extensions of service descriptions as introduced in the examples of this section. For this kind of rule we have defined standard transformations to pre-defined context data types. Thus, service providers who want to implement context-sensitive services using the SALSA framework only needs to map facts to a standard transformation with a condition and a conclusion. These rules are called dynamic rules, since they can be changed, refined and deleted anytime.

The presented transformation approach applies the same context representation used in context processing using the same data types and namespaces. This is a prerequisite for the context matching approach that will be presented in the next subsection.

### C. *Context Matching*

The final task in our context-sensitive service discovery approach is the matching of the client-delivered and pre-processed context set to the server-side transformed service descriptions, each containing its optimal context set. The process of context matching iterates over all pre-filtered service descriptions, extracts their optimal context set and iterates over each contained context attribute. For each context attribute, the data type and the namespace are extracted and the context matcher iterates over the client-delivered context set and searches for context attributes from both descriptions that have the same data type and namespace. For each equal context attribute the context matcher applies a predefined matching routine that is defined for each context data type.

The result of this context matching process is a list of services ordered and ranked based on the degree to which context attributes match. To transform service descriptions into the contextual representation, the service provider may choose which context attributes should be mandatory and which should be optional for matching. If a context attribute is marked as optional, then the matching degree is calculated based on the matching of optional attributes. If a context attribute is marked as mandatory, then one non-matching mandatory context attribute may lead to a matching degree of zero. Note that context attributes are only matched if they appear in both kinds of descriptions. If a mandatory context attribute does not appear in the client-delivered context set it is not evaluated in the matching process, since it might appear locally as a context attribute to be matched locally on the client. Further mechanisms can be applied at the client side using personalization based on preferences and previously analyzed user behavior to further refine the choice of services [26].

In Section II, we mentioned that our context model and schema may be extended with new context data types if context attributes are required that cannot be represented by an already available context data type. If a new context data type is added to the schema, a new context matching routine needs to be implemented to support context matching for the respective type.

## VI. RELATED WORK

Since Schilit et al. initiated research on context-aware computing in 1993 starting with their PARCTab project [28] various other researchers have also focused on the subject of context-sensitive service discovery. However, in the early days, many research projects focused exclusively on context-sensitive service discovery related to hardware, like near-by printers or other low-level services. In the following we give a general overview of work related to service discovery using context and to work that relates to context frameworks.

The CB-Sec project [29] introduces an architecture that focuses on the discovery of Web services using context. To this end, a service description schema was developed that includes constraints, requirements and context functions that are used by a brokering agent to evaluate, filter and rank services that best fit the conditions represented by a specific context. Context is collected by the context gatherer that receives contextual information from software and hardware sensors and is stored over time in the context data base that is available to the whole system. In the CB-Sec project, the context matching process evaluates for each specified context functions if a service is suitable for the requestor at the time of the request. Our approach allows a similar specification of the service's optimal context, but with the advantage of additional context matching on the mobile client under consideration of context privacy.

In [30], Kuck and Reichartz present an approach for the context-sensitive discovery of Web services based on the matching of the user's context and enhanced service descriptions that are stored in a UDDI repository with additional information. Their service descriptions contain information inferred from the syntactical and textual contents of WSDL descriptions as well as feedback information, e.g. the context at the time of service recommendation. Unlike our work, however, this work is restricted to the context-sensitive service discovery of Web services.

In the COSS approach [31], ontologies are used for the description of context attributes and services. Service advertisements and requests are represented as documents, and service requests include attributes defined by the user. An attribute like "nearby" is enhanced by rules that are evaluated during the matching process, for example the user's location is within a certain distance to the service's location. In other work of this research group, the WASP project, a service platform for mobile context-aware applications [32] was developed. In both approaches the rules are defined as actions that are executed if the criterion for a certain context attribute becomes true. The framework is intended more to support context-sensitive applications in general, while our approach directly targets context-sensitive service discovery.

In [33], Korpipää presents the Context Management Framework (CMF) that was created especially for context-sensitive mobile applications. The context manager is the main component of the CMF's. Applications can use the context manager to register for context sources to be able to receive and update their values. The context recognition services can infer new context values from low-level context-sources, similar to our approach of context resolution. The context model applied in CMF is based on RDF. Unlike our approach, however, the CMF offers no mechanisms to ensure the privacy of context information.

Other work, like the NEXUS project [34], the SOCAM architecture [35] or the DAIDALOS project [36] also present and implement architectures for a context framework. Each of these introduces a different model and representation format for context as well as different components and processing. Compared to the architecture in this paper, they offer a more generic approach for the development of context-sensitive applications, while our approach focuses on the context-sensitive discovery of services. Apart from the DAIDALOS project, none of these consider the privacy of context information as we do.

## VII. Conclusions

In this paper we have introduced a generic, component-based framework that enables the development of (a) services that support context-sensitive service discovery, and (b) context-aware mobile applications that make use of these services. To build the SALSA framework, we introduced a simple model and representation format for context as well as an extensible and flexible description schema for services that have various advantages. Through the clear separation of context processing layers, we have defined a context processing architecture that can be embedded within mobile clients and within services as needed. The client framework supports the handling of context sources and the management of context attributes using a user-friendly mechanism to configure context attributes with different permissions as introduced in Section III. The client framework also introduces a flexible architecture that offers the dynamic reconfiguration and integration of the provider's service components for execution at run-time.

Within the server framework we have introduced several mechanisms for context handling. The step of context resolution allows service providers to obtain as much context information as possible without the need for complex implementation work. Our novel approach for context matching, based on the transformation of service descriptions into a contextual representation, is a key advantage of our approach. It (a) offers the possibility of dynamic rule declaration for service providers, and (b) allows context matching on the mobile client using "*private*" context attributes and the embedded context representation within service descriptions. Using this approach the privacy of the user's context can be retained.

By implementing the presented prototype scenario with example services (context-sensitive gastronomy guide, event guide, tourist guide and a bargain hunter), we have shown that the approach can be applied and extended in a simple way. Arbitrary context data types can be defined as needed, the service description schema can be extended within the domain-specific part and different kinds of applications that apply context-sensitive service discovery can be created. Furthermore, our prototype implementation for mobile commerce applications shows that services can easily be made context-sensitive to provide high precision and personalized service retrieval. This also minimizes the mobile user's effort in service discovery and opens new revenue chains for service as well as for the context providers.

In future work, we are planning to extend the SALSA framework to support a more complex model of context that better supports the inference and aggregation layer in context handling. We also plan to implement example applications for mobile business to show that our approach can be applied in other application fields in a similar way.

## ACKNOWLEDGMENT

This work was supported by a grant from the Landesstiftung Baden-Wuerttemberg.

## REFERENCES

[1] C. Atkinson, P. Bostan, and T. Butter, “Context-Sensitive Service Discovery for Mobile Commerce Applications,” in Proc. of the 4th International Conference on Wireless and Mobile Communications (ICWMC 08), IEEE Computer Society, Washington, DC, 2008, pp. 352-357, doi= 10.1109/ICWMC.2008.33.

[2] M. Baldauf, S. Dustdar and F. Rosenberg, “A survey on context-aware systems,” Int. Journal of Ad Hoc and Ubiquitous Computing, vol. 2 (4), 2007, pp.263–277, doi=10.1504/IJAHUC.2007.014070.

[3] A. Zimmermann, A. Lorenz, and R. Oppermann, “An Operational Definition of Context,” in Proc. of the conference CONTEXT 2007, LNAI4635, Springer-Verlag Berlin Heidelberg, August 2007, pp. 558-571, doi=10.1007/978-3-540-74255-5_42.

[4] The Merriam-Webster Free Online Dictionary, http://www.merriam-webster.com/dictionary/context

[5] The Free On-line Dictionary of Computing, http://foldoc.org/context

[6] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in Proc. of the Workshop on Mobile Computing Systems and Applications, 1994, pp. 85-90, doi=10.1.1.37.9380.

[7] P. J. Brown, "The Stick-e Document: a Framework for Creating Context-aware Applications," in Special Issue: Proc. of the Sixth International Conference on Electronic Publishing, Document Manipulation and Typography, Palo Alto, A. Brown, A. Brüggemenn-Klein, and A. Feng, Eds., vol. 8, no. 2&3, John Wiley and Sons, June 1996, pp. 259-272.

[8] P. J. Brown, J. D. Bovey, and X. Chen, "Context-aware Applications: from the Laboratory to the Marketplace," Personal Communications, IEEE [see also IEEE Wireless Communications], vol. 4, no. 5, 1997, pp. 58-64.

[9] N. Ryan, J. Pascoe, and D. Morse, “Enhanced Reality Fieldwork: The Context-aware Archaeological Assistant,” Computer Applications in Archaeology. Oxford, 1997.

[10] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a Better Understanding of Context and Context-awareness," in Proc. of the 1st International symposium on Handheld and Ubiquitous Computing (HUC 99), London, UK: Springer-Verlag, 1999, pp. 304-307.

[11] J. Pascoe, “Adding Generic Contextual Capabilities to Wearable Computers,” in Proc. of 2nd International Symposium on Wearable Computers, October 1998, pp. 92-99.

[12] H. Lieberman and T. Selker, "Out of context: Computer systems that adapt to, and learn from, context," IBM Systems Journal, vol. 39, 2000, pp. 617-632.

[13] T. Strang and C. L. Popien, "A Context Modeling Survey," in Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, September 2004, doi=10.1.1.2.2060.

[14] P. Dockhorn Costa, G. Guizzardi, J.P.A. Almeida, L. Ferreira Pires, M. van Sinderen, “Situations in Conceptual Modeling of Context,” in Proc. of the 2nd International Workshop on Vocabularies, Ontologies and Rules for The Enterprise (VORTE’06), Hong Kong, 2006, doi=10.1109/EDOCW.2006.62.

[15] The OWL Service Coalition, “OWL-S: Semantic Markup for Web Services,” http://www.daml.org/services/owl-s/1.1/overview, 2004.

[16] UNSPSC – United Nations Standard Products and Service Codes, http://www.unspsc.org/download.aspx.

[17] NAICS – North American Industry Classification Standard, http://www.census.gov/epcd/www/naics.html.

[18] M. Aleksy, C. Atkinson, P. Bostan, T. Butter and M. Schader, “Interaction Styles for Service Discovery in Mobile Business Applications,” in Proc. of the 17th International Workshop on Database and Expert Systems Applications (DEXA), IEEE Computer Society, 2006, pp. 60–65. doi= 10.1109/DEXA.2006.75.

[19] Natix – A native database system for XML, http://pi3.informatik.uni-mannheim.de/~moer/natix.html.

[20] M. Brantner, S. Helmer, C. Kanne, G. Moerkotte, “Full-Fledged Algebraic XPath Processing in Natix,” in Proc. of the 21st International Conference on Data Engineering (ICDE), IEEE Computer Society, Washington, DC, 2005, pp. 705-716, doi=10.1109/ICDE.2005.69.

[21] M. Aleksy, T. Butter, and M. Schader, “Architecture for the development of context-sensitive mobile applications” in Int. Journal of Mobile Information Systems, vol. 4,no.2 , April 2008, pp. 105-117.

[22] Object Management Group, “The Common Object Request Broker: Architecture and Specification. Version 3.0.3,” OMG Technical Document Number formal/04-03-01, 2004, ftp://ftp.omg.org/pub/docs/formal/04-03-01.pdf.

[23] M. Gudgin, M. Hadley, N. Mendelsohn, J.J. Moreau, H.F. Nielsen, A. Karmarkar and Y. Lafon, “W3C SOAP Version 1.2 Part 1: Messaging Framework (Second Edition),” W3C Recommendation, April 2007, http://www.w3.org/TR/soap12-part1/.

[24] T. Butter, M. Aleksy, P. Bostan, M. Schader, “Context-aware User Interface Framework for Mobile Applications,” in Proc. of the 27th international Conference on Distributed Computing Systems Workshops (ICDCSW), IEEE Computer Society, Washington, DC, p. 39, 2007, doi=10.1109/ICDCSW.2007.31.

[25] T. King, S. Kopf, T. Haenselmann, C. Lubberger, and W. Effelsberg, "Compass: A Probabilistic Indoor Positioning System Based on 802.11 and Digital Compasses," in Proceedings of the First ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH), Los Angeles, CA, USA, September 2006.

[26] T. Butter, S. Deibert, and F. Rothlauf, "Using Private and Public Context - An Approach for Mobile Discovery and Search Services," In: T. Kirste, B. König-Ries, K. Pousttchi, and K. Turowski (eds): Mobile Informationssysteme - Potentiale, Hindernisse, Einsatz, pp. 144-155, Bonner Köllen Verlag, Bonn, 2006.

[27] Drools – Rete OO, Java Rule Engine, http://jboss.org/drools/documentation.html.

[28] B.N. Schilit, M.M. Theimer, and B.B. Welch, “Customizing Mobile Applications,” in Proceedings of USENIX Symposium on Mobile & Location-Independent Computing, USENIX Association, August 1993, pp 129–138.

[29] S. K. Mostefaoui and B. Hirsbrunner, “Context Aware Service Provisioning,” in Proc. of the IEEE/ACS International Conference on Pervasive Services (ICPS), IEEE Computer Society, Washington, DC, 2004, pp. 71-80, doi=10.1109/ICPS.2004.13.

[30] J. Kuck and F. Reichartz, “A collaborative and feature-based approach to Context-Sensitive Service Discovery,” in Proc. of 5th WWW Workshop on Emerging Applications for Wireless and Mobile Access (MobEA), 2007.

[31] T.H.F Broens, S.V Pokraev, S.V. M. van Sinderen, J. Koolwaaij, and P. Dockhorn Costa, “Context-aware, Ontology-based, Service Discovery,” in: European Symposium on Ambient Intelligence (EUSAI), Lecture Notes in Computer Science 3295, Springer, 2004, pp. 72-83.

[32] S. Pokraev et al., “Service Platform for Rapid Development and Deployment of Context-Aware, Mobile Applications,” in Proc. of the IEEE International Conference on Web Services (ICWS), IEEE Computer Society, Washington, DC, 2005, pp. 639-646, doi=10.1109/ICWS.2005.106.

[33] P. Korpipaa, J. Mantyjarvi, J. Kela, H. Keranen, and E. Malm, “Managing Context Information in Mobile Devices,” in IEEE Pervasive Computing, vol. 2, no. 3, 2003, pp. 42-51.

[34] F. Dürr, N. Hönle, D. Nicklas, C. Becker, K. Rothermel, "Nexus - A Platform for Context-aware Applications," in Proc of the GI-Fachgespräch "Ortsbezogene Dienste", 2004.

[35] T. Gu, H.Wang, H. K. Pung, and D. Q. Zhang, "An Ontology-based Context Model in Intelligent Environments," in Proc. of Communication Networks and Distributed Systems Modeling and Simulation Conference, 2004, pp. 270-275.

[36] M. Strimpakou, I. Roussaki, and M. E. Anagnostou, "A Context Ontology for Pervasive Service Provision," in Proc. of International Conference on Advanced Information Networking and Applications (AINA), 2006, pp. 775–779, doi=10.1109/AINA.2006.15.

# VPIN:
# An Event Based Knowledge Inference for a User Centric Information System

Netzahualcoyotl Ornelas [2], Noëmie Simoni[1], Chunyang Yin[1], Antoine Boutignon[3]
[1]*Institut TELECOM / TELECOM ParisTech*
*46 rue Barrault, 75634 Paris Cedex 13 France*
*{simoni, yin}@telecom-paristech.fr*
[2]*L2TI - Institut Galilee - University Paris 13*
*99 Avenue J-B Clément, 93430 Villetaneuse France*
*netza@univ-paris13.fr*
[3]*Société Française du Radiotéléphone (SFR)/DRI*
*40-42, quai du Point du Jour, 92659 Boulogne Billancourt Cedex, France*
*antoine.boutignon@sfr.com*

## Abstract

*In the Next Generation Network, the telecommunication environment becomes more and more heterogeneous and mobile (concerning user, terminal, network, service). User must be the central point of the consideration due to this ambient and dynamic network environment. The future information systems should be user oriented and have to perform well with ever changing of the telecom environment due to mobility and usage. Unfortunately, the existing solutions are traditionally inspired by the methodologies which are object-oriented, application-related, which leads a semantic gap between the user new requirements and the real environment.*

*In this paper, to reduce the mentioned semantic gap, we propose to adopt a vision of User Centric and a common information model as the departure point. We propose a persistent VPIN (Virtual Private Information Network) as a knowledge base applying this information model. Therefore, VPIN is able to take into account all the heterogeneous elements in the NGN context and be independent of any application. To mange the dynamicity of the ambient environment, we propose that the VPIN acts as an inference through the events handing. To insure that the events can be handled automatically, we propose alternative ways to manage and update the VPIN.*

*Keywords: User-centric approach, information model, QoS, State, inference*

## 1 Introduction

Next Generation Networks offer the telecommunication consumers with numerous business benefits as well as the opportunity to work faster and smarter. In order to take advantage of these benefits and opportunities, we should take into account the technologies evolution and the application domain enlargement. The technologies evolution brings us the heterogeneity. Nowadays, although the core network rests the same, the access to the backbone is differed from wired to wireless ways. Especially for the wireless access, we have now not only different ways (WiFi, Bluetooth, IrDA) to construct a local network, but also the ways (GPRS, UMTS, Wimax) to have a very large coverage. The heterogeneity exists equally in the terminal types as PC, laptop, PDA, Mobile phone etc. As a result, the services are thus also heterogeneous due to different supports and providers.

As we mentioned, NGN has also brought us an enlarged application domain, which enables the users to move during the utilization of the network and the services. According to the different actors in the NGN, we resume that there are generally four types of mobility:

First, we can consider the user mobility, which is the capacity of a user to change of terminal. For example when the user has different types of devices, he desires to use his PDA to check his e-mails and after, as he moves to another room, he uses his personal computer to continue checking his e-mails. Besides this mobility, there is the terminal mobility. It refers to the capacity of a terminal to change of access networks. For example, when a user is having a conversation using his cell phone and the user is driving in his car, in that case, the cell phone cross several access networks, and there is not an interruption in the communication. An additional type of mobility is the network mobility. It permits a network equipment to move of place, allowing the communication with different network equipments that are in constant movement too. An example is a vehicular network. It means, that any type of vehicle that contains the same router equipment, allows communicating with the different vehicles located nearby and that are constantly in movement. And in last, the service mobility, which is the capacity of a service to change of component when a component that provides a service stop working. We can imagine a server of streaming that diffuses a Radio station, therefore, if that server stops working, there could be another server that retakes the same diffusion in real time, as a result, the service continue to be delivered.

At the apparition of personal computer systems, there was an obligation for the user to have a minimum technical knowledge of the system installed in order to use it and take advantage of it. Nevertheless, it's not an easy task for user to control every kind of Operating System in the market (Windows, Linux, and Mac, etc.), as a result, the utilization of the system by the user was minimum compared to the capacity offered by the personal computer.

However, the utilization of systems resources started to be insufficient to fulfil the user requirements. All capabilities in the resources were integrated in the system; consequently, the access feasibility to these resources can be improved through a network. For example, when stocking a big volume of information, it's better to do it through a disk available in the network whose capacities are bigger than in the user system. In addition, the access to this information can be done by any computer even of different user location. This optimizes the use of this resource without the necessity to change the computer hard disk when this is already almost at 100 percent of utilization.

However, this new way to access resources through a network, implies end user to learn some additional, and sometimes, complicated process to achieve the use of a service. Therefore, these technical skills that must be learned by end users are not easy to accomplish as we think; especially when network applications evolve quickly and a constant apprenticeship is needed.

Although the access to resources by the network is a viable solution, end users felt an important limitation to adapt the use of their services, according to the technical specifications, and configuration of the network. In fact, a limitation in the network could be, when end user wishes to watch a TV program with his personal computer. Most of the time, the session establishment is achieved, but sometimes quality in the reception is not good enough as the user wished. More over, if the user wants to move to another place and his computer is connected by an Ethernet cable, he is obligated to disconnect the cable, and to reconnect via other access network such as the Wi-Fi for example. Besides, it could happen that the reception of the TV program is not possible due to some interference in the Wi-Fi access network. Therefore, a cut of his service is introduced. Otherwise, some user skills are required to achieve this procedure, and re-establish the reception of the TV program.

Hence, telecommunications operators and service providers realized the difficult encountered by the end users. The complexities of those user problems are not only when accessing his services, but also when trying to adapt them according their preferences. Therefore, from now, operators and service providers, consider the user as the main point of interest in the offer of new services. It means, they prefer to better know what the end users desire to access their services than the follow the technological tendencies.

In fact, that's the end-users who desire access to all types of services; for example, the user own services (e.g. check e-mail) or services that are offered according to his location (e.g. receive announces of reduction in his PDA when he is in a supermarket). In addition, it is always the user who desires access to these services with any kind of device available around him at any time. Beside these requirements, the user is usually confronted to some technical barriers that are obstacles to access his services.

It is for this reason, in order to fulfil those user needs, a new approach in the technological conception should be considered. Instead of the design vision of System Centric or Application Centric, we will take the vision of User Centric as the centre of the global architecture. This could change the way the technological evolution was carried out until now. In this new approach, user needs could make the creation of new technologies and a new vision for service delivery, trying to respect, an adaptability of those services according to user location and his preferences.

Anyway, user preferences are more difficult to be achieved as we thought. They can change in a short period of time, according to the user context and they needs. One of these user preferences could be the cost. For example; when a user wants to hire the cheapest Internet service connection at home, he will check the offers and the packets of the different services providers. The selected operator will be the one that offer a major of services with the cheapest price in the market. Another type of user preferences might be technical. Based on the cost example, we can imagine that after the selection of the Internet service provider, the user wishes to choose the type of connection (e.g.

Ethernet, Wi-Fi), besides, maybe the user desires to use another device instead of his personal computer (e.g. PDA). Or even, when user accesses his services, in order to be in communication with his family, he prefers employ a Videoconference program installed in his computer, instead of using the e-mail.

At last, we can consider the usage preferences. They consist in the different configurations that are supported in a component. That is, once the user decided to watch the TV on his computer, some adaptations can be done in the quality of the image (pixels, image in colour or black and white) or the quality of the sound (e.g. sound stereo, etc.).

All of these user preferences could help to accomplish some user needs; but not all of them, as we need to take into account that the user is in constant movement. It’s important to recall that these different types of mobility can produce a user-session interruption.

In order to avoid user-session interruptions, (due to the different types of mobility recently explained), we believe that it would be useful to obtain a collection of necessary information according to a specific user, i.e. a user Information System.

The expected Information System could allow a user having a global vision of all available resources no matter which providers support them or where they are placed. This could make user possible to check their capacities as well as what can be offered by a resource.

Actually, this user side information system is expected to be a knowledge system which contains necessary and sufficient information for a dynamic and sophisticate environment, as the behavior related information, user preferences etc to enable the decision to be taken according to the contained information.

This Information System should also be an application-independent knowledge inference managing the user side applying rules’ information and maintaining itself by responding automatically and alternatively the dynamic mobile telecom world.

Unfortunately, the existing solutions are traditionally inspired by the methodologies which are object-oriented, application-related, which leads a semantic gap between the user new requirements and the real environment.

To have a clear idea of which kind of User Information System we should conceive, some questions need to be answered, as:

a) How to define the “relevant and decisional information” for the user information system? b) How to structure this information in the best possible way to let the Information System be application-independent? c) How to manage the information most efficiently? What are the automatic processes in order to maintain the information system dynamic? d) When a change occurs in the IS, how to update the whole IS automatically in order to keep in touch with the real time world?

To answer the first question, we consider that the relevant information permits to have a global vision of the behavior of all the components, such as terminals, networks and services. This information is present at any place and any time. We believe that the decisional information implies that we must assure the provision of this relevant information at the right time and the right place; that is not only at session establishment but also during the session itself. This could introduce a dynamic session management requirement.

Responding the second question that relates the structure information, we consider that it must be provided and structured according to the real world’s vision, which means the different types of mobility and heterogeneity aspects, while respecting the user’s requirements (preferences, QoS), in order to make the adequate decisions.

And to answer the last two questions, we think that the automatic process starts when an event is produced, this event could be a change of a component’s behavior. This means that an action must be taken in order to fulfill the self-management and offer services transparently (without interruptions).

In order to establish and maintain a user Information System where the heterogeneity (terminal, network, and service) and mobility (user, terminal, network and service) are omnipresent, we consider necessary to have the necessary, sufficient and decisional information (knowledge base), as well as the actions to take when an event is produced (change of the QoS in a component), in order to make the correct decision autonomously. The fact of taking correct decision autonomously makes a transparency for the end user when using his services.

This transparency must be valid no matter the user location, the utilization of terminals and networks that are to his disposition.

Hence, we propose a VPIN (Virtual Private Information Network) [1] based on an information model. The Information Model represents the real world by an informational structure, which is semantic of the new environment. VPIN provides a complete information image of the components of the user’s system (including QoS information). The persistent information in the VPIN enables also the separation of the user system and the user’s different applications, which helps to manage user sessions in ambient networks.

This paper is organized as follows. We present in Section 2 the related work, which allows us to take into account the recent solutions proposed until now. In

Section 3, we introduce the background of our work to explain the architectural model derived form Meta-model NLN (Node, Link, and Network), as well as our QoS based information model, on which we base our proposition. Afterwards, in Section 4, we propose organizationally the VPIN, which possesses the adequate information for the user. In section 5, the functional dimension is detailed by the identification of the events [2][3][4] the inference functional specifications. In order to examine the feasibility of our concept, in Section 6, we describe an experimentation to illustrate our proposition. And finally, Section 7 presents the conclusions and perspectives for future work.

## 2 Related work

Before presenting our proposals concerning the management information, we analyze the content of some related structural proposals, which concern different *information models*.

Currently, in Information Technologies environments, some efforts have been done in standards works to consider the *informational models* as the representation of information about entities to be managed. These works are more interested in the representation of information related with the different entities to be managed but they don't have the same abstraction level, or the same coverage. We find the GNIM for the Networks elements, the CIM for the enterprise elements, the WBEM for allowing a uniform access through the web, the DEN-ng for the integration of the context constraints, and finally, the SID, which the aim is to cover all the information system from the strategy, the services, to billing.

The GNIM (Generic Network Information Model) [5] is a recommendation (M.3100) proposed by the ITU-T [6] through the TMN (Telecommunications Management Network). It specifies a generic network management information model for the management of telecommunications networks. It consists of a set of abstract or common managed object classes and their properties that may be specialized to support the management of various technologies, architectures and services. This model can be seen as a set of reusable managed object classes that may be adapted to support the management of various telecommunications networks. As is a generic model, it allows being independent of a technology as ATM (Asynchronous Transfer Mode), Frame Relay, SDH, PDH, PSTN, etc.

The GNIM defines four levels of management. These levels are the Business, Service, Network and Network Element (NE) management levels. Note that the service and network management level aspects are weak.

The CIM (Common Information Model) [7] is proposed by the DMTF (Distributed Management Task Force) [8], which is an open standard that defines how to manage elements and their relationships in an IT environment. The elements are represented as a common set of objects, which allow consistent management independent of their manufacturer or provider.

CIM is an information model that presents a conceptual view of a managed environment. A first goal of CIM is to unify and extend the existing management standards such as SNMP, DMI, CMIP, etc. by using and object-oriented design. The second goal of CIM is enabling to model all the different aspects in a managed environment. These aspects are represented in the "Common Models" created to address system, device, network, user and application aspects. The CIM is comprised of a specification and a schema. The CIM Specification defines the details for integration with other management models, while the CIM Schema provides the actual model descriptions. The CIM Schema captures the notions, which are applicable to all common areas of management, independent of implementations.

CIM itself is structured into three distinct layers:

- A "Core model" that captures the different notions, which are applicable to all areas of management.
- A "Common model" to capture the notions that are common to particular management area (systems, applications, networks and devices), but independent of a particular technology or implementation. The Core and Common models together are referred as the CIM schema.
- The "Extension schemas" to represent the technology-specific extensions of the Common model, for example in operating systems (UNIX or Microsoft Windows).

Thus, CIM is a conceptual model that is not bound to a particular implementation. This allows it to be used to exchange management information in a variety of ways.

One of fundamental aspect in CIM is the ability to exchange information between management applications. The current mechanism for exchanging management information is the Management Object Format (MOF) [9]. MOF defines the meta-schema (a formal definition of the model) to be used to represent the syntaxes and semantics aspects of the model. The meta-schema defines the basic object-oriented concepts: classes, relationships, properties, methods, operations, inheritance, associations, objects, cardinality and polymorphism. A CIM-capable system must be able to import and export properly formed MOF constructs.

The Web-Based Enterprise Management (WBEM) [10] proposed by the DMTF (Distributed Management Task Force) is a set of management and Internet standard technologies developed to unify the management of distributed computing environments. WBEM provides the ability for the industry to deliver a well-integrated set of standard-based management tools, facilitating the exchange of data across otherwise disparate technologies and platforms. It defines a standard common model (i.e., description) and protocols (i.e., interface) for monitoring and controlling resources from diverse sources.

An important part of WBEM is the Common Information Model (CIM), a standard for defining device and application characteristics so that system and network administrators and management programs are able to control devices and applications from different manufacturers or sources in the same way. WBEM standards provide a Web-based approach for exchanging CIM data across different technologies and platforms. CIM data is encoded using Extensible Markup Language (XML) [11] and usually transmitted between WBEM servers and clients using the Internet's Hypertext Transfer Protocol (HTTP).

The WBEM is designed to be extensible, allowing new applications, devices, and operating systems to be specified in the future. Open-source implementations of WBEM are available from several vendors, including OpenPegasus, OpenWBEM, and WBEMsource. WBEM is said to be particularly appropriate for storage networking, grid computing, utility computing, and Web services.

The Directory Enabled Network (DEN) [12] initiative is designed to provide the building blocks for more intelligent management by mapping concepts from CIM (such as systems, services and policies) to a directory, and integrating this information with other WBEM elements in the management infrastructure. This utilizes existing user and enterprise-wide data already present in a company's directory, empowers end-to-end services, and supports distributed network-wide service creation, provisioning and management.

The use of CIM in defining a directory schema enables consistent schema for, and a common understanding of, directory information. Common schema and semantics are especially important when defining and decomposing platform-neutral, high-level policies.

The DMTF DEN Special Interest working group is focused on communicating the benefits of DEN as a key component of the DMTF's management standards. It is working at two levels, as the first one is to use a directory FIRST to "direct" management clients to relevant services, and to hold a subset of management data; the second level is to specify the directory schema (LDAP mappings) for DMTF's CIM Version 2.5 and later releases. Specific modeling and mapping efforts are addressed in the DMTF's LDAP, Network, Policy, and User and Security Working Groups.

The DEN-ng is designed to provide a rich and extensible classification of managed entities. It overcomes UML limitations by linking to ontology's, as well as specifies and design autonomic architectures and generates code dynamically for managing autonomic systems. The DEN-ng models the aspects of a ManagedEntity; other models represent a ManagedEntity as an atomic object.

The DEN-ng object-oriented information model provides a cohesive, comprehensive and extensible means to categorize and represent things of interest in a managed environment, including users, policies, processes, routers, services, and anything else that needs to be represented in a common way to facilitate its representation and management.

The DEN-ng information model defines the static and dynamic characteristics and behavior of these managed entities as independent of any specific type of repository, software usage, or access protocol. Note that the explicit use of dynamic models differentiates it from other current management efforts.

The DEN-ng uses dynamic models to represent the life cycle of managed elements. Many different stakeholders are required to work together to build a product. However, they all have different perspectives on how the product works.

This means that one concept might mean different things to different people. For example, when a business analyst looks at an SLA, that person thinks of contractual obligations and different options for realizing revenue.

The SID (Shared Information Data) [13] is suggested by the NGOSS (New Generation Operations Systems and Software) [14], which provides a "common language" for software providers and integrators to describe information management. SID is used to run business processes and enable reporting; companies can create rules that describe how data must be created and used in SID.

The SID is an object model, which uses Unified Modeling Language (UML) [15] to define the entities and the relationships between them, as well as the attributes and processes (termed methods) which make up the entity or object. SID provides the NGOSS “glue” in giving a representation of different views: Business view, System view, Implementation view and Deployment view. These views are necessary to ensure that business requirements can drive system design and implementation. The SID focuses on modelling network elements and services covering a business, system and implementation viewpoints.

Furthermore, some open issues exist today and concerning the implementation of services with this model. The service activation methods are not described and there is no knowledge of the network itself. Here the notion of Knowledge represent only a Database or Repository of sharable information about model, but does not give us the opportune and needed information to assure a correctly running of service in real time. This includes tools and guidance for service providers, suppliers and systems integrators, with the definition of a Business Process, Systems and Software integration "maps", and the development of an architecture and Knowledge Base or Repository of documents, models and reference code to support developers, integrators and users. Their principal goal is to provide a rapid development of flexible, low cost of ownership solutions to meet the business needs of the Internet enabled economy.

The NGOSS methodology is addressed by a Business Process Model-based Viewpoint where SID represents the key to Interoperability based on the shared of Business Process and Experience Driven. SID is considered such as the NGOSS Meta-Model for Shared Info & Data.

A consideration to be taken is to search how to have a better knowledge of a service to manage it correctly. For years, it has been gotten information through databases that storage the information required by the system, that is the case of the SNMP [16] protocol. It has been one of the first works where there is a database for monitoring the network.

Another solution is the CMDB (Configuration Management Data Base) [17]. It proposes structured databases that conform to the ITIL (Information Technology Infrastructure Library) [18] norm. It contains a repository of information for components in an Information System. This approach allows different configuration management processes to share data, but requires a lot of resources to create and maintain the integration of all the information.

For years, the human has processed information with the help of external tools (graphics, word processor, etc.); nevertheless, with the increase of volumes of information and computer technology evolution it becomes more difficult to manage and storage it. A new solution has been deployed named Data mining [19]. The goal of the Data mining approaches is to get knowledge from simple data patterns collection. The term of Data mining is often related to two processes, the knowledge discovery and the prediction. This solution is employed by enterprises desiring to know if their business model works correctly, or to find hidden patterns from data. However, this solution doesn't allow taking decisions in real time, because at the beginning there is a static collection of information and according to rules determined, we can thus obtain the knowledge desired.

All the cited works present the good efforts to provide the conceptual and informational views about the entities to be managed. Therefore, some open issues are still present concerning the methods and mechanisms to implement these models and also to deal with the deployment and dynamic maintain of services deployed. The models cited study more the functional aspects but don't include behavior and non-functional aspects, which are actually necessary to treat the end-to-end service QoS associated to a service cycle-life. In fact, what we want to have as information becomes important when is used to take decisions; the problem is to establish a knowledge base working as an inference. This knowledge base should automatically update and find the relevant decisional information to take decisions dynamically.

## 3 Background

In order to have a better comprehension of this paper's driving concepts, we present the previous works, which are finalized by the Telecom ParisTech INFRES-Lab. The lab is engaged in the dynamic service management framework design [20] [21] [22].

To be specific, the framework concerns first, the architectural models (§3.1) which give out a global and full-scale image of the existing heterogeneous environment; second, the informational structures (§3.2), which provide the basic generic structures to ease the integration of the different information.

### 3.1 Architectural Models

The focus point of our concepts is how to model the telecommunication world in order to have a generic representation. Our modeling is conceived through *Abstraction*, which is defined by universal encyclopedia to structure the data according to simplification, generalization, selection and schematization. Therefore, the proposed Meta model (Figure 1) of NLN (Node-Link-Network) includes the following three abstracted elements:

The *Node* is defined as an entity, an element that is responsible for a specific process.

The *Link* is the representation of the interaction between two nodes. It can be considered as a virtual communication channel between two ends. It designates any component offering its transfer capacities in order to provide the nodes a support of interconnection.

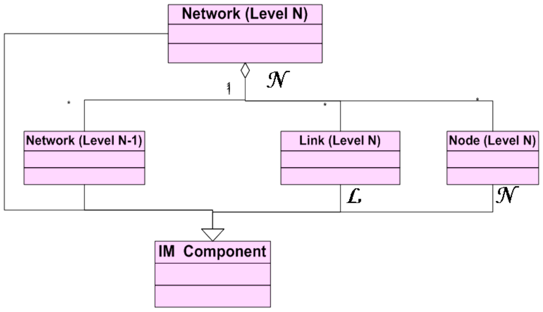


**Figure 1: Meta Model: NLN**

The *Network* is a set of nodes and links offering a global service in a transparent way. It allows the nodes and the links cooperate in order to offer a certain service. It is defined through four visibility levels. (User, Terminal, Network, and Service). [23][24].

By applying the NLN Meta-model, a visibility level can be represented as a network of components, while a link responds to a communication between two visibility levels. The Meta-model enables the modeling to represent the real world. As we can see in Figure 2, each visibility level represents a network (horizontal), which could be an *equipment network* (Figure 2-1), a *provider network* (Figure 2-2), or a *service network* (Figure 2-3). In every network, there is a behavior vision for each component. This helps when a component can't fulfill any more its SLA or the user's required QoS: a counterpart that respects those requirements can be found to replace the current component. With the help of these networks at different levels, the user's preferences and his required QoS can be satisfied (Figure 2-4).

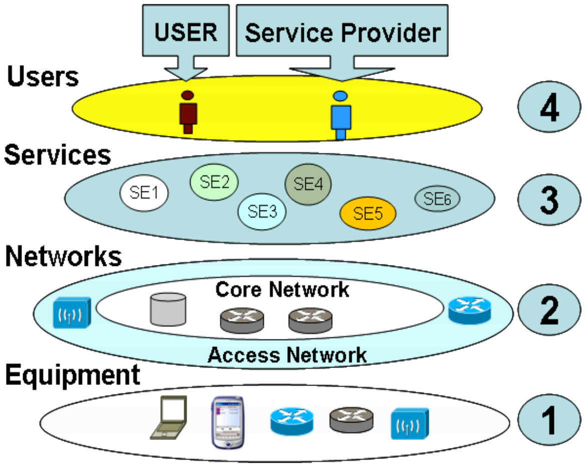


**Figure 2: Modeling vision**

Besides the horizontal view of this modeling vision, it is possible to have a vertical network composition, which allows the establishment of a session according to the user preferences.

For example, at the equipment level (Figure 2-1) and in the network of the equipment level (Figure 2-2), we can apply the user preferences using the VPIN to choose equipments and the access network in order to offer the required service, providing that the components fulfill the user's required QoS.

Facing the heterogeneous environment, the importance of this Meta-model is the decomposition of the whole telecommunication world into several abstraction levels according to the different service levels, which permits management system to be complete and thorough. Therefore, applying the Meta model at each decision level, we can thus obtain an architecture model (Figure 3), which covers the ambient context. The horizontal and vertical relationships (Figure 3 - 1) are guaranteed thanks to the composition and recursiveness features provided by our model. In fact, each level of visibility is modeled in the same way. At the service level (Figure 3 - 2), service components are managed as service overlay according to their types (SCU, SCA, SCN). Together with the virtual links, the service components are linked by the service logic to form a VPSN (Virtual Private Service Network) [20], which is essential for handling the overall service session. The user selected network from the VPSN represents a complete use centric service.

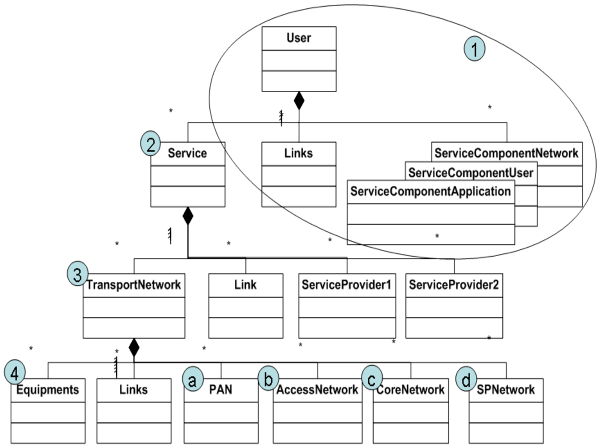


**Figure 3: Architecture Model**

The VPSN stays always on a transport network (Figure 3 - 3) according to the user's geographical location. As the services are always supported by the service providers, the latter is represented also in the model in order to have all the potential service suppliers at anywhere user arrives.

In fact, the VPSN is supposed to support the four types of mobility in the NGN (that we previously pointed out) within a single E2E connectivity:

- The user mobility, i.e. the capacity of a target user to move from one terminal to another, is supported by the PAN (Personal Access Network) node (Figure 3 - a).
- The terminal mobility, i.e. the capacity of a given terminal to move from one access network to another, is supported by the AccessNetwork node (Figure 3 -b).
- The network mobility, i.e. the fact that the access network itself is moving, it can be supported jointly by AccessNetwork node (Figure 3-b) and CoreNetwork node (Figure 3 -c), and finally,
- The service mobility, i.e. the possibility to replace a service component by another (for example, a nearby service component which is more suitable than the current remote one). This type of mobility is supported by the mode of SPNetwork (Service Provider Network) (Figure 3 -d).

Note that all these four nodes (Figure 3 - a to d) have self-management capabilities to preserve the offered service. The links between these nodes insure interactions between them to assure the service delivery to be continuous throughout the user demanded global service. The E2E service can thus be dynamically maintained by linking certain equipments in the equipment level (Figure 3 - 4), where the routers as the hosts are the equipment nodes, the cables are equipment links and the whole is the equipment network.

## 3.2 Informational Structures

For efficient representation of the real world, we must have a uniform information structure containing the relevant and synthetic information to make the right decisions at the right place at the right time. It means having both the description information of all resources, but also knowledge of the behavioral aspects, i.e., whatever is on the QoS. The informational model we defined is generic and abstract to describe any ambient resource. In order to have a dynamic management, the "Real Time Profile" (Figure 4) is instantiated in real time and it refers to an element that may belong to any management level.

The component of "Real Time Profile" will have in an instant *"t",* one of four states, these states are: *Una*vailable, *Available, Activable* and *Activated* (Figure 4-1). The state *Unavailable* means that the resource is temporary or permanently inaccessible. The state *Available* means that the resource is or can be accessible. The state *Activable* takes into account certain logical conditions, which are necessary to assure the activation of a component. For example, concerning the authentication of a user, when he introduces a correct login/password, the resource becomes *Activable*. The state *Activated* means that the resource is being used.

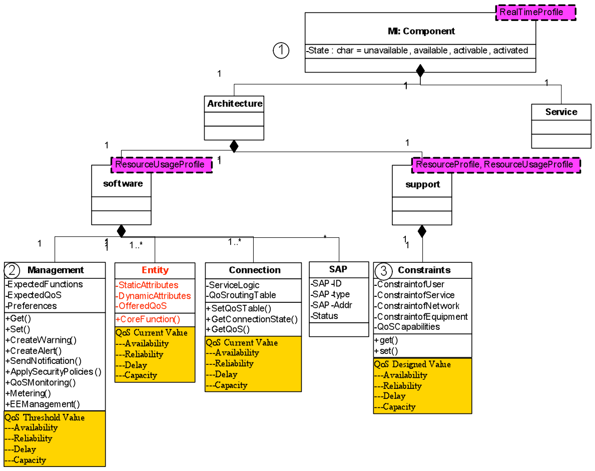


**Figure 4: Real Time Profile**

We have self-management in each resource by extracting information from the resource profile and resource usage profile [20]. Concerning the component behavior, it is necessary to have a homogeneous expression of its QoS to evaluate the end-to-end behavior. The behavior of each component is obtained by measurable QoS parameters that can be categorized according to four criteria: *Availability, Reliability, Delay*, and *Capacity*. The self-management is done according to this QoS model.

The management class (Figure 4-2) contains *QoS threshold values* (it indicates the limit of a node's normal operation or a link's normal interaction realization, in normal condition of the service exploitation and usage) concerning a required service for QoS self-management. The entity class has the *QoS current values* (it indicates the current status of the node's treatments and link's interactions. This kind of value is to monitor, during the exploitation, by the provisioning in order to have a real time image of the service behavior.), which provide information for the constraints class (Figure 4-3) which includes the *QoS conception values* (it's decided at the phase of service conception. It introduces the maximum possibilities of the node's treatments and the link's interactions.). These classes verify that the service can be offered. This QoS information helps to support the management treatment and the decision-making process of the component.

In order to integrate the personalization, we have also proposed *User profile*. It includes several sub profiles: *"General Information Profile"* which indicates all the information about his resources; *"Location Profile"* which describes each resource according to different location and *"Agenda Profile"* which describes each resource according to diverse activity. The user preferences are applied when sorting and filtering the information for each kind of profile.

## 4. The VPIN

The VPIN (Virtual Private Information Network) has been proposed [1] to have the whole information concerning the user information system in real time, taking into account his location and activity. The fact of getting the complete information allows having knowledge of the component that conforms the user system information. This collect of information is known as a knowledge base, which allows taking decisions according to the user needs when accessing a service. In order to have the whole user system information, the VPIN is organized in four visibility levels that will be detailed (§4.1). These visibility levels permit to get the information of a resource and its behavior (QoS) at an instant "t". At every level, a network of those resources is formed, which allows having knowledge of the behavior of the resources (QoS). The goal of forming a network at each visibility level is to have the knowledge of the resources available to be used when the user wants to access to a service and establish a session. Therefore, if there is a degradation of the QoS in a component, it could be replaced by another one that fulfills the expected QoS. The change of a component can be fulfilled with the help of the Real Time Profile (§3.2), which is present at every resource and at every visibility level. It's for this reason that the Real Time Profile has been chosen for the proposition. In order to see the information contained in the Real Time Profile, there will be an example of the relevant information (§4.2) that is used in a resource.

### 4.1 Organization of the VPIN

The VPIN allows organizing the information in a structured way. This organization enables get the pertinent information that is needed in right time and the right place when needed. This organization follows the visibility levels in order to have the whole vision of the user information system. This is applied at every visibility level (User, Terminal, Network, and Service) based on the Meta-Model NLN explained before (§3.1). Applying the Meta-Model NLN to the network level, we can say that the node is a router and the link is the way they are connected, in this case it will be through the routing protocols, and the set of these nodes and links form the network at this visibility level.

The organization of this VPIN (**Figure 5**) offers a global vision of all the components that constitute the user centric information system. The benefit of this organization is that, the modeling structure is the same image of the real world, which allows having pertinent information in a real time. This organization is user centric, it means that it has been conceived to fulfill the user needs, breaking down technical barriers that avoid a user to access his services.

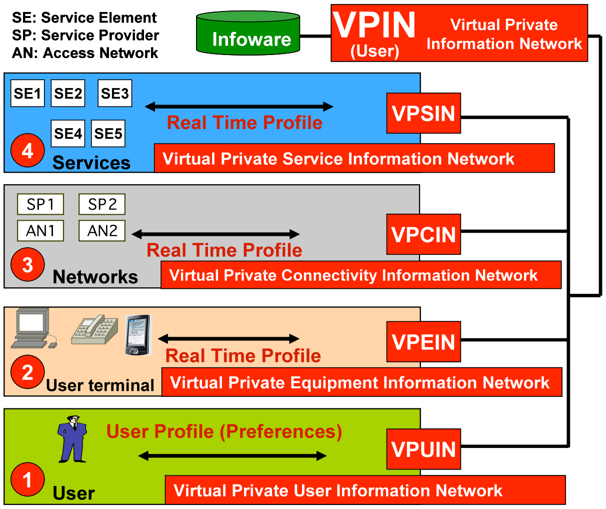

SE: Service Element
SP: Service Provider
AN: Access Network
Infoware
VPIN
(User)
Virtual Private Information Network
SE1
SE2
SE3
SE4
SE5
Real Time Profile
VPSIN
4
Services
Virtual Private Service Information Network
SP1
SP2
AN1
AN2
Real Time Profile
VPCIN
3
Networks
Virtual Private Connectivity Information Network
Real Time Profile
VPEIN
2
User terminal
Virtual Private Equipment Information Network
User Profile (Preferences)
VPUIN
1
User
Virtual Private User Information Network

**Figure 5: VPIN organization**

The organization of this VPIN is done in four levels. At the first level, there is the VPUIN "Virtual Private User Information Network" (Figure 5-1). In this level, there is all the information about the user profile where are included the user preferences. These preferences can change according the user location and activity.

In the second level, there is the VPEIN "Virtual Private Equipment Information Network" (***Figure*** **5**-2). It is composed of all the equipment information belonging to a user. The equipment information referred is not only that which belongs to the user, but the equipments that are available and can be used by the user according to his location.

As third level, there is the VPCIN "Virtual Private Connectivity Information Network" (***Figure*** **5**-3). It concerns all the access networks information from which user can be connected. The particularity in this level consists that it can be found the access networks from operators as well as service providers.

In the fourth level, there is the VPSIN "Virtual Private Service Information Network" (***Figure*** **5**-4). It provides the service composition information related to a user. An example of this level could be, when a user is located at the airport and desires to buy a ticket with his PDA. There is a server that broadcast this information, but if this server stops working, in the VPSIN, there is going to be another server that takes over the task. Hence the user should have always the necessary information about this service.

## 4.2 Relevant information

As it has been showed (§4.1) the organization of the VPIN allows having the necessary information at every visibility level. But, instead of keeping the allocation-related information in the VPIN, we have decided to have all the information necessary, sufficient and persistent in our knowledge base.

It's within the *Real Time Profile* (described in section 3.2) that there is the QoS information, which is present in every component and in the visibility levels as VPEIN, VPCIN and the VPSIN.

The *Real Time Profile* is capable of providing all the information of a component that conform the user information system. When this information is collected, it allows making the adequate decisions when a user-session is established. The users preferences are considered to take the decisions. They are not only considered at the time of the establishment of the session, but also during the entire session. This makes possible the dynamic session management in real time. The session components are terminals, access networks, core network and services that the user desires to employ according to the QoS component and preferences. In these components, we have to take into account what information needs to be included in each visibility level for the dynamism of the user-session. According to the Real Time Profile structure (Figure 4), we display the relevant information for a PDA (Figure 6) and the service component "E-mail" (Figure 7).

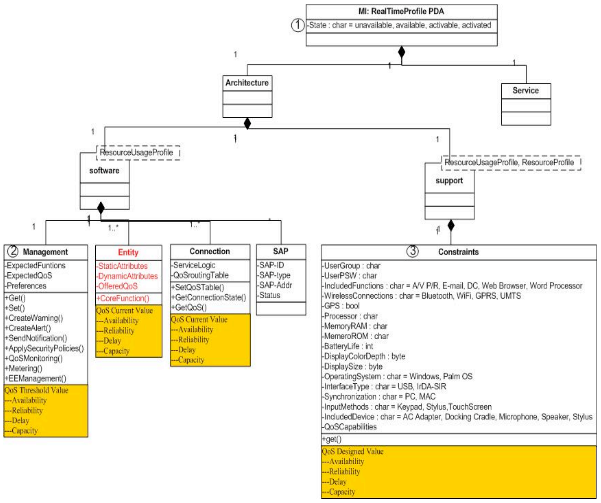


**Figure 6: Relevant Information in PDA**

These examples can demonstrate that even if there are different components, the information structure is the same. In these components also includes the QoS information, which allows them to know the behavior of the component.

In this Real Time Profile we focus in the third part (Figure 6-3) from which we have the constraint class of the PDA. As we can see, the representation of the information of the PDA follows the same structure given by the Meta-Model NLN (§3.1) and the information that is contained in the constraint class follows the same structure about the visibility levels. It means, there is information that concerns the user, terminal, network and services in the respective order.

The "E-mail" service component (Figure 7) is represented as information model in order to identify the relevant information. In this component, there are displayed the constraints of this service component concerning the user (login, password), equipment (10 MB to be used), the network (protocols POP3, SMTP, IMAP) and the service (text to vocal converter).

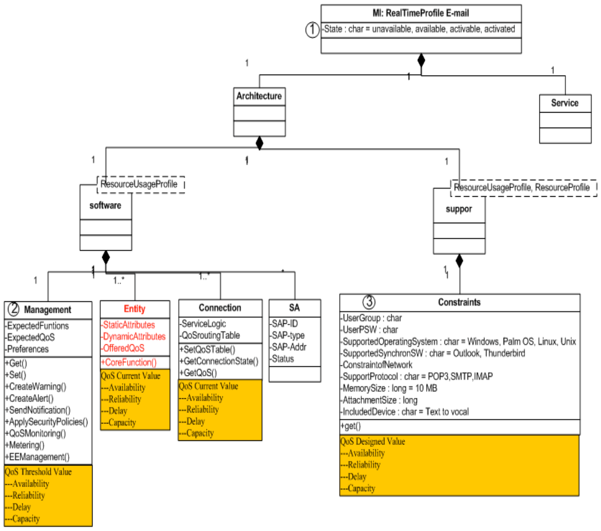


**Figure 7: Relevant Information in service component "E-mail"**

On of the advantages of having the same structure in a component is that, it allows a homogeneous correlation of the information that is stored in the component. In that way, the information of that component can be demanded at any moment with the security that the given information is completely trusted.

## 5. VPIN: Functional dimension proposition

After a briefly explanation about the organization of the VPIN (§4), the next aspect to consider is the functional issues in the knowledge base. That is, how to identify the factors that makes changing the information that is stored in the knowledge base. This knowledge base is the representation of changes that comes from a real environment. Another issue to treat is, what information will be modified due to this change in the knowledge base. It is through the information change that an event is triggered.

In this section, we propose the functional dimension, in which we will verify how to identify the inferential events (§5.1), and how this knowledge base going to react when a modification of the information (inference) occurs (§5.2).

## 5.1 Inferential Events

The VPIN contains relevant information, including QoS information about the component. The collection of this information makes the knowledge base to be decisional. When a change is done in the knowledge base, there is the need to check the architecture to verify the impact that is affected in the knowledge base, therefore, according to the structure, the inference will be done. Nevertheless, this information needs to be updated when the QoS information of a component changes. With the purpose to always have the component adapted to the user needs, we need to know how to react, internally, when there is a change of QoS in a component. That is why the knowledge base can be decisional, due to the management QoS of each component.

In order to detect the causes that make produce a inference in the knowledge base, we identified two kinds of events, the event that is triggered when the QoS in a component changes (*QoS change event*), and the other type of event is produced by the change of state of the component (*state change event*). These two kinds of events are integrated in an agent that is integrated in the Real Time Profile; we called this agent “EMA”, which means Event Monitoring Agent.

### QoS change event

We have identified an event that makes having changes in the knowledge base. This event is triggered due a change of QoS in a component. We have called this event as *“QoS change event”*.

An example of this king of event could be, when a terminal (PDA) that is used in a session arrives to the maximum capacity in the RAM memory; therefore, the terminal cannot be used anymore. Thus, due to the change of QoS in the component, the QoS change event launches the trigger (Figure 8).

Therefore, an action must be taken; in this case, it will be a change of terminal, for example, changing to the PC terminal.

In that way, the terminal that has failed won’t participate anymore to the user-session, due to the internal problem.

It is with the help of the EMA’s agent (Event Monitoring Agent) that the behavior of a component can be monitored

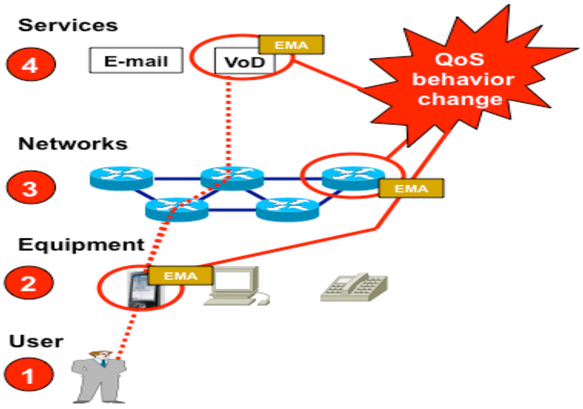


**Figure 8: QoS change event detected by EMA**

The main focus of the *QoS change event* is the change of QoS, but that is not the only cause in a component that an event has to be triggered.

The other kind of event works through the change of state in a component. We called the second kind of event as *“state change event”*.

### State change event

We identify that the “*state change event”* can be caused by a change of state in the component. That is, unavailable, available, activable and activated.

It means that if a component changes the state from unavailable into available, that is a change where the QoS of the component doesn’t intervene at all. It’s only the state of the component that changes.

The EMA’s agent can be used for the *QoS change event* as well as the *state change event*. The difference is that when the QoS of a component changes, it is through the “*QoS change event*” that an action will be taken. In the other hand, if the EMA’s agent identifies that the component changes of state, in that case, it’s through the *“state change event”* that an action will be taken.

As already explained (§1), the different types of mobility; that is mobility of the user, terminal, network and service. The *state change event* can be produced, due to the different types of mobility. That is, when a terminal moves to another place, maybe it will not be connected to the same access point, nevertheless, it doesn’t mean that the terminal had a change of QoS, the terminal has the same capacities, but as it has been moved, a change of the state has been produced.

As showed in (Figure 9), we present an example where the user mobility is invocated. As at every component the EMA's agent is present, when the user moves, the event is triggered when the user tries to login in the other terminal. Therefore, there is a notification that the PDA is no more used for access the service. Therefore, the terminal to use will be the PC. This change of terminal makes an inference at the network and service level. The change is that as the terminal equipment is different (from PDA to PC), a new routing path must be built in order to continue to access the service.

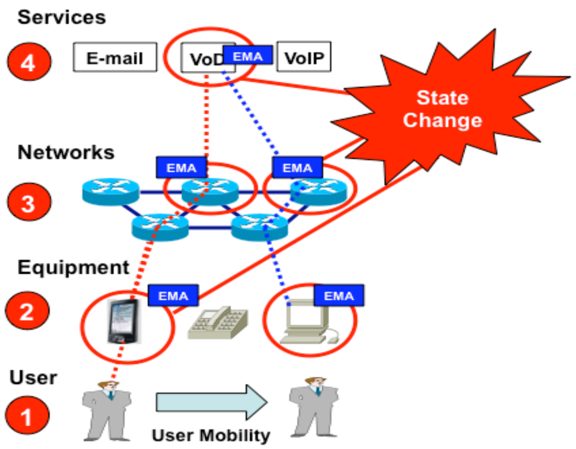


**Figure 9: State change event detected by EMA**

The inference at the service level will be, the change of the address from the router that receives the traffic. Therefore, another router will send the traffic to the Video over Demand (VoD) service.

Once identified the factors that makes the inference in the knowledge base, we show how this is achieved through the informational model.

As showed in (Figure 10), in the Real Time Profile the agent is placed at two points, when a change of QoS in the component is done, and when the state of the component changes.

The *QoS change event* is triggered through the management class (Figure 10-1) is produced. In the Management class, it exists the "QoS Monitoring" operation; the function of this operation is to monitor the component in a real time. There is also the "Send Notification" operation; the goal of this operation is to send a notification to which it concerns when the event is triggered.

Considering the *state change event*, as mentioned before, this event is triggered when there is a change of the state in a component. The agent is located at the beginning of the component, where the sate of the component is treated (Figure 10-2).

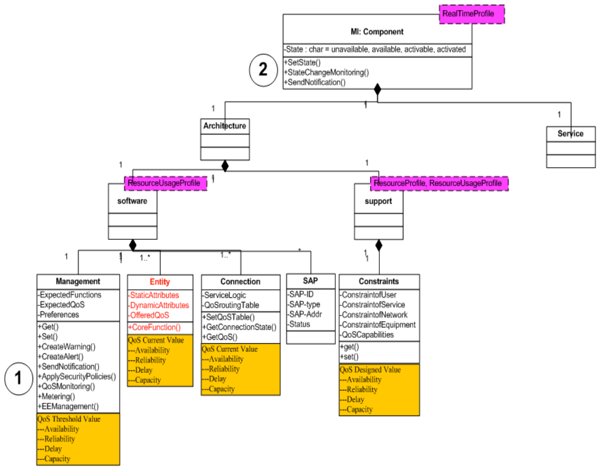


**Figure 10: QoS change and state change events managed by the agent**

The change of state can be caused due a mobility type. Thus, the operations that are at the beginning of the component are three, the first operation is "Set State", the second operation is "State Change Monitoring", and the third operation is "Send Notification". These three operations allow monitoring the component in real time. The "Set State" operation is used to set the state of the component. The "State Change Monitoring" verifies and updates the sate of the component, and the "Send Notification" operation send the notification to which it concerns.

The agent that is in charge of the events (by component and by state) is directly integrated in the Real Time Profile (Figure 10) in the management class and the state class respectively.

## 5.2 Inferential management

According to the events identified before (§5.1), it can be detected the different causes of information change in the knowledge base.

Considering these information changes (inference) in the knowledge base, we have to be able how to treat them, and what changes in the knowledge base are done. It means, when the behavior or the state of a component have been changed, the information of that component is registered in the knowledge base. Thus, according to this change, there is an update of the information that has a relation with that component.

First, we focus in the events that are produced due a change of the behavior in a component. As showed in Figure 11, there is the representation of the Real Time Profile as a component (right side) and the Real Time Profile when it represents a network (left side). The network Real Time Profile represents the components at the same visibility level.

As an example, we can say that the component is a terminal (PDA), and that the whole terminals available, form the network of terminals (PDA, PC, Telephone, etc.). Thus, starting with the component behavior, there is the management class (Figure 11-1), which monitor the component internally. When a behavior change is done in the component, and if the management class can't treat this change, there is a notification that is sent through the service class (Figure 11-2) to the Real Time Profile Network. Note that the service class is the interface by which the notifications are sent or received from a component. When the notification is received through the service class (Figure 11-3) of the Real Time Profile Network, it is forwarded to the management class (Figure 11-4), which allows treating the event notification that has been sent from a component. After, the management class notifies to the Entity class (Figure 11-5), and it makes an update about the QoS information that is contained in this Real Time Profile.

Once showed how the inference is done when a QoS behavior in a component changes, now, we show how the inference occurs when there is a state change in a component.

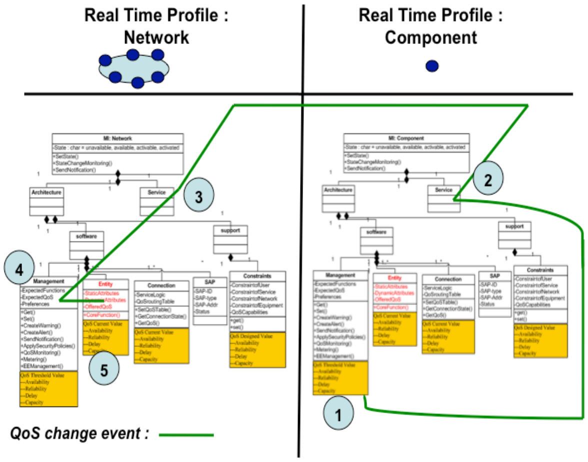


**Figure 11: QoS change event inference**

Following the same architecture of the Real Time Profile (Figure 11), which can represent a component, or a network, we show in Figure 12 how the inference is treated when the state of a component changes.

When the state in a component changes, the QoS management could not be notified about this change. For example if a component as a PC is being used, and if there is a cut of electricity, the components will be unavailable, even if the QoS of the terminal was working properly. Therefore, it is demonstrated how the inference in that case will be treated.

When a component changes of state, that could be due to the mobility of the terminal, network, service, and for the user mobility, the change of state is produced when the user tries to access to a service through another terminal.

Note that the operations that indicate, set and update the state of a component are at the beginning of the component.

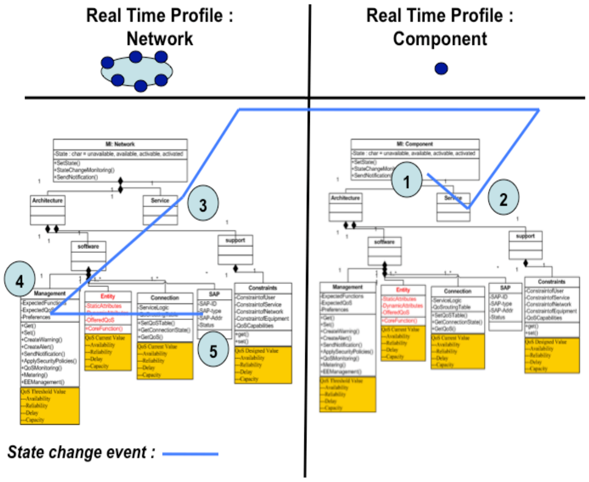


**Figure 12: State change event inference**

First, when a change of state in a component is done (i.e. available to unavailable), there is a notification sent (Figure 12-1) from the component class to the network Real Time Profile. This notification is sent through the service class (Figure 12-2) of the Real Time Profile component to the service class (Figure 12-3) of the Real Time Profile Network that contains the set of elements that conform the network of the components. Once the service class of the Real Time Profile Network receives this notification, it will be forwarded to the management class (Figure 12-4). With the help of the QoS management that is integrated, the notification can be treated. Consequently, the management class will notify the SAP class (Figure 12-5). The SAP class has the whole list of address of the components that conform the network of components. Therefore, when the notification is received to this class, the address of the component that became unavailable will be erased from the list. This inference can produce an internal change of the Real Time Profile Network, due to the unavailability of the component, a recalculation of the QoS need to be done to have the real information of the components that are available.

It is with the help of the event manager agent (EMA), which is present at every component, that a control of the changes done in the knowledge base can be achieved. Thus, in this way, the QoS information that is stored is always the real image of the components that are available.

## 6. Experimentation

We describe a scenario with the purpose of presenting the feasibility of implementing the concepts proposed about the events management.

We depart from a scenario where is presented a real world vision (§6.1), and how this scenario function in a platform which the software to do it is Oracle Database (§6.2).

### 6.1 Real vision scenario

The real-world vision of the scenario (Figure 13) is presented to separate the components that take part of a session throughout different levels.

The goal of the scenario is to ensure an end-to-end user-service session in a seamless and continuous way despite the user's mobility, with the help of the relevant information. In Figure 13-1, the user has to his disposition a Personal Computer (state Activable), a telephone (state Activable) and a PDA (state Activable). At instant “t”, the user checks his e-mail on his PC; the choice of using the PC follows the user preference, which the intention to use the Personal Computer (state becomes Activated) is due to the resolution in the screen. Therefore, in that moment, the user’s VPIN is set according to the user location, which in this case is the office. The current session is supposed to be supported by the service composition including SE1 (authorization), SE2 (e-mail), SE3 (SMS) and SE6 (E-mail client). At the end of the day, the user has to go home and the transport used is his car.

As the user mobility is produced, there’s a terminal exchange based on the user’s preferences. According to user's preferences, he desires to continue checking mails in his PDA. At this instant, the states of the equipments are Personal Computer unavailable, telephone unavailable and PDA activated. The user consults his PDA (SE4), which implies an adaptation to continue checking his e-mail. Once this change has been done, it could be dangerous to the driver while he reading his e-mails (Figure 13-2), therefore, according to his preferences previously configured, the service delivery changes. From now, the e-mail text is translated into vocal SMS. This implies a new component in the terminal (SE5), allowing user to listen to his e-mails while driving. In order to make the adaptation of the service component “text to vocal”, verification should be done to check whether this new component could be supported by the PDA. For the user, all the adaptations according to his service were transparently done with the help of the information that is contained in the VPIN. In this case, the VPIN acts as an orchestrator in choosing adequate service components according to user preferences to ensure the end-to-end service of the user.

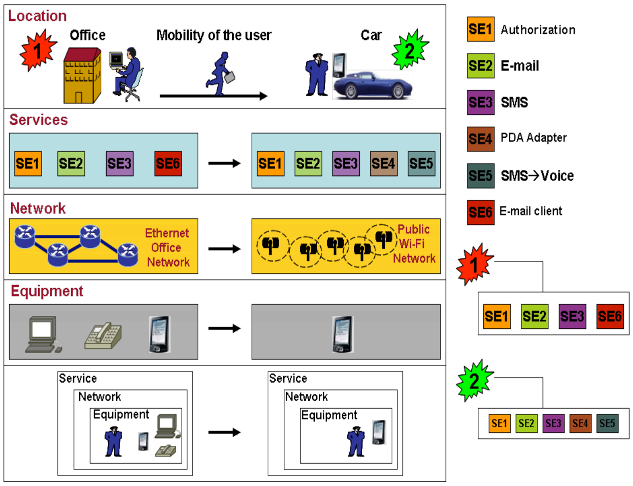


**Figure 13: Scenario**

This scenario demonstrates a real case of how the services that are employed by a user. To achieve this scenario, the information that is contained in the VPIN has to be the same that reflect the components that are used by the user, thus, the user will access services in a continuous way without interruptions.

### 6.2 Functional process

In order to demonstrate the proposed concepts about the inferential aspects, we have implemented a structured relational database on the Oracle 11g platform. This implementation demonstrates the first works to implement the *knowledge base*.

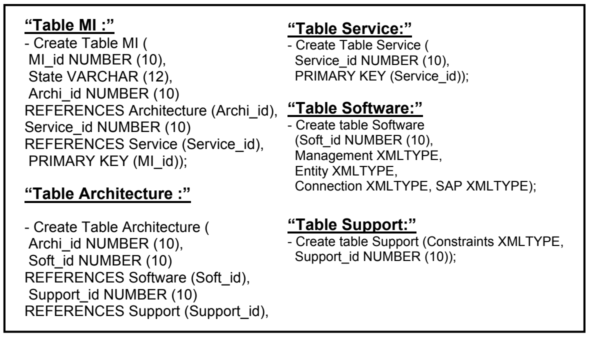
Based in the description of the scenario (§6.1), the test to demonstrate is how the inference is taken into account in the knowledge base due to the user mobility. The user mobility implies a change of the terminal for access the service.

The structure proposed is based on the Real Time Profile (§3.2). This structure allows having the knowledge of the whole user-session when the user accesses a service, no matter the user location.

The realization of this implementation has been developed in three steps.

The first one is to create a database structure that reflects the Real Time Profile. As second step, we create the triggers that launch the inference when information is added or updated. At the moment of the creation of the trigger, the process is launched, thus the trigger is ready to be executed when the condition is accomplished. The third step is to feed this database with real information, and as the triggers are already launched, when the information is added or updated, the trigger will be executed.

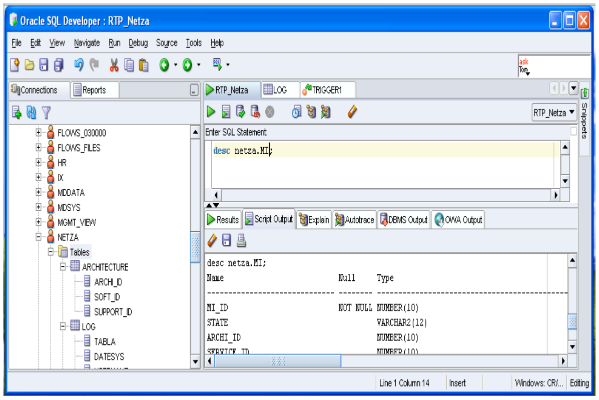
As the firs step, a creation of the structure in a database has been done (Figure 14). This structure allows having the structure where the information of the component will be stored.



```
"Table MI :"
- Create Table MI (
 MI_id NUMBER (10),
 State VARCHAR (12),
 Archi_id NUMBER (10)
REFERENCES Architecture (Archi_id),
Service_id NUMBER (10)
REFERENCES Service (Service_id),
 PRIMARY KEY (MI_id));

"Table Architecture :"

- Create Table Architecture (
 Archi_id NUMBER (10),
 Soft_id NUMBER (10)
REFERENCES Software (Soft_id),
 Support_id NUMBER (10)
REFERENCES Support (Support_id),

"Table Service:"
- Create Table Service (
  Service_id NUMBER (10),
  PRIMARY KEY (Service_id));

"Table Software:"
- Create table Software
  (Soft_id NUMBER (10),
  Management XMLTYPE,
  Entity XMLTYPE,
  Connection XMLTYPE, SAP XMLTYPE);

"Table Support:"
- Create table Support (Constraints XMLTYPE,
  Support_id NUMBER (10));
```

**Figure 14: Oracle's database structure**

We propose a series of events allowing the Database to react achieving the inference as described before (§5). These events help to adapt the information according to an event occurred in the user-session. This will avoid the service interruption when user is accessing the service. The tools used for this implementation is Oracle Data Base and SQL Developper for Oracle (**Figure 15**). Theses graphical software tools allow simplifying database development tasks.



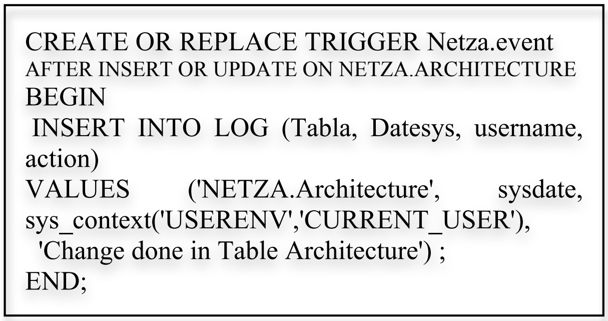
```
desc netza.MI;
```

```
desc netza.MI;
Name                           Null     Type
------------------------------ -------- ----------------------------------------
MI_ID                          NOT NULL NUMBER(10)
STATE                                   VARCHAR2(12)
ARCHI_ID                                NUMBER(10)
SERVICE_ID                              NUMBER(10)
```

**Figure 15: Oracle SQL Developer**

In oracle databases, there is the possibility to create process that allows launching actions in a database. The Triggers are programmed internally in the database, thus a space is entirely reserved to store the Triggers. The triggers make possible to modify necessary information when a change in the database has been done. In our scenario, the change of the state is going to produce an action. This permit the inference in our knowledge base through the events previously defined (§5). For creating a trigger, it is necessary a CLI (Command Line Interface) in order to indicate by a command what table to monitor as well as the action to be taken when a change is done in the table.

In the **Figure 16**, there is illustrated an example of the creation of a Trigger in Oracle.



```
CREATE OR REPLACE TRIGGER Netza.event
AFTER INSERT OR UPDATE ON NETZA.ARCHITECTURE
BEGIN
 INSERT INTO LOG (Tabla, Datesys, username, action)
VALUES ('NETZA.Architecture', sysdate, sys_context('USERENV','CURRENT_USER'),
 'Change done in Table Architecture') ;
END;
```

**Figure 16: Creation of Trigger in Oracle**

The trigger showed in the Figure 16 describes the creation o a trigger. The name of the trigger is "netza.event", and the factor that is going to launch an action is when there is an insertion or update of information in the table "Netza.Architecture". When this condition is accomplished, the action to be taken is to insert a data in a table called "LOG". The content of the Log's table can be defined previously or generated dynamically with the help of the declaration of variables. In this case, in the log table, we make insert the name of the table (netza.architecture), the data of the modification, and the user that has done the modification; as well a message saying that there was a change in the table.

The third step to do is, the insertion of information in the data base structure. With the insertion or update of information, this modification will imply the react.

We have configured some triggers to react in the database, hence if the resource information changes because of behavior, a reaction can be taken place according to the event. The triggers configured, can represent the user mobility, i.e., when the user moves to another place, it's at the moment of the login in another terminal that the event is triggered and the inference occurs.

## 7. Conclusion and perspectives

In this paper, we analyze initially the user new requirements for accessing services in the Next Generation Network's context. These new requirements are confronted with the increasing development of a heterogeneous environment in equipments, networks and services, as well as the mobility (equipment, network and service). This new paradigm leads, as result, the impossibility of dynamic service adaptation according to user preferences. The latter inspired us to regard the user as the central of the whole system and reconsider the Information System in a User Centric way.

We regard the different solutions proposed by different standardization organisms and point out the lacks concerning the use requirements. We perceived that the pertinent information related to the context NGN (heterogeneity, mobility, user-centric) are not taken into account. Therefore, based on models from Telecom ParisTech, the QoS agent and the state agent, we have introduced the inference events.

Hence, we have proposed our knowledge base (VPIN), which is an information model based capable of dynamically managing the information. Architecturally, it is different from the existing proposals thanks to its application-independent characteristics due to basing on a modeling of the real telecom world.

Functionally, it provides the decisional information as QoS information for a certain user session, and takes into account the user's preferences as an important filter of the information. With the help of this knowledge base, we can make the necessary decisions by the reaction to the events in order to always conform to the user desired service with his QoS. We studied, in the first step, the events activated by the change of QoS and those activated by the change of state. The different agents proposed allow the management of events that can occur during the user-session. Two detailed examples are illustrated to detail how the inferential management works.

The implementation has permitted us to examine the feasibility of the event driven aspect, i.e. how to manage the studied event in the Oracle environment, with a created VPIN in the database. What remains as the future work is the fully development of the event agents which can include all the real time events, and test them in a real time platform in order to evaluate the performance of our proposals.

## Acknowledgments

The authors would like to thank Gladys Diaz, Ken Chen and Xiaofei Xiong for their contribution in this work.

## 8. References

[1] Netzahualcoyotl Ornelas, Noëmie Simoni, Ken Chen, Antoine Boutignon, VPIN: User-Session Knowledge Base for Self-Management of Ambient Networks, UBICOMM 2008

[2] Jürgen Dunkel, Alberto Fernández, Rubén Ortiz and Sascha Ossowski, Event-Driven Architecture for Decision Support in Traffic Management Systems, ITSC 2008

[3] Zakir Laliwala, Sanjay Chaudhary, Event-driven Service-Oriented Architecture, ICSSSM 2008

[4] A. Kumar Harikumar, R. Lee, C. Chiang, H. Yang, An Event Driven Architecture for Application Integratino using Web Services, IEEE IRI 2005

[5] ITU-T, "Recommendation M.3100 TMN-Generic Network Information Model," 1996.

[6] ITU-T, "Telecommunication Standardization Sector http://www.itu.int/ITU-T/

[7] Common Information Model (CIM): http://www.dmtf.org/standards/cim

[8] Distributed Management Task Force (DMTF): http://www.dmtf.org/

[9] Management Object Format (MOF): http://www.dmtf.org/education/mof/

[10] Web-Based Enterprise Management (WBEM): http://www.dmtf.org/standards/wbem/

[11] W3C, "Extensible Markup Language (XML) 1.0"; http://www.w3.org/TR/2006/REC-xml-20060816.

[12] Directory Enable Network (DEN): http://www.dmtf.org/standards/wbem/den

[13]Shared Information Data/Modell (SID): http://www.tmforum.org/browse.aspx?catID=2008

[14] New Generation Operations Systems and Software, NGOSS:http://www.tmforum.org/BestPracticesStandards/SolutionFrameworks/1468/Home.html

[15] Unified Modeling Language (UML) http://www.uml.org/

[16] Simple Network Management Protocol (SNMP): http://www.ietf.org/rfc/rfc1157.txt

[17] Atherton M. "Deploying CMDB Technology Pragmatism and realism will deliver the benefits, Freeform Dynamics Ltd.

[18] Information Technology Infrastructure Library (ITIL) www.itil-officialsite.com

[19] Data Mining: K. Cios, W. Pedrycz, R. Swiniarski, L. Kurgan, Data Mining: A Knowledge Discovery Approach, Springer, ISBN: 978-0-387-33333-5, 2007.

[20] Z. Benahmed Daho, N. Simoni, Towards Dynamic Virtual Private Service Networks: Design and Self-Management, in IEEE/IFIP NOMS 2006

[21] H. Huynh, E. Lavinal, N. Simoni, A Dynamic QoS Management for Next Generation of Services, ICAS 2007

[22] Noëmie Simoni, Chunyang Yin, Ghislain Du Chéné, Service continuity management through an E2E dynamic session in NGN, NOMS 2008

[23] Noëmie Simoni, Chunyang Yin, Ghislain Du Chéné, "An Intelligent user centric middleware in NGN: Infosphere and AmbientGrid" COMSWARE 08

[24] Noëmie Simoni, Simon Znaty, Gestion de Réseau et de service: Similitude des concepts, Spécificité des solutions. ISBN:2225829802

## Vitae

Netzahualcoyotl Ornelas was born in Guadalajara (Mexico) on 1980. He graduated as Engineer in Computer Science from University Guadalajara Lamar in 2004 (Mexico). He received the MSc. in Networks from the Pierre et Marie Curie University (Paris 6) in 2006 (France). At present, he is preparing his PhD. in the Department of Networks at the University Paris 13 since 2007 (France). He is founder member of the IPv6 Task Force Mexico. He participated in research projects with University of Guadalajara, RENATER, SFR and Telecom ParisTech about IPv6, Multicast, ENUM, QoS, and NGN architectures. He collaborates in the UBIS project with University Paris 13, Telecom ParisTech and SFR. His current research area of interest is the self-management of networks and services in a NGN context.

Professor Noëmie Simoni is a chairman of Architecture and Engineering of Networks and Services (AIRS) research group at the Department of Computer Science and Network, Telecom ParisTech. Her research interests include QoS management and the end to end modeling of the entire telecommunication system. Through academic projects and industrial contracts, her expertise covers wide range of management topics. Her main works is focused on information modeling, management process re-engineering, workflow integration and convergence of control and management planes. She published over 70 technical papers and is co-author of two books, one on Network and Service management and another one on NGS (New Generation of Services). She is chair and program chair of the French-speaking conference on Networks and Services Management (GRES) which is held every two years and sponsored by Telecommunication Operators.

Doctor Chunyang YIN was born in China, in 1980. She is graduated from Tongji University, China, in Computer Science in 2002. She received the MSc. Degree in Computer Science from ENST, Paris, France in 2005 and the MSc. Degree in Network Security from Tongji University, China, in 2006. She received her PhD. degree in Computer Science and Network from TELECOM ParisTech in 2008. She's currently working as Post Doc. in the Department of Computer Science and Networks, TELECOM ParisTech. As a member of AIRS research group since September, 2004, she has participated projects cooperated with SFR as IA/PO, VTC2E and Seamless Userware. Her main research interests lie in the area of the service continuity in an NGN context.

Antoine Boutignon was born in Lille (France) on 1952. He graduated from ESME Sudria in 1977 (Engineer in Mechanics and Electronics). From 1977 to 1982, he worked at Steria to develop a real time system for EADS. In 1982, he joined the CS group to participate in the development of Datagram / X25 software and systems for telecommunication equipment. From 1990 to 1997, he worked as a study engineer in the Thales Communication R&D and Architecture division in the field of ATM Network equipment and services. He joined SFR in 1997 to participate in architecture studies on Data Networks. He is attached to the SFR R&D and Standards department as manager in the area of IP transport technologies and Mobile Core Network technologies. He has been involved in several research programs with Institut Telecom, Paris 13 and INRIA about IPv6, QoS, IMS, Service and Architecture. He is involved in several collaborative projects European as well as French. He published about 15 papers.

# Performance Analysis of a Priority Queue with Session-based Arrivals and its Application to E-commerce Web Servers

Joris Walraevens, Sabine Wittevrongel and Herwig Bruneel
Department of Telecommunications and Information Processing (IR07)
Ghent University - UGent
Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium.
E-mail: {jw,sw,hb}@telin.UGent.be

## Abstract

*In this paper, we analyze a discrete-time priority queue with a session-based arrival process. We consider an infinitely large user population, where each user can start and end sessions. Sessions belong to one of two classes and generate a variable number of fixed-length packets which arrive to the queue at the rate of one packet per slot. The lengths of the sessions are generally distributed. Packets of the first class have transmission priority over packets of the other class. The model is motivated by E-commerce web servers and web servers handling delay-sensitive and delay-insensitive content. By using probability generating functions, performance measures of the queue such as the moments of the packet delays of both classes are calculated. The impact of the priority scheduling discipline and of the session nature of the arrival process is demonstrated. We furthermore use our analysis to provide specific results for an E-commerce web server.*

***Keywords-priority; session arrivals; E-commerce; web server; queueing analysis***

## 1 Introduction

We analyze a two-class discrete-time Head-Of-the-Line (HOL) priority queue with a session-based arrival process.

HOL priority scheduling is one of the main scheduling types in network buffers to diversify the delays of traffic streams with different delay requirements [24].When delay-sensitive high-priority packets (packets of voice and video streams, gaming ...) are present in the buffer, they are transmitted. Best-effort low-priority packets can thus only be transmitted when no high-priority traffic is present. Another reason why one would like to diversify the delay characteristics of different applications is the following: one application might provide revenues for the provider while another does not (or to a lesser extent). It is then natural (and profitable) to give priority to the packets of the first application.

Besides priority scheduling, there are numerous other scheduling types proposed in the literature to diversify the Quality-of-Service (QoS) of different applications. A (theoretical) scheduling discipline is Generalized Processor Sharing (GPS). With this discipline, the 'transmission unit' spends weighted fractions of its capacity on the different classes. Delay-sensitive traffic gets a larger weight than delay-insensitive traffic, so that it gets some kind of preferential treatment. One possible implementation of GPS is Weighted Fair Queueing. For a further overview of such scheduling disciplines, we refer to [12]. However, priority scheduling is still one of the most popular scheduling types, since it is relatively easy to implement and to operate.

In the current paper, we further consider an arrival process induced by a two-layered structure. *Sessions* are started and terminated by users on the higher layer. These sessions inject trains of *packets* in the network. Since we perform a discrete-time analysis, we assume time is divided into slots of equal length and we assume that packets of a session arrive to the queue at the rate of one packet per slot. Note that this two-layered structure introduces *time correlation* in the packet arrival process. Indeed, since the packets in a session arrive in consecutive slots, the number of packet arrivals in one slot depends on the number of arrivals in previous slots. Session-based arrival processes are an adequate choice to model, e.g., the common segmentation of data files into packets before their transmission through a telecommunication network [14, 17].

In particular, the suggested arrival process is an ideal candidate to model the output buffer of a web server [15]. A web server is a computer system that accepts requests from users for a certain web page or embedded file and that responds by sending the requested file to the user. Traffic generated by a web server towards its output buffer can be described by a session-based arrival process. In the case of an E-commerce web server, it makes sense to prioritize the downloads on a (potential) revenue base [27], that is, to give priority to the transmission of packets of content that is likely to provide (large) revenues. Furthermore, most web pages contain content that is delay-sensitive, for instance multimedia content. Priority can then also be given to the transmission of files containing this content over other downloads [37].

From a queueing-analysis point of view, the combination of a priority scheduling discipline and a session-based arrival process with generally distributed session lengths forms the main novelty of this paper. We thereby extend previous analyses [5, 32] where the session lengths were assumed to have a specific distribution (deterministic and geometric respectively). The distributions of the session lengths may further be class-dependent, which reflects that different priority classes represent different applications. We analyze the buffer contents (i.e., the number of packets in the buffer) as well as the packet delays (i.e., the number of slots a packet stays in the buffer) of both the high-priority and low-priority class using *probability generating functions* (pgfs). In contrast with the specific session-length distributions studied in the past (see [5, 32]), an infinite-dimensional state vector has to be defined when dealing with generally distributed session lengths. This combined with the priority scheduling makes the analysis of the low-priority buffer content and packet delay far from straightforward. Nevertheless, closed-form formulas for the means of these stochastic variables (and in most cases also for higher moments) can be found by means of the analysis technique developed in this paper. From a networking point of view, the added value lies in the application of our results to an E-commerce web server. We finally note that this paper is the extended version of [1].

The remainder of the paper is structured as follows. In the next two sections, we describe some related literature and present the mathematical model respectively. In section 4, we construct a functional equation. This functional equation is the starting point of the analysis of the steady-state number of arrivals per slot, the steady-state buffer content and steady-state packet delay, described in sections 5, 6 and 7, respectively. Numerical examples are treated in section 8, while we apply our results to an E-commerce web server in section 9. We finally conclude this paper in section 10.

## 2 Related literature

A first property of our model is the HOL priority scheduling. There have been a large number of contributions in the related literature with respect to the performance analysis of HOL priority queues. In particular, discrete-time HOL priority queues with deterministic service times equal to one slot have been studied in [4, 9, 11, 13, 19, 20, 22, 25, 26, 28, 30, 31, 35]. Hashida and Takahashi [13] analyze a two-class priority system, where the high-priority arrivals and low-priority arrivals are governed by a two-state Markov-modulated Batch Bernoulli Process and a Batch Bernoulli Process respectively. The numbers of per-slot arriving high-priority packets are governed by an underlying Markov chain and the numbers of per-slot low-priority arrivals are independent and identically distributed (i.i.d.). Application of a conservation law leads to expressions for the mean delays of both classes. Takine et al. [26] analyze the same model as in [13] by means of matrix-analytic techniques. Moments of high-priority, low-priority and total system contents and moments of high-priority and low-priority delay are calculated. In [25], bounds for the delay distribution are given in a multi-server queue with a rather general arrival process. Xabier Albizuri et al. [35] study the delay of the low-priority traffic in a multi-server queue by assuming that the number of servers available for the low-priority traffic is variable (depending on the number of high-priority packets served at the time). Mehmet Ali and Song [22] analyze a queue with the arrival process existing of a number of two-state Markovian sources and by using probability generating functions. In [19], priority queueing systems with a general number of priority classes are analyzed. The distribution of the number of per-slot arrivals depends on the state of a two-state Markov chain. In [4, 20], two-class multiserver queues are analyzed with the number of arrivals i.i.d. from slot to slot. The joint pgf of the system contents of both classes is calculated in both papers (although the analysis in [4] is more tedious than in [20]). The pgfs of the delays of both types of packets are also calculated in [20]. From these pgfs, moments of the analyzed stochastic variables are calculated in both papers. In [4], the corresponding probabilities are furthermore numerically determined using Fast Fourier Transforms, while these probabilities are analytically approximated for high values of the stochastic variable (tail probabilities) in [20]. Walraevens et al. [30, 31] study the steady-state buffer content and packet delay in the special case of an output-queueing switch with Bernoulli arrivals and the transient buffer content respectively. Finally, in [9, 11, 28], different queueing models with finite buffer size are studied.

A second important characteristic of our model is the session-based nature of the arrival process. First-In-First-Out (FIFO) queues with session-based arrivals are analyzed in [2, 3, 8, 33, 34]. Bruneel [2, 3] and Wittevrongel [33] analyze different aspects of FIFO queues with a session-based arrival process and geometrically distributed session lengths. This model is further extended to generally distributed session lengths by Wittevrongel and Bruneel [33, 34]. De Vuyst et al. [8] further added a second correlation in the model (besides the session nature of the arrival process) by introducing a two-state environment that determines the number of starting sessions. Somewhat related on/off-type arrival models are considered in [10,18,36], also for the FIFO case. Further in [6], sessions consisting of a fixed number of packets are considered in case of an uncorrelated packet arrival process.

In view of the importance of priority scheduling, HOL priority queues with session-based arrivals have been studied as well. Daigle [7] calculates mean session delays in a continuous-time priority queue with session-based arrivals. Our current analysis is a direct extension of the analyses in [5] and [32] where discrete-time HOL priority queues are analyzed with deterministic and geometric session lengths respectively.

## 3 Framework and queueing model

We make extensive use of probability generating functions (pgfs) in this paper. The pgf of a generic discrete random variable $X$ is defined as $X(z) \triangleq \mathrm{E}[z^X]$ with E[.] the expected-value operator. There is a one-to-one map between the probability mass function (pmf) $x(n) \triangleq \mathrm{Prob}[X = n], n \geq 0$ and its pgf $X(z)$, as $X(z)$ is the $z$-transform of the sequence $\{x(n), n \geq 0\}$:

$$X(z) = \sum_{n=0}^{\infty} x(n) z^n. \tag{1}$$

$X(z)$ thus completely characterizes the random variable. Note that $X(1) = 1$. Furthermore, moments of the random variable are easily calculated by means of the moment-generating property of pgfs. For instance, the mean value of a random variable is given by taking the derivative of its pgf in 1: $\mathrm{E}[X] = X'(1)$. It is straightforward to extend the notion of pgfs to the joint pgf of more than one random variable.

We consider a discrete-time single-server system with infinite buffer space. Time is assumed to be slotted. There are two types of sessions, namely sessions of class 1 and sessions of class 2. The numbers of newly generated class-$j$ sessions during consecutive slots are independent and identically distributed (i.i.d.). The numbers of newly generated class-1 and class-2 sessions during slot $k$ are denoted by $b_{1,k}$ and $b_{2,k}$ respectively. Their joint pgf is defined as

$$B(z_1, z_2) \triangleq \mathrm{E}\left[z_1^{b_{1,k}} z_2^{b_{2,k}}\right]. \tag{2}$$

Note that the numbers of sessions of both classes generated during a slot may be correlated. The corresponding marginal pgfs are denoted by $B_j(z)$ ($j = 1, 2$) and are given by $B(z, 1)$ and $B(1, z)$ respectively.

Each class-$j$ session lasts a random number of slots which is assumed generally distributed with pgf $L_j(z)$ and pmf $l_j(i)$, $j = 1, 2$, $i \geq 1$. The packets of a session arrive back to back at the rate of one packet per slot. For further use, we define $p_j(n)$ as the probability that a class-$j$ session that is going on for $n$ slots continues at least one more slot, i.e.,

$$p_j(n) \triangleq \frac{1 - \sum_{i=1}^{n} l_j(i)}{1 - \sum_{i=1}^{n-1} l_j(i)}. \tag{3}$$

The total numbers of class-1 and class-2 packets arriving during slot $k$ are denoted by $a_{1,k}$ and $a_{2,k}$ respectively and their joint pgf is defined as

$$A_k(z_1, z_2) \triangleq \mathrm{E}\left[z_1^{a_{1,k}} z_2^{a_{2,k}}\right]. \tag{4}$$

The transmission times of the packets equal one slot and per slot one packet is transmitted (if there is any).

Packets of class 1 have HOL priority over packets of class 2. This means that as long as there are class-1 packets in the buffer, they are transmitted. A class-2 packet can only be transmitted when there are no class-1 packets present.

On average, $B_j'(1)$ class-$j$ sessions are started in a random slot, each generating, on average, $L_j'(1)$ packets (the mean value of a random variable is given by the first derivative in 1 of the pgf of the variable). Therefore the load generated by class-$j$ packets equals

$$\rho_j = B_j'(1) L_j'(1), \tag{5}$$

$j = 1, 2$. We assume a stable system, i.e., the total load $\rho_T$ is smaller than 1:

$$\rho_T \triangleq \rho_1 + \rho_2 = B_1'(1)L_1'(1) + B_2'(1)L_2'(1) < 1. \tag{6}$$

## 4 Start of the analysis

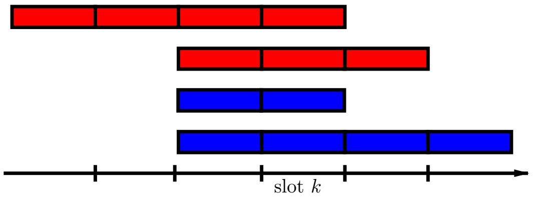
In this section, we first give a Markov-chain description of the system. In a second part, we construct a functional equation that summarizes this Markov chain and that is the starting point of further calculations in the next sections.

### 4.1 Markov-chain description

The arrival process is fully described by the random variables $e_{j,n,k}$, representing the number of class-$j$ sessions that deliver their $n$-th packet during slot $k$. Indeed, the following relationships hold:

$$\begin{aligned} &e_{j,1,k} = b_{j,k}; \\ &e_{j,n+1,k} = \textstyle\sum_{i=1}^{e_{j,n,k-1}} c_{j,n,k-1}^{(i)}, \qquad n \geq 1, \end{aligned} \tag{7}$$

$j = 1, 2$. For a given $n$, the $c_{1,n,k-1}^{(i)}$'s are i.i.d. random variables with values 0 or 1. The same holds for the $c_{2,n,k-1}^{(i)}$'s. The random variable $c_{j,n,k-1}^{(i)}$ equals 1 if and only if the $i$-th



$$e_{1,2,k} = 1, e_{1,4,k} = 1 \Rightarrow a_{1,k} = 2$$

$$e_{2,2,k} = 2 \Rightarrow a_{2,k} = 2$$

$$c^{(1)}_{1,2,k} = 1, c^{(1)}_{1,4,k} = 0$$

$$c^{(1)}_{2,2,k} = 0, c^{(2)}_{2,2,k} = 1$$

**Figure 1. Example illustrating the involved random variables of the arrival process. During slot $k$ two high-priority sessions (red, on top) and two low-priority sessions (blue, bottom) are sending a packet. All non-zero random variables concerning slot $k$ are given.**

active session of class $j$ that has sent the $n$-th packet during slot $k-1$ continues to send a packet in the next slot. The equations (7) can then be understood as follows: $e_{j,1,k}$ represents the number of class-$j$ sessions that deliver their first packet during slot $k$ and therefore equals the new number of sessions that start in that slot. The variable $e_{j,n+1,k}$ corresponds to the number of class-$j$ packets that deliver their $(n+1)$-st packet and therefore equals the number of class-$j$ packets that delivered their $n$-th packet in the previous slot ($e_{j,n,k-1}$) and that are still sending a packet during the current slot.

The variable $a_{j,k}$, the total number of class-$j$ packets arriving during slot $k$, can be expressed as

$$a_{j,k} = \sum_{n=1}^{\infty} e_{j,n,k}, \qquad j = 1, 2. \tag{8}$$

The above defined variables are illustrated in Figure 1.

We further denote the buffer content of class-1 packets and class-2 packets at the beginning of slot $k$ by $u_{1,k}$ and $u_{2,k}$ respectively. The following system equations then directly follow from the HOL priority scheduling of class-1 packets over class-2 packets:

$$\begin{aligned} u_{1,k+1} &= [u_{1,k} - 1]^{+} + a_{1,k}; \\ u_{2,k+1} &= [u_{2,k} - \mathbf{1}_{\mathbf{u_{1,k}=0}}]^{+} + a_{2,k}, \end{aligned} \tag{9}$$

where $[.]^{+}$ denotes the maximum of the argument and 0 and with $\mathbf{1_X}$ the indicator function of $X$ (1 if $X$ is true and 0 if $X$ is false).

A Markovian state description of the system is given by $(e_{1,1,k-1}, e_{1,2,k-1}, \ldots, u_{1,k}, e_{2,1,k-1}, e_{2,2,k-1}, \ldots, u_{2,k})$ and equations (7)-(9) fully describe the behavior of the system.

## 4.2 Construction of the functional equation

We introduce the joint pgf of the state vector:

$$P_k(x_{1,1}, x_{1,2}, \ldots, z_1, x_{2,1}, x_{2,2}, \ldots, z_2) \triangleq \mathrm{E}\left[\prod_{j=1}^{2}\left(\prod_{n=1}^{\infty} x_{j,n}^{e_{j,n,k-1}}\right) z_j^{u_{j,k}}\right]. \tag{10}$$

It follows that

$$\begin{aligned} &P_{k+1}(x_{1,1}, x_{1,2}, \ldots, z_1, x_{2,1}, x_{2,2}, \ldots, z_2) = \\ &\mathrm{E}\left[\left(\prod_{j=1}^{2}\prod_{n=1}^{\infty}(x_{j,n}z_j)^{e_{j,n,k}}\right) z_1^{[u_{1,k}-1]^{+}} z_2^{[u_{2,k}-\mathbf{1}_{\mathbf{u_{1,k}=0}}]^{+}}\right] \\ &= \mathrm{E}\left[(x_{1,1}z_1)^{b_{1,k}} (x_{2,1}z_2)^{b_{2,k}}\right] \\ &\times \left\{ \mathrm{E}\left[\left(\prod_{j=1}^{2}\prod_{n=2}^{\infty}\prod_{i=1}^{e_{j,n-1,k-1}} (x_{j,n}z_j)^{c^{(i)}_{j,n-1,k-1}}\right)\right.\right. \\ &\left. \times\, z_2^{[u_{2,k}-1]^{+}} \mathbf{1}_{\mathbf{u_{1,k}=0}} \right] \\ &+ \mathrm{E}\left[\left(\prod_{j=1}^{2}\prod_{n=2}^{\infty}\prod_{i=1}^{e_{j,n-1,k-1}} (x_{j,n}z_j)^{c^{(i)}_{j,n-1,k-1}}\right)\right. \\ &\left.\left. \times\, z_1^{u_{1,k}-1} z_2^{u_{2,k}} \mathbf{1}_{\mathbf{u_{1,k}>0}} \right]\right\} \\ &= B(x_{1,1}z_1, x_{2,1}z_2) \\ &\times \left\{ \mathrm{E}\left[\left(\prod_{j=1}^{2}\prod_{n=1}^{\infty}(C_{j,n}(x_{j,n+1}z_j))^{e_{j,n,k-1}}\right)\right.\right. \\ &\left. \times\, z_2^{[u_{2,k}-1]^{+}} \mathbf{1}_{\mathbf{u_{1,k}=0}} \right] \\ &+ \mathrm{E}\left[\left(\prod_{j=1}^{2}\prod_{n=1}^{\infty}(C_{j,n}(x_{j,n+1}z_j))^{e_{j,n,k-1}}\right)\right. \\ &\left.\left. \times\, z_1^{u_{1,k}-1} z_2^{u_{2,k}} \mathbf{1}_{\mathbf{u_{1,k}>0}} \right]\right\}, \end{aligned} \tag{11}$$

by using the law of total probability, using system equations (7)-(9) and by taking into account that $b_{1,k}$ and $b_{2,k}$ are statistically independent of the other random variables involved. Here,

$$C_{j,n}(z) \triangleq \mathrm{E}\left[z^{c^{(i)}_{j,n,k-1}}\right] = 1 - p_j(n) + p_j(n)z, \tag{12}$$

$n \geq 1, j = 1, 2$. This follows from the fact that the $c^{(i)}_{j,n,k-1}$'s are Bernoulli-distributed random variables as mentioned before (see Figure 1). We now use the property that a system void of class-$j$ packets at the beginning of a slot implies that no class-$j$ packets arrived in the system during the previous slot. Or in other words, using that $a_{j,k-1} = 0$ - or equivalently that $e_{j,n,k-1} = 0$ for all $n$ - if $u_{j,k} = 0$, we find

$$\begin{aligned}&P_{k+1}(x_{1,1},x_{1,2},..,z_1,x_{2,1},x_{2,2},..,z_2)\\&=\frac{B(x_{1,1}z_1,x_{2,1}z_2)}{z_1z_2}[z_1(z_2-1)P_k(0,\ldots,0)+z_2\times\\&P_k(C_{1,1}(x_{1,2}z_1),C_{1,2}(x_{1,3}z_1),..,z_1,C_{2,1}(x_{2,2}z_2),..,z_2)\\&+(z_1-z_2)P_k(0,..,0,C_{2,1}(x_{2,2}z_2),C_{2,2}(x_{2,3}z_2),..,z_2)].\end{aligned} \tag{13}$$

In steady state, $P_k$ and $P_{k+1}$ both converge to the same limiting function $P$. It then follows from equation (13) that this function must satisfy the following functional equation:

$$\begin{aligned}&P(x_{1,1},x_{1,2},..,z_1,x_{2,1},x_{2,2},..,z_2)\\&=\frac{B(x_{1,1}z_1,x_{2,1}z_2)}{z_1z_2}[z_1(z_2-1)P(0,\ldots,0)+z_2\times\\&P(C_{1,1}(x_{1,2}z_1),C_{1,2}(x_{1,3}z_1),..,z_1,C_{2,1}(x_{2,2}z_2),..,z_2)\\&+(z_1-z_2)P(0,..,0,C_{2,1}(x_{2,2}z_2),C_{2,2}(x_{2,3}z_2),..,z_2)].\end{aligned} \tag{14}$$

The functional equation (14) contains all information concerning the steady-state behavior of the system, although not in transparent form. Nevertheless, several explicit results can be derived from it, which is the subject of the following sections.

For future reference, we end this section with the definition of some joint pgfs concerning the class-1 and the total system content:

$$P_1(x_1,x_2,..,z) \triangleq P(x_1,x_2,..,z,1,..,1), \tag{15}$$

$$\begin{aligned}P_T(x_{1,1},x_{1,2},..,&x_{2,1},x_{2,2},..,z)\\&\triangleq P(x_{1,1},x_{1,2},..,z,x_{2,1},x_{2,2},..,z),\end{aligned} \tag{16}$$

that is, $P_1$ equals $P$ with arguments $x_{2,j}$ (for all $j \geq 1$) and $z_2$ equal to 1 and $P_T$ equals $P$ with arguments $z_1$ and $z_2$ both equal to $z$. The corresponding functional equations are

$$\begin{aligned}P_1(x_1,x_2,..,z) =&\frac{B_1(x_1z)}{z}[(z-1)P_1(0,..,0)\\&+P_1(C_{1,1}(x_2z),C_{1,2}(x_3z),..,z)],\end{aligned} \tag{17}$$

$$\begin{aligned}&P_T(x_{1,1},x_{1,2},..,x_{2,1},x_{2,2},..,z)\\&=\frac{B(x_{1,1}z,x_{2,1}z)}{z}[(z-1)P_T(0,..,0)\\&\quad+P_T(C_{1,1}(x_{1,2}z),C_{1,2}(x_{1,3}z),..,C_{2,1}(x_{2,2}z),..,z)].\end{aligned} \tag{18}$$

## 5 Number of arrivals

Define the joint pgf $E(x_{1,1},x_{1,2},..,x_{2,1},x_{2,2},..)$ as follows:

$$E(x_{1,1},x_{1,2},..,x_{2,1},x_{2,2},..) \triangleq \lim_{k\to\infty} \mathrm{E}\left[\prod_{j=1}^{2}\prod_{n=1}^{\infty} x_{j,n}^{e_{j,n,k}}\right], \tag{19}$$

i.e., it is the joint pgf of the numbers of class-1 and class-2 sessions that deliver their $n$-th packet (for all $n \geq 1$) during an arbitrary slot in steady state. This pgf is given by

$$\begin{aligned}E(x_{1,1},&x_{1,2},..,x_{2,1},x_{2,2},..)\\&=P(x_{1,1},x_{1,2},..,1,x_{2,1},x_{2,2},..,1)\\&=B(x_{1,1},x_{2,1})\\&\quad\times E(C_{1,1}(x_{1,2}),C_{1,2}(x_{1,3}),..,C_{2,1}(x_{2,2}),..).\end{aligned} \tag{20}$$

The last step is found by putting $z_1 = z_2 = 1$ in (14). Successive applications of (20) then lead to the following explicit result for $E$:

$$\begin{aligned}E(x_{1,1},&x_{1,2},..,x_{2,1},x_{2,2},..)\\&=\prod_{n=0}^{\infty} B(g_1^{(n)}(x_{1,n+1}),g_2^{(n)}(x_{2,n+1})),\end{aligned} \tag{21}$$

with

$$g_j^{(n)}(x) \triangleq \sum_{i=1}^{n} l_j(i) + x\left(1-\sum_{i=1}^{n} l_j(i)\right), \tag{22}$$

$j = 1, 2$. To obtain (21), we have used the following relationships, which can easily be derived from (3) and (12):

$$\begin{aligned}&C_{j,1}(C_{j,2}(..C_{j,n}(x)..))\\&\qquad=\sum_{i=1}^{n} l_j(i) + x\left(1-\sum_{i=1}^{n} l_j(i)\right),\end{aligned} \tag{23}$$

$$\lim_{n\to\infty} C_{j,i}(C_{j,i+1}(..C_{j,n}(x)..)) = 1, \qquad i \geq 1, \tag{24}$$

$j = 1, 2$.

The joint pgf of the total numbers of arrivals of both classes during a random slot in steady state is given by

$$\begin{aligned}A(z_1,z_2) =&E(z_1,z_1,..,z_2,z_2,..)\\=&\prod_{n=0}^{\infty} B(g_1^{(n)}(z_1),g_2^{(n)}(z_2)),\end{aligned} \tag{25}$$

which is found from (21). Taking the necessary derivatives of this expression delivers all moments of the class-1, class-2 and total numbers of arrivals per slot in steady state. We find, for instance, that

$$\mathrm{E}[a_j] = B_j'(1)L_j'(1), \tag{26}$$

as expected.

# 6 Buffer content

For general $(x_{1,1}, x_{1,2}, .., z_1, x_{2,1}, x_{2,2}, .., z_2)$, the functional equation (14) is hard to solve. Therefore, we solve it for a specific set of these arguments and discuss how moments of the steady-state buffer content are calculated. We also comment on the consequences of the fact that we are not able to solve the functional equation for general arguments.

## 6.1 Solving the functional equation

We here select only those values of $x_{j,n}$ and $z_j$, $n \geq 1, j = 1, 2$, for which the $P$-functions on both sides of equation (14) have identical arguments (when non-zero), i.e., we choose $x_{j,n} = C_{j,n}(x_{j,n+1}z_j)$ for $j = 1, 2$, $n \geq 1$. By using (3) and (12) in this expression, $x_{j,n}$ can be solved in terms of $z_j$. Denoting this solution by $\chi_{j,n}(z_j)$, we find

$$\chi_{j,n}(z_j) = \frac{\sum_{i=n}^{\infty} l_j(i) z_j^{i-n}}{1 - \sum_{i=1}^{n-1} l_j(i)}, \qquad n \geq 1. \tag{27}$$

In particular, we have that $\chi_{j,1}(z_j) = L_j(z_j)/z_j$ and $\chi_{j,n}(1) = 1$, $n \geq 1$. Choosing $x_{j,n} = \chi_{j,n}(z_j)$ in (14), we obtain

$$\begin{aligned} &P(\chi_{1,1}(z_1), \chi_{1,2}(z_1), .., z_1, \chi_{2,1}(z_2), \chi_{2,2}(z_2), .., z_2) \\ &= \frac{B(L_1(z_1), L_2(z_2))}{z_2\left[z_1 - B(L_1(z_1), L_2(z_2))\right]}[z_1(z_2 - 1)P(0, .., 0) \\ &\quad + (z_1 - z_2)P(0, .., 0, \chi_{2,1}(z_2), \chi_{2,2}(z_2), .., z_2)]. \end{aligned} \tag{28}$$

$P(\chi_{1,1}(z_1), \chi_{1,2}(z_1), .., z_1, \chi_{2,1}(z_2), \chi_{2,2}(z_2), .., z_2)$ can be fully determined by applying Rouché's theorem and the normalization condition, as is e.g. done in [32]. This leads to

$$\begin{aligned} &P(0, .., 0, \chi_{2,1}(z_2), \chi_{2,2}(z_2), .., z_2) \\ &= \frac{Y(z_2)(z_2 - 1)P(0, .., 0)}{z_2 - Y(z_2)}, \end{aligned} \tag{29}$$

$$P(0, .., 0) = 1 - \rho_T \tag{30}$$

and finally

$$\begin{aligned} &P(\chi_{1,1}(z_1), \chi_{1,2}(z_1), .., z_1, \chi_{2,1}(z_2), \chi_{2,2}(z_2), .., z_2) \\ &= (1 - \rho_T)\frac{B(L_1(z_1), L_2(z_2))(z_2 - 1)}{z_1 - B(L_1(z_1), L_2(z_2))}\frac{z_1 - Y(z_2)}{z_2 - Y(z_2)}, \end{aligned} \tag{31}$$

with $Y(z)$ implicitly defined as

$$Y(z) \triangleq B(L_1(Y(z)), L_2(z)), \quad |Y(z)| < 1 \text{ if } |z| < 1. \tag{32}$$

We note that $Y(z)$ is a pgf. As a result $Y(1) = 1$ and all derivatives of $Y$ in 1 can be calculated from (32). The first derivative for instance is given by

$$Y'(1) = \frac{\rho_2}{1 - \rho_1}. \tag{33}$$

By putting $z_1 = z$ in (31) and by substituting $z_2$ by 1 and $z$ respectively, we find

$$P_1(\chi_{1,1}(z), \chi_{1,2}(z), .., z) = (1 - \rho_1)\frac{B_1(L_1(z))(z - 1)}{z - B_1(L_1(z))}, \tag{34}$$

$$\begin{aligned} &P_T(\chi_{1,1}(z), \chi_{1,2}(z), .., \chi_{2,1}(z), \chi_{2,2}(z), .., z) \\ &= (1 - \rho_T)\frac{B(L_1(z), L_2(z))(z - 1)}{z - B(L_1(z), L_2(z))}, \end{aligned} \tag{35}$$

with $P_1$ and $P_T$ defined in (15) and (16), respectively. Note that in order to obtain (34) from (31), l'Hôpital's rule has to be applied. This calculation further needs expression (33) for $Y'(1)$. Expressions (34) and (35) will be used in the next subsection and the following sections.

## 6.2 Calculation of moments

By substitution of $x_{1,n}$ and $x_{2,n}$ ($n \geq 1$) by 1 in expression (14), a functional equation is found for the joint pgf of the buffer contents of both classes. It does not seem to be possible to derive an explicit expression for this pgf from this functional equation. However, all moments of the class-1 and the total buffer content as well as the mean of the class-2 buffer content can be calculated from the results of subsection 6.1. The moments of the class-1 content can be calculated from (17) and (34) by taking appropriate derivatives (for more details on this we refer to [34]). Similarly, the moments of the total buffer content are calculated from (18) and (35). The mean class-2 buffer content is finally calculated as the difference between the mean total buffer content and the mean class-1 content.

Obtaining higher moments of the class-2 buffer content is still an open issue at the moment, since the dependency between the class-1 and class-2 buffer contents influences these. As discussed before, we are not able to characterize this dependency. However, we show in the following section that this does not prohibit us from obtaining the moments of the low-priority packet delay.

# 7 Packet delay

The delay of a packet is defined as the number of slots between the end of the packet's slot of arrival and the end of its departure slot (thus excluding its arrival slot and including its departure slot). Within each class, we assume that packets are transmitted in the order of their arrival. Recall that class-1 packets have HOL priority over class-2 packets. We analyze the class-1 and class-2 packet delays separately in the remainder of this section.

### 7.1 Class-1 packet delay

The analysis of the class-1 packet delay is rather easy once the observation is made that transmission of class-1 packets is not influenced by class-2 packets in the system, due to the HOL priority scheduling discipline. Due to a distributional form of Little's law being applicable here [29], $D_1(z)$, the pgf of the class-1 packet delay in steady state, is expressed in terms of the pgf $P_1(1,..,z)$ of the buffer content of class 1 at the beginning of a random slot, as follows:

$$D_1(z) = \frac{P_1(1,..,z) - 1 + \rho_1}{\rho_1}. \qquad (36)$$

We may thus derive the moments of the class-1 packet delay from the moments of the class-1 system content. We argued in the previous section that we are able to calculate the latter. The mean class-1 packet delay E$[d_1]$ is given by

$$\mathrm{E}[d_1] = D_1'(1) = 1 + \frac{\rho_1 B_1'(1) L_1''(1) + B_1''(1)(L_1'(1))^2}{2\rho_1(1-\rho_1)}. \qquad (37)$$

The mean delay of a high-priority packet is thus influenced by the mean values and the second moments of the class-1 session lengths and of the number of starting sessions of class 1 in a slot.

### 7.2 Class-2 packet delay

The analysis of the steady-state class-2 packet delay is more involved, because of the HOL priority discipline. We tag a random class-2 packet and denote it by $Q_2$. We denote the slot during which $Q_2$ arrives by $S_2$. We first make the following key observation: if a class-1 packet is transmitted before $Q_2$, all packets of the same session of this class-1 packet are transmitted before $Q_2$ as well. Indeed, only other class-1 packets can be transmitted between the transmissions of two randomly chosen packets of a *same* class-1 session.

Furthermore, we denote the number of class-1 sessions that have sent their $n$-th packet during slot $S_2$ by $e^*_{1,n}$, and the total system content at the beginning of the following slot by $u^*_T$. Furthermore, let $r_2$ indicate the number of packets arriving during slot $S_2$ and to be transmitted after packet $Q_2$. Before writing down an expression for and analyzing the delay of $Q_2$, we first concentrate on the *virtual delay* $w_2$ of $Q_2$. This virtual delay is here defined as the delay when no *new sessions* are generated after slot $S_2$. Then $w_2$ equals

$$w_2 = u^*_T - r_2 + \sum_{n=1}^{\infty} \sum_{i=1}^{e^*_{1,n}} l^+_{1,n,i}, \qquad (38)$$

with $l^+_{1,n,i}$ the number of packets arriving after slot $S_2$ of the $i$-th class-1 session that generated its $n$-th packet during slot $S_2$. The virtual delay thus equals the superposition of the buffer content just after slot $S_2$ and to be transmitted no later than $Q_2$ and the packets that arrive after slot $S_2$ of class-1 sessions which were already generating a packet during slot $S_2$. Note that the $l^+_{1,n,i}$'s are all independent of the system state just after slot $S_2$. Their pgf is given by $\chi_{1,n}(z)$ (see (27)). With the definition

$$Q(x_1, x_2, .., y, z) \triangleq \mathrm{E}\left[\left(\prod_{n=1}^{\infty} x_n^{e^*_{1,n}}\right) y^{r_2} z^{u^*_T}\right], \qquad (39)$$

expression (38) leads to the pgf of $w_2$:

$$W_2(z) \triangleq \mathrm{E}[z^{w_2}] = Q(\chi_{1,1}(z), \chi_{1,2}(z), .., 1/z, z). \qquad (40)$$

Relating the buffer content distribution just after the arrival slot of a random class-2 packet to the buffer content distribution at the beginning of a random slot (i.e., a manifestation of the typical renewal-theory paradox, see e.g. [23]), we find

$$Q(x_1, x_2, .., y, z) = \frac{P_T(x_1, x_2, .., 1, .., z) - P_T(x_1, x_2, .., y, .., z)}{\rho_2(1-y)}, \qquad (41)$$

with $P_T$ the function analyzed in section 6.

We now relate the delay $d_2$ and the virtual delay $w_2$ of packet $Q_2$. Obviously, the virtual delay is an integral part of the delay. During the transmission of a certain packet belonging to the virtual delay workload, say packet $P$, new class-1 sessions may be generated, the transmission of their packets adding to the delay of $Q_2$. During the transmission of the packets of these class-1 sessions new class-1 sessions may in turn be generated, which further add to the delay of $Q_2$, etc. The total number of all packets of all these sessions (including packet $P$ itself) is called the *sub-busy period* initiated by $P$. Summarizing, we can write

$$d_2 = \sum_{i=1}^{w_2-1} v_{1,i} + 1, \qquad (42)$$

with $v_{1,i}$ the sub-busy period added by the $i$-th packet of the virtual delay workload. Note that these $v_{1,i}$'s are all i.i.d. and their common pgf is denoted by $V_1(z)$. By $z$-transforming expression (42), we then obtain

$$D_2(z) \triangleq \mathrm{E}[z^{d_2}] = \frac{zW_2(V_1(z))}{V_1(z)}. \qquad (43)$$

Using (40), we find

$$D_2(z) = \frac{zQ(\chi_{1,1}(V_1(z)), \chi_{1,2}(V_1(z)), .., 1/V_1(z), V_1(z))}{V_1(z)}. \qquad (44)$$

The use of (41) in the latter expression provides us with an expression for $D_2(z)$ in terms of the $P_T$-function and $V_1(z)$. The $P_T$-function is characterized by (18) and (35). So what remains is the calculation of the function $V_1$.

In order to do this, we note that the $v_{1,i}$'s in expression (42) can be expressed as

$$v_{1,i} = 1 + \sum_{m=1}^{b_{1,i}} \sum_{n=1}^{l_{1,i}^{(m)}} v_{1,i}^{(m,n)}, \tag{45}$$

with $b_{1,i}$ the number of new class-1 sessions generated during the transmission of the $i$-th packet of the virtual delay workload, $l_{1,i}^{(m)}$ the number of packets the $m$-th session of $b_{1,i}$ contains and $v_{1,i}^{(m,n)}$ the sub-busy period initiated by the $n$-th packet of the $m$-th session of $b_{1,i}$. Indeed, a sub-busy period initiated by a packet consists of the transmission slot of that packet and the sub-busy periods of all packets of all sessions that are generated during that slot. Note that the $v_{1,i}^{(m,n)}$'s are i.i.d. having the same pgf as the $v_{1,i}$'s, i.e., $V_1$. Expression (45) then leads to the following implicit expression for $V_1$:

$$V_1(z) = zB_1(L_1(V_1(z))). \tag{46}$$

Although this does not lead to an explicit formula for $V_1$, its derivatives in 1 can be explicitly calculated due to the knowledge that $V_1(1) = 1$, since $V_1$ is a pgf.

Expression (44) combined with expressions (41) and (46) enables us to calculate the moments of the class-2 packet delay as functions of (partial) derivatives of the $P_T$-function, evaluated for all arguments equal to 1. We have argued in the previous section that these derivatives can be calculated. In general, the calculations of the moments of the class-2 delay are however highly complex, since several partial derivatives of $P_T$ have to be calculated, which is a non-trivial task. For instance, the first derivative of expression (44) evaluated in $z = 1$ leads to an expression containing (partial) derivatives of $\chi_{1,m}$, $V_1$ and $P_T$. These derivatives can in turn be calculated from expressions (27), (46) and (18) and (35) respectively. The following final expression for the mean class-2 packet delay can then be obtained

$$\mathrm{E}[d_2] = D_2'(1) = 1 + \frac{\rho_T L_2''(1)}{2L_2'(1)(1-\rho_T)} + \frac{B_2''(1)L_2'(1)}{2B_2'(1)(1-\rho_T)} + \frac{\frac{\partial^2 B}{\partial z_1 \partial z_2}(1,1)L_1'(1)}{B_2'(1)(1-\rho_T)} + \frac{B_1'(1)L_1''(1) + B_1''(1)(L_1'(1))^2}{2(1-\rho_1)(1-\rho_T)}. \tag{47}$$

The mean low-priority packet delay is thus influenced by the mean values and the second moments of the class-1 and class-2 session lengths and of the number of starting sessions of class 1 and class 2 in a slot. It further depends on the covariance between the number of class-1 and class-2 starting sessions in a slot (through $\frac{\partial^2 B}{\partial z_1 \partial z_2}(1,1)$).

Higher moments of the class-2 packet delay can be calculated as well.

## 8 Numerical examples

Our results can be used by practitioners to estimate the (mean) delay that high- and low-priority packets sustain in a particular network node. The influence of the correlation in the arrival process on the mean delays can also be characterized.

We illustrate our findings by means of a numerical example. We assume that class-1 and class-2 sessions are both generated according to independent Poisson processes with means $\lambda_1$ and $\lambda_2$ respectively. We thus have

$$B(z_1, z_2) = e^{\lambda_1(z_1-1)} e^{\lambda_2(z_2-1)}. \tag{48}$$

We are primarily interested in the influence of the variability of the session lengths on the performance of the system, i.e. on the mean packet delays of both classes (for the influence of the mean session lengths we refer to [5, 32]). Therefore, we firstly consider the example of negative binomially distributed class-$j$ session lengths with parameters $m_j$ and $p_j$, i.e., with pgf

$$L_j(z) = \left(\frac{p_j z}{1-(1-p_j)z}\right)^{m_j}. \tag{49}$$

By decreasing $m_j$ while keeping $\mathrm{E}[l_j] = L_j'(1) = m_j/p_j$ constant, the variance of the session lengths $\mathrm{Var}[l_j] = m_j(1-p_j)/p_j^2$ can be increased while the mean value is kept constant. It may be noted that $m_j = 1$ corresponds to a geometric distribution, while $p_j = 1$ corresponds to deterministic session lengths.

Throughout this section, we consider the high-priority load to be a quarter of the total load, i.e., $\alpha \triangleq \rho_1/\rho_T = 0.25$. The means of the session lengths equal 16 slots for both classes.

In Figure 2 (Figure 3 respectively), we depict the mean delays of packets of both classes as functions of the total load $\rho_T$ when $m_2 = 2$ ($m_1 = 2$ respectively) and for varying $m_1$ ($m_2$ respectively). Firstly, it can be concluded from these figures that priority scheduling indeed differentiates the delay characteristics of both classes. Secondly, we see that the mean delays of packets are influenced by the variance of the session lengths of their own class. Thirdly, it is shown that the mean delay of low-priority packets is also influenced by the variance of the high-priority session lengths, although not as much as by the variance of the lengths of

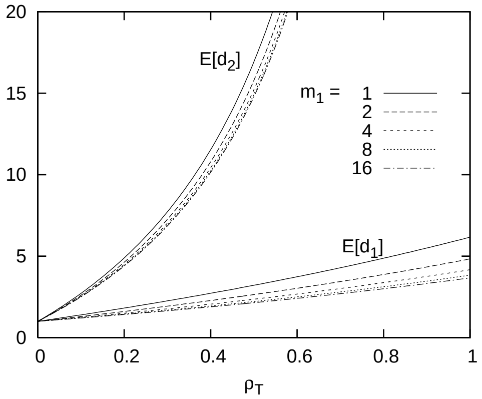


**Figure 2. Mean packet delays of both classes versus the total load for $\alpha = 0.25$, E$[l_1] = 16$, E$[l_2] = 16$ and $m_2 = 2$. Higher $m_1$ means a lower variance of the session lengths of class 1.**

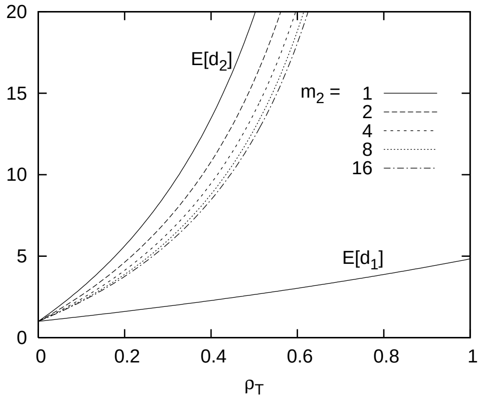


**Figure 3. Mean packet delays of both classes versus the total load for $\alpha = 0.25$, E$[l_1] = 16$, E$[l_2] = 16$ and $m_1 = 2$. Higher $m_2$ means a lower variance of the session lengths of class 2.**

the sessions of its own class. Obviously, the high-priority packet delay does not depend on the low-priority arrival process.

In the first two figures, we showed the mean delays when the variance of the session lengths was less than or equal to the variance of geometrically distributed session lengths (with the same mean value). To conclude, we show the impact of higher variances of the session lengths in Figures 4 and 5. In Figure 4, the class-2 session lengths are geometrically distributed, while the variance of the class-1 session lengths is assumed to equal $K_1(16^2 - 16)$. The relative deviation of the mean class-2 packet delay, defined as $(\mathrm{E}[d_2]_{K_1=K} - \mathrm{E}[d_2]_{K_1=1})/\mathrm{E}[d_2]_{K_1=1}$, is plotted for several values of $K$. Note that the reference case $K = 1$ corresponds to the geometric distribution. The case $K = 0$ corresponds to the deterministic case while $K > 1$ corresponds to distributions that have a larger variance than the geometric one. Note that a variance with $K > 1$ can easily be constructed by using a mix of geometric distributions. In Figure 5, the class-1 session lengths are geometrically distributed and the variance of the class-2 session lengths is assumed to equal $K_2(16^2 - 16)$. Now, the relative deviation $(\mathrm{E}[d_2]_{K_2=K} - \mathrm{E}[d_2]_{K_2=1})/\mathrm{E}[d_2]_{K_2=1}$ of the mean class-2 packet delay is plotted for several values of $K$. From both plots, it is once again concluded that the variances of the class-1 and class-2 session lengths have a non-negligible impact on the mean class-2 delay. Furthermore, we conclude from Figure 5 that in this case $\mathrm{E}[d_2]_{K_2=K} = C(K).\mathrm{E}[d_2]_{K_2=1}$, with $C(K)$ nearly independent of the total load when the load is high. This is not the case when the high-priority lengths are varied. A linear relation between the relative deviation and $K$ can still be envisaged though.

## 9 Performance of an E-commerce web server

We consider an E-commerce web server. Users request files and the web server responds by sending the requested files to the users. Two types of content are stored on the web server, content that provides revenues (class 1) and content that does not (class 2). We apply our model on the situation described in [16] and depicted in Figure 6. The web server is connected to the Internet through a gateway, which is considered the bottleneck. In the gateway, a buffer is therefore installed and packets of class 1 are transmitted, via the output channel, with priority over class-2 packets. Our analysis is used to calculate the mean delay that packets sustain in the gateway.

We use the model from this paper to analyze the performance of the web server. Therefore, we first assign values to some relevant model parameters. We assume that the output channel of the gateway has a bandwidth of 100 Mbit/s. Likewise, the packets of each session are transferred by the web server to the gateway at the rate of 100 Mbit/s. We assume further that each packet contains 100 bytes. Since it takes exactly one slot to transmit a packet, the slot length equals 8 $\mu$s. Sessions correspond to the requested files. The session length (i.e., the file sizes) distribution is taken from a real trace. The trace can be found at http://ita.ee.lbl.gov/html/contrib/EPA-HTTP.html[1], and contains the recordings of web requests of one day. We have rounded the byte sizes to the nearest multiple of 100 Bytes. The mean session size then equals 8502

[1] The logs were collected by Laura Bottomley (laurab@ee.duke.edu)

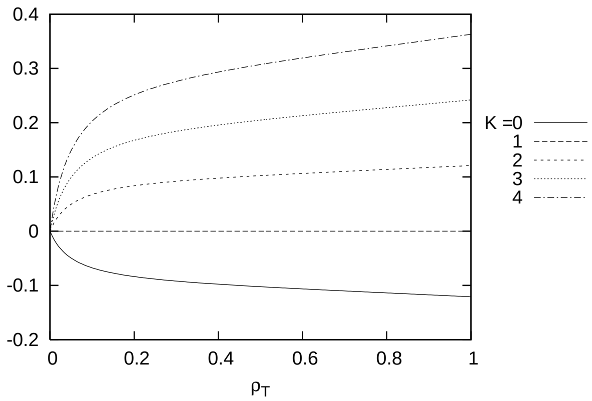


**Figure 4. Relative deviation of the mean class-2 delay versus the total load for $\alpha = 0.25$, E$[l_j] = 16$, Var$[l_j] = K_j(16^2 - 16)$, $j = 1, 2$ and $K_1 = K, K_2 = 1$.**

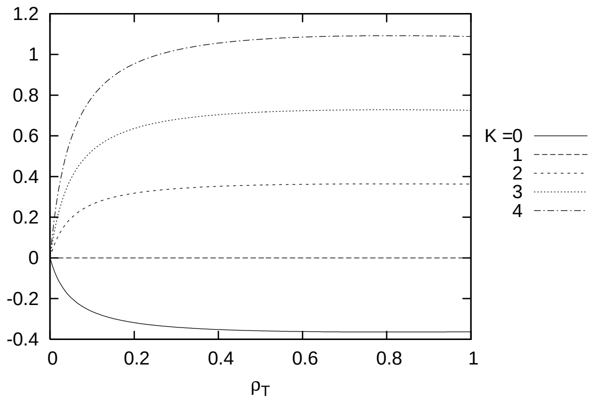


**Figure 5. Relative deviation of the mean class-2 delay versus the total load for $\alpha = 0.25$, E$[l_j] = 16$, Var$[l_j] = K_j(16^2 - 16)$, $j = 1, 2$ and $K_1 = 1, K_2 = K$.**

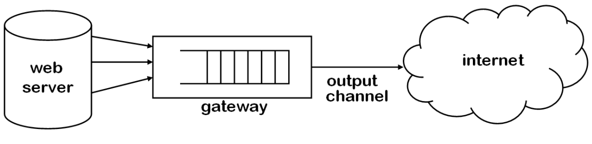


**Figure 6. Conceptual scheme of a web server connected to the Internet through a gateway**

| | E$[d_1]$ | E$[d_2]$ | FIFO |
|---|---|---|---|
| $\alpha = 0.25$ | 9.724 | 16.622 | 14.897 |
| $\alpha = 0.5$ | 11.448 | 18.347 | 14.897 |
| $\alpha = 0.75$ | 13.173 | 20.072 | 14.897 |

**Table 1. Mean class-1 and class-2 packet delays (in $\mu$s) in the E-commerce web server for some values of $\alpha$. The packet delay in case of FIFO is included as reference value.**

Bytes, with a variance of 5.004e9. We assume that the numbers of requests during a slot are distributed according to a Poisson process. The trace exists of 36677 (valid) requests over 24 hours, which leads to a mean number of 3.396e-6 requests per slot. Finally, we assume that each request is of class 1 with probability $\alpha$ and of class 2 with probability $1 - \alpha$ (independent of other requests).

In a first scenario, we assume that the file sizes of both classes have the same distribution, i.e., the distribution calculated from the trace. In Table 1, some values of the mean packet delays of both classes are given for three different values of $\alpha$. As a reference value, we have also added the mean packet delay when FIFO scheduling is implemented instead of priority scheduling. In the latter case, the mean delay is independent of the class and of $\alpha$, and equals about 7 $\mu$s more than the transmission time of a packet (8 $\mu$s). Thus, on average, a packet has to wait 7 $\mu$s in the gateway. It is seen that this value can be reduced to about 2 $\mu$s by giving priority to requests that provide revenues if these requests are only a small part (a quarter) of the total number of requests, and to about 5 $\mu$s if it constitutes a big part (3/4) of the requests. Of course, the price to pay is an increase of the mean low-priority packet delay, namely about 2 $\mu$s more for $\alpha = 0.25$ and more than 5 $\mu$s extra for $\alpha = 0.75$.

In a second scenario, we split the trace into two groups: group A contains all request files with size smaller than or equal to 1900 Bytes and group B consists of the request files that are larger than 1900 Bytes (1900 Bytes is the median of the request file size distribution, so groups A and B approximately exist out of the same number of requests). The request file sizes of group A have a mean of 734 Bytes and a variance of 2.453e5, while those of group B have a mean of 16369 Bytes and a variance of 9.950e9. In Table 2, we show the mean delay of both priority classes for two different cases: (a) class 1 equals group A and class 2 equals group B (small request files have priority), and (b) class 1 equals group B and class 2 equals group A (large request files have priority). We conclude that the advantage of a priority scheduling is much larger when the high-priority request files are generally small. Furthermore, giving priority is especially advantageous to packets of small request

| | E$[d_1]$ | E$[d_2]$ |
|---|---|---|
| class 1 = group A | 8.001 | 15.254 |
| class 1 = group B | 14.897 | 14.912 |

**Table 2. Mean class-1 and class-2 packet delays (in $\mu$s) in the E-commerce web server for some class-dependent distributions of the packet sizes.**

files: there is a difference of almost 7 $\mu$s between both cases for the mean packet delay of group-A packets, while there is only a minor difference of $.4$ $\mu$s for the mean packet delay of packets of group B.

## 10 Conclusion

In this paper, we studied a discrete-time two-class priority queue with a two-layered arrival process. Packets of variable-length sessions of both classes arrive to the system at the rate of one packet per slot. The session lengths of both classes can have general distributions and these distributions can be different for both classes. Since the arrival process is fairly general, the analysis is obviously nontrivial. Using probability generating functions, we have shown that explicit closed-form expressions for the mean values of the system contents and packet delays of both classes can be derived, as well as higher moments for the packet delays of both classes. We have shown the influence of the variance of the session lengths of both classes on the mean (low-priority) packet delay through numerical examples. We have finally applied our results to an E-commerce web server and showed how the performance of such a web server can be predicted by means of the results of our analysis.

Our main qualitative conclusions are: (i) give priority to only a small fraction of the requests to the web server, i.e., only to those applications that generate the largest revenues, and (ii) giving priority to applications with small request file sizes is more effective than giving priority to time-consuming applications.

This research can be extended in different ways. A non-exhaustive list is a) the calculation of tail probabilities of the packet delay, which is non-trivial for priority queues, see e.g. [20, 21]; b) the extension to more than two priority classes; and c) the analysis of a model where the packets in a session do not necessarily arrive back to back, which would highly complicate the analysis since we used this assumption several times in this paper.

**Acknowledgment.** The first author is Postdoctoral Fellow with the Research Foundation, Flanders (F.W.O.-Vlaanderen), Belgium.

## References

[1] J. Walraevens, S. Wittevrongel, and H. Bruneel. Analysis of priority queues with session-based arrival streams. In *Proceedings of the Seventh International Conference on Networking (ICN 2008)*, pages 503–510, Cancun, April 2008.

[2] H. Bruneel. Packet delay and queue length for statistical multiplexers with low-speed access lines. *Computer Networks and ISDN Systems*, 25(12):1267–1277, 1993.

[3] H. Bruneel. Calculation of message delays and message waiting times in switching elements with slow access lines. *IEEE Transactions on Communications*, 42(2/3/4):255–259, 1994.

[4] J. Chang and Y. Harn. A discrete-time priority queue with two-class customers and bulk services. *Queueing Systems*, 10:185–212, 1992.

[5] B. Choi, D. Choi, Y. Lee, and D. Sung. Priority queueing system with fixed-length packet-train arrivals. *IEE Proceedings-Communications*, 145(5):331–336, 1998.

[6] I. Cidon, A. Khamisy, and M. Sidi. Delay, jitter and threshold crossing in ATM systems with dispersed messages. *Performance Evaluation*, 29(2):85–104, 1997.

[7] J. Daigle. Message delays at packet-switching nodes serving multiple classes. *IEEE Transactions on Communications*, 38(4):447–455, 1990.

[8] S. De Vuyst, S. Wittevrongel, and H. Bruneel. Statistical multiplexing of correlated variable-length packet trains: an analytic performance study. *Journal of the Operational Research Society*, 52(3):318–327, 2001.

[9] T. Demoor, J. Walraevens, D. Fiems, and H. Bruneel. Mixed finite-/infinite-capacity priority queue with interclass correlation. In *Proceedings of the 15th International Conference on Analytical and Stochastic Modelling Techniques and Applications (ASMTA 2008), LNCS 5055*, pages 61–74, Nicosia, 2008.

[10] K. Elsayed and H. Perros. The superposition of discrete-time Markov renewal processes with an application to statistical multiplexing of bursty traffic sources. *Applied Mathematics and Computation*, 115(1):43–62, 2000.

[11] D. Fiems, J. Walraevens, and H. Bruneel. Performance of a partially shared priority buffer with correlated arrivals. In *Proceedings of the 20th International Teletraffic Congress (ITC20), LNCS*, volume 4516, pages 582–593, Ottawa, 2007.

[12] R. Guérin and V. Peris. Quality-of-service in packet networks: basic mechanisms and directions. *Computer Networks*, 31(3):169–189, 1999.

[13] O. Hashida and Y. Takahashi. A discrete-time priority queue with switched batch Bernoulli process inputs and constant service time. In *Proceedings of ITC 13*, pages 521–526, Copenhagen, 1991.

[14] G. Heijenk, M. E. Zarki, and I. Niemegeers. Modelling of segmentation and reassembly processes in communication networks. In *Proceedings of ITC14*, pages 513–524, Antibes, 1994.

[15] L. Hoflack, S. De Vuyst, S. Wittevrongel, and H. Bruneel. Analytic traffic model of web server. *Electronics Letters*, 44(1), 2008.

[16] L. Hoflack, S. De Vuyst, S. Wittevrongel, and H. Bruneel. Modeling web server traffic with session-based arrival streams. In *Proceedings of the 15th international conference on analytical and stochastic modelling techniques and applications (ASMTA 2008), LNCS 5055*, pages 47–60, Nicosia, 2008.

[17] H. Inai and J. Yamakita. A two-layer queueing model to predict performance of packet transfer in broadband networks. *Annals of Operations Research*, 79:349–371, 1998.

[18] F. Kamoun. Performance analysis of a discrete-time queuing system with a correlated train arrival process. *Performance Evaluation*, 63(4-5):315–340, 2006.

[19] A. Khamisy and M. Sidi. Discrete-time priority queues with two-state markov modulated arrivals. *Stochastic Models*, 8(2):337–357, 1992.

[20] K. Laevens and H. Bruneel. Discrete-time multiserver queues with priorities. *Performance Evaluation*, 33(4):249–275, 1998.

[21] T. Maertens, J. Walraevens, and H. Bruneel. Priority queueing systems: from probability generating functions to tail probabilities. *Queueing Systems*, 55(1):27–39, 2007.

[22] M. Mehmet Ali and X. Song. A performance analysis of a discrete-time priority queueing system with correlated arrivals. *Performance Evaluation*, 57(3):307–339, 2004.

[23] I. Mitrani. *Modelling of Computer and Communication Systems*. Cambridge University Press, Cambridge, 1987.

[24] J. Roberts. Internet traffic, QoS, and pricing. *Proceedings of the IEEE*, 92(9):1389–1399, 2005.

[25] S. Shakkottai and R. Srikant. Many-sources delay asymptotics with applications to priority queues. *Queueing Systems*, 39(2-3):183–2000, 2001.

[26] T. Takine, B. Sengupta, and T. Hasegawa. An analysis of a discrete-time queue for broadband ISDN with priorities among traffic classes. *IEEE Transactions on Communications*, 42(2-4):1837–1845, 1994.

[27] T. Tan, K. Moinzadeh, and V. Mookerjee. Optimal processing policies for an e-commerce web server. *INFORMS Journal on Computing*, 17(1):99–110, 2005.

[28] J. Van Velthoven, B. Van Houdt, and C. Blondia. The impact of buffer finiteness on the loss rate in a priority queueing system. *Lecture Notes in Computer Science*, 4054:211–225, 2006.

[29] B. Vinck and H. Bruneel. A note on the system contents and cell delay in FIFO ATM-buffers. *Electronics Letters*, 31(12):952–954, 1995.

[30] J. Walraevens, D. Fiems, and H. Bruneel. Time-dependent performance analysis of a discrete-time priority queue. *Performance Evaluation*, 65(9):641–652, 2008.

[31] J. Walraevens, B. Steyaert, and H. Bruneel. Performance analysis of a single-server ATM queue with a priority scheduling. *Computers & Operations Research*, 30(12):1807–1829, 2003.

[32] J. Walraevens, S. Wittevrongel, and H. Bruneel. A discrete-time priority queue with train arrivals. *Stochastic Models*, 23(3):489–512, 2007.

[33] S. Wittevrongel. Discrete-time buffers with variable-length train arrivals. *Electronics Letters*, 34(18):1719–1721, 1998.

[34] S. Wittevrongel and H. Bruneel. Correlation effects in ATM queues due to data format conversions. *Performance Evaluation*, 32(1):35–56, 1998.

[35] F. Xabier Albizuri, M. Graña, and B. Raducanu. Statistical transmission delay guarantee for nonreal-time traffic multiplexed with real-time traffic. *Computer Communications*, 26(12):1365–1375, 2003.

[36] Y. Xiong and H. Bruneel. Buffer behavior of statistical multiplexers with correlated train arrivals. *International Journal of Electronics and Communications (AEÜ)*, 51(3):178–186, 1997.

[37] T. Yu and K. Lin. QCWS: an implementation of QoS-capable multimedia web services. *Multimedia Tools and Applications*, 30(2):165–187, 2006.

# Congestion Control of a Single Router With an Active Queue Management

Yassine Ariba Yann Labit Frédéric Gouaisbaut
CNRS; LAAS; 7, avenue du Colonel Roche, F-31077 Toulouse, France.
Université de Toulouse, UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France.
{yariba, ylabit, fgouaisb}@laas.fr

## Abstract

*Several works have shown the link between congestion control in communication networks and feedback control system. This paper is an extended version of [1] and proposes the design of an Active Queue Management (AQM) that ensures the congestion control stability. To this end, tools from control theory, and especially in a time delay systems framework, are considered. We aim at stabilizing the Transmission Control Protocol (TCP) as well as the queue length of the congested router. Furthermore, the control mechanism is then completed to deal with the stability issue under some non-responsive crossing traffic modeled as perturbation. Finally, a numerical example and simulations via the Network Simulator NS support our study.*

*Keywords: Active Queue Management, congestion control, control theory, time delay system, networks.*

## I. Introduction

Congestion control consists in adjusting data flow rates sent by end users into the network based on the network load status. Since the congestion avoidance algorithm of Jacobson [2], it has motivated a huge amount of work aiming at understanding the congestion phenomenon and achieving better performances in terms of *Quality of Service* (QoS). As a matter of fact, there has been a growing recognition that the network itself must participate in congestion control and ressource management [3], [4].

The AQM principle consists in dropping (or marking when ECN, *Explicit Congestion Notification* [5] option is enabled) some packets before buffer saturates. Hence, following the *Additive-Increase Multiplicative-Decrease* (AIMD) behavior of TCP, sources reduce their congestion window size avoiding then the full saturation of the router. Basically, AQM support TCP for congestion avoidance and feedback to the latter when traffic is too heavy. Indeed, an AQM drops/marks incoming packet with a given probability related to a congestion index (such as queue length or delays) allowing then a kind of control on the buffer occupancy at routers. Various mechanisms have been proposed in the literature such as Random Early Detection (RED) [6], Random Early Marking (REM) [7], Adaptive Virtual Queue (AVQ) [8] and many others [9]. Their performances have been evaluated in [9] and empirical studies have shown their effectiveness [4]. A study proposed by [10] have redesigned the AQMs using control theory and *PI* (Proportional and Integral) have been developed in order to address the packet dropping strategy issue. Then, using dynamical model of TCP developed by [11], many researches have been devoted to deal with congestion problem in a control theory framework (for example see [12], [13], [9], [14] and references therein). Nevertheless, most of these papers do not take into account the delay and ensure the stability in closed-loop for all possible delays which could be conservative in practice.

The study of congestion control in a time delay system framework is not new and has been successfully exploited (see for example [15], [16], [17], [18], [19]). The global stability analysis of TCP has been addressed in [20], [15] through the Lyapunov-Krasovskii theory. But no constructive algorithm is proposed to embed a control on routers. In [19], a delay dependent state feedback controller is provided by compensation of the delay with a memory feedback control. This latter methodology is interesting in theory but hardly suitable in practice. At last, all these papers deal with the congestion control stability considering constant delays requiring then restrictive assumptions.

In this paper, we focus in regulating buffer queue length of a congested router as well as rate at which TCP sources send data into the network. The proposed control has the objectives to ensure QoS, to avoid severe congestion and to maintain a prescribed *Round Trip Time* (RTT) with a low delay jitter. Stability of communications, guaranteed by an AQM, is proved through the Lyapunov method. It is worthy to note that unlike most of the studies in the literature, we take into account the time-varying nature of the RTT. The packet dropping strategy computed by the AQM is designed as a state feedback for time-varying delay systems based on a recently developed Lyapunov-Krasovskii functional [21]. Then, the methodology is extended to cope with additional non-responsive crossing traffics (like for example UDP and ICMP). Indeed, non-TCP traffics are not reactive to packet dropping and may affect the equilibrium of the communications. Hence, the second proposed control law stabilizes the TCP network (queue length and rates) to a desired equilibrium in spite of the presence of some non-responsive traffics, ensuring then a level of QoS.

The paper is organized as follows. The second part presents the model of a network supporting TCP and the time delay system representation. Section III is dedicated to the design of the AQM ensuring the stabilization of TCP. Section IV presents application of the exposed theory and simulation results using NS-2 [22].

*Notations:* For two symmetric matrices, $A$ and $B$, $A >$ $(\geq)$ $B$ means that $A-B$ is (semi-) positive definite. $A^T$ denotes the transpose of $A$. $1_n$ and $0_{m\times n}$ denote respectively the identity matrix of size $n$ and null matrix of size $m\times n$. If the context allows it, the dimensions of these matrices are often omitted. At last, for a given matrix $B \in \mathrm{R}^{m\times n}$ such that $rank(B) = r$, we define $B^{\perp} \in \mathrm{R}^{n\times(n-r)}$ the right orthogonal complement of $B$ by $BB^{\perp} = 0$.

## II. Problem statement

This second section is dedicated to the modeling of a network supporting TCP and the time delay system representation.

### A. The linearized fluid-flow model of TCP

We consider a network consisting of $N$ homogeneous TCP sources (i.e with the same propagation delay) connected to destination nodes through a single router (see Figure 1). The bottleneck link is shared by $N$ flows and TCP applies the well known congestion avoidance algorithm to cope with the phenomenon of congestion collapse [2]. Many studies have been dedicated to the modeling of TCP and its AIMD (*Additive-Increase Multiplicative-Decrease*) behavior (see [12], [13], [14] and references therein). We consider in this note the model (1) developed by [11] widely used for automatic control purpose [12]. This latter may not capture with high accuracy the dynamic behavior of TCP but its simplicity allows us to apply our methodology. Let us consider the following model

$$\left\{\begin{array}{rcl} \dot{W}(t) & = & \frac{1}{\frac{q(t)}{C}+T_p} - \frac{W(t)W(t-R(t))}{2\frac{q(t-R(t))}{C}+T_p} p(t-R(t)) \\ \dot{q}(t) & = & \frac{W(t)}{\frac{q(t)}{C}+T_p} N - C + d(t) \end{array}\right. \tag{1}$$

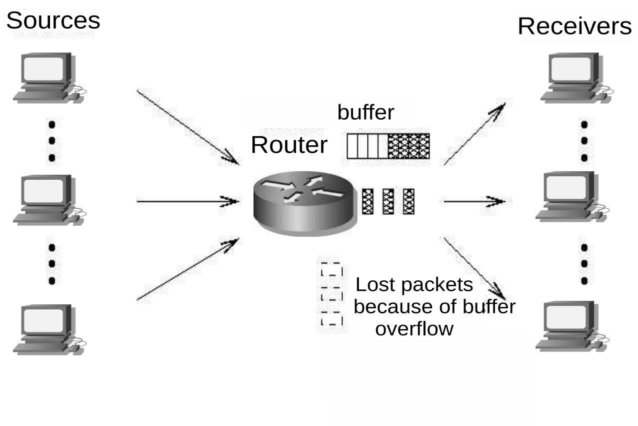


**Fig. 1. Network configuration**

where $W(t)$ is the TCP window size, $q(t)$ is the queue length of the router buffer, $R(t)$ is the round trip time (RTT) and can be expressed as $R(t) = q(t)/C + T_p$. $C$, $T_p$ and $N$ are parameters related to the network configuration and represent the transmission capacity of the router, the propagation delay and the number of TCP sessions respectively. The variable $p$ is the marking/dropping probability of a packet (depending whether the ECN option, is enabled, see [5]). In the mathematical model (1), we have introduced an additional signal $d(t)$ which models cross traffics through the router, filling the buffer. These traffics are not TCP based flows (not modeled in TCP dynamics) and can be viewed as perturbations since they are not reactive to packets dropping (for example, UDP traffic). Note that the model (1) is non linear and thus difficult to handle. Consequently, defining the set of equilibrium points $(W_0, q_0, p_0)$ by

$$\left\{\begin{array}{l} \dot{W} = 0 \;\Rightarrow\; W_0^2 p_0 = 2, \\ \dot{q} = 0 \;\Rightarrow\; W_0 = \frac{R_0 C}{N},\; R_0 = \frac{q_0}{C} + T_p, \end{array}\right. \tag{2}$$

TCP model can be linearized as follows

$$\left\{\begin{array}{l} \delta\dot{W}(t) = -\frac{N}{R_0^2 C}\Big(\delta W(t) + \delta W(t-R(t))\Big) \\ \qquad -\frac{1}{R_0^2 C}\Big(\delta q(t) - \delta q(t-R(t))\Big) - \frac{R_0 C^2}{2N^2}\delta p(t-R(t)) \\ \delta\dot{q}(t) = \frac{N}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t) + d(t) \end{array}\right. \tag{3}$$

where $\delta W \doteq W - W_0$, $\delta q \doteq q - q_0$ and $\delta p \doteq p - p_0$ are the signals variations around the operating point [10]. This approximation is valid as long as signals $\delta W(t)$, $\delta q(t)$ and $\delta p(t)$ remain small enough. Regulation is required in order to regulate quantities $\delta W(t)$ and $\delta q(t)$ around zero (and thus the congestion phenomenon), we have to control the dropping probability $\delta p(t)$. This probability is computed the help of an AQM, playing thus the role of a controller. In

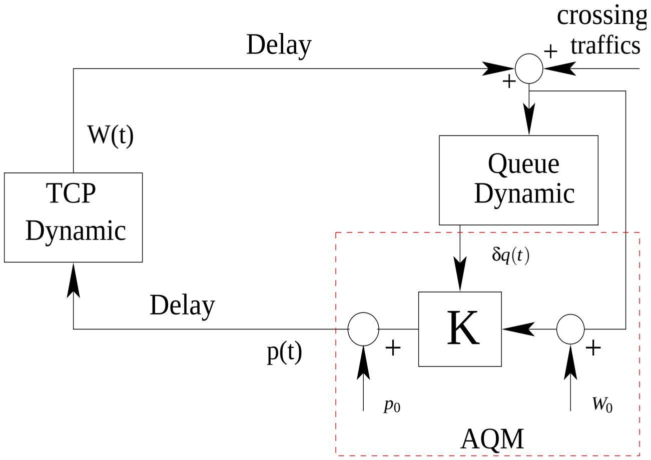


**Fig. 2. Design of an AQM as a state feedback**

this paper, this regulation problem is addressed in Section III with the design of a stabilizing state feedback for time-varying delay systems. Consequently, based on [23] and considering a state feedback, the queue management strategy of the drop probability will be expressed as (see Figure 2):

$$p(t) = p_0 + k_1\delta W(t) + k_2\delta q(t). \quad (4)$$

Scalars $k_1$ and $k_2$ are the components of the matrix gain $K$ which have to be designed to ensure the stability of the overall system.

## B. Time delay system model

The linearized fluid flow model of TCP (3) can be rewritten as a time-varying delay system of the general form:

$$\dot{x}(t) = Ax(t) + A_d x(t-R(t)) + Bu(t-R(t)) + B_d d(t) \quad (5)$$

with

$$A = \begin{bmatrix} -\frac{N}{R_0^2C} & -\frac{1}{CR_0^2} \\ \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix}, A_d = \begin{bmatrix} -\frac{N}{R_0^2C} & \frac{1}{R_0^2C} \\ 0 & 0 \end{bmatrix},$$
$$B = \begin{bmatrix} -\frac{C^2R_0}{2N^2} \\ 0 \end{bmatrix}, B_d = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, x(t) = \begin{bmatrix} \delta W(t) \\ \delta q(t) \end{bmatrix}. \quad (6)$$

$x(t)$ is the state vector and represents the network variables status. A suitable control $u(t)$ (processed by the AQM) must be applied to system (5) in order to ensure a stable congestion control. Section 3 is devoted to the stability analysis of the interconnected system (system (5) + AQM). To this end, the Lyapunov Krasovskii method (see for example [24]) is used which is an extension of the traditional Lyapunov theory. It is an effective and practical method which provides LMI (Linear Matrix Inequalities, [25]) criteria easy to test.

# III. Stabilization: design of an AQM

In Section II, the model of TCP/AQM has been formulated in the general form of a time delay system. The stability of the congestion control requires the construction of a controller which regulates the buffer queue length as well as data flows. In this section, we are first going to present a delay dependent stability condition for time delay systems in general [21] (when $u(t)$ and $d(t)$ equal 0). Secondly, based on this criterion, a design method that provides a stabilizing state feedback is deduced. This control aims to ensure the convergence of $x(t)$ to 0 (thus the convergence of $W(t)$ and $q(t)$ to $W_0$ and $q_0$ respectively).

## A. Stability analysis of time delay systems

In this subsection, our goal is to derive a stability condition which takes into account an upperbound of the delay. The delay dependent case starts from a system asymptotically stable without delays and looks for the maximal delay that preserves stability. In this paper, we prove the stability property of (5) with the Lyapunov method which consists in looking for a positive function $V(x,t)$ such that its derivative $\dot{V}(x,t)$ along the trajectories of (5) is negative. This function $V(x,t)$ can be viewed as an energy function of the considered system which converges to zero proving then the stability of the system.

Usually, the method involves Lyapunov-Krasovskii functionals (see [24] and references therein), and more or less tight techniques to bound some cross terms. These choices of specific Lyapunov functionals and overbounding techniques are the origin of conservatism. In the present paper, we choose a recently developed Lyapunov-Krasovskii functional [21] which deals with the stability of time-varying delay systems and shows interesting results in terms of conservatism reduction. The key idea is to consider an extended state $z(t)$ (10) as it has been proposed in [26] in a robustness context. We make the following assumptions on the delay:

$$0 \leq h(t) \leq h_m \text{ and } |\dot{h}(t)| \leq r, \quad (7)$$

where $h_m$ and $r$ are upperbounds of the delay and the delay derivative respectively. Given a time delay system:

$$\dot{\varsigma}(t) = A\varsigma(t) + A_d\varsigma(t-h(t)) \quad (8)$$

where $\varsigma(t) \in \mathrm{R}^{\mathrm{n}}$ is the state vector, $A, A_d \in \mathrm{R}^{\mathrm{n}\times\mathrm{n}}$ are known constant matrices. Differentiating the system (8), we get:

$$\ddot{\varsigma}(t) = A\dot{\varsigma}(t) + (1-\dot{h}(t))A_d\dot{\varsigma}(t-h(t)).$$

Consider now the artificially augmented system

$$\begin{cases} \dot{\varsigma}(t) = A\varsigma(t) + A_d\varsigma(t-h(t)) \\ \ddot{\varsigma}(t) = A\dot{\varsigma}(t) + (1-\dot{h}(t))A_d\dot{\varsigma}(t-h(t)) \end{cases} \tag{9}$$

composed of the original system (8) and its derivative. Introducing the augmented state

$$z(t) = \begin{bmatrix} \varsigma(t) \\ \dot{\varsigma}(t) \end{bmatrix} \tag{10}$$

and specifying the relationship between the two components of $z(t)$ with the equality $[1 \;\; 0]\dot{\mathrm{z}}(\mathrm{t}) = [0 \;\; 1]\mathrm{z}(\mathrm{t})$, we get the new augmented system:

$$E\dot{z}(t) = \bar{A}z(t) + \bar{A}_d z(t-h(t)), \tag{11}$$

where

$$\begin{gathered} E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} A & 0 \\ 0 & A \\ 0 & 1 \end{bmatrix}, \\ \bar{A}_d = \begin{bmatrix} A_d & 0 \\ 0 & (1-\dot{h}(t))A_d \\ 0 & 0 \end{bmatrix}. \end{gathered} \tag{12}$$

Finally, we obtain a descriptor linear time delay and time varying system. To cope with the time-varying nature of system (11) a method consists in embedding the time varying parameters $h$ and $\dot{h}$ into an uncertain set, described by a polytopic set and employing quadratic stability framework (see [25] and [24]). The following Theorem is then proposed (see [21] for more details)

*Theorem 1:* Given scalars $h_m > 0$ and $r \geq 0$, the linear system (8) is asymptotically stable for any time-varying delay $h(t)$ satisfying (7) if there exists $2n \times 2n$ matrices $P > 0$, $Q_1 > 0$, $Q_2 > 0$, $R > 0$ and $X \in \mathrm{R}^{8n \times 3n}$ such that the following LMI holds for $i = \{1,2\}$:

$$\Gamma^{(i)} + XS^{(i)} + S^{(i)^T}X^T < 0 \tag{13}$$

where $\Gamma^{(i)}$ and $S^{(i)}$ (defined in (18) and (19)) for $i = 1,2$ are the two vertices of $\Gamma(\dot{h}) \in \mathrm{R}^{8n \times 8n}$ ($S(\dot{h}) \in \mathrm{R}^{3n \times 8n}$ respectively), replacing the term $\dot{h}(t)$ by $r_i$. $r_i$, $i = \{1,2\}$ corresponding to the bounds of $\dot{h}(t)$: $r_1 = r$ and $r_2 = -r$.

**Proof :** We consider the following Lyapunov-Krasovskii functional associated with the augmented state vector $z(t)$:

$$\begin{aligned} V(z_t) = \;& z_t^T(0)Pz_t(0) + \int_{-h(t)}^{0} z_t^T(\theta)Q_1 z_t(\theta)d\theta \\ & + \int_{-h_m}^{0} z_t^T(\theta)Q_2 z_t(\theta)d\theta \\ & + \int_{t-h_m}^{t}\int_{\theta}^{t} \dot{z}^T(s)R\dot{z}(s)dsd\theta. \end{aligned} \tag{14}$$

Remark that since $P$, $Q_1$, $Q_2$, $R$ are positive definite, we can conclude that for some $\varepsilon > 0$, the Lyapunov-Krasovskii functional condition $V(x_t) \geq \varepsilon\|x_t(0)\|$ is satisfied [24]. The derivative along the trajectories of (11) leads to

$$\begin{aligned} \dot{V}(z_t) \;=\;& 2z^T(t)P\dot{z}(t) + z^T(t)Q_1 z(t) \\ & -(1-\dot{h}(t))z^T(t-h(t))Q_1 z(t-h(t)) \\ & + z^T(t)Q_2 z(t) - z^T(t-h_m)Q_2 z(t-h_m) \\ & + h_m\dot{z}^T(t)R\dot{z}(t) - \int_{t-h_m}^{t} \dot{z}^T(\theta)R\dot{z}(\theta)d\theta. \end{aligned} \tag{15}$$

As noted in [27], the derivative of $\int_{t-h_m}^{t}\int_{\theta}^{t}\dot{z}^T(s)R\dot{z}(s)dsd\theta$ is often estimated as $h_m\dot{z}^T(t)R\dot{z}(t) - \int_{t-h(t)}^{t}\dot{z}^T(\theta)R\dot{z}(\theta)d\theta$ and the term $-\int_{t-h_m}^{t-h(t)}\dot{z}^T(\theta)R\dot{z}(\theta)d\theta$ is ignored, which may lead to considerable conservatism. Hence, the last term of (15) can be separated in two parts:

$$\begin{aligned} -\int_{t-h_m}^{t} \dot{z}^T(\theta)R\dot{z}(\theta)d\theta \;=\;& -\int_{t-h_m}^{t-h(t)} \dot{z}^T(\theta)R\dot{z}(\theta)d\theta \\ & -\int_{t-h(t)}^{t} \dot{z}^T(\theta)R\dot{z}(\theta)d\theta. \end{aligned} \tag{16}$$

Using the Jensen's inequality [24], (16) can be bounded as follow:

$$\begin{aligned} & -\int_{t-h_m}^{t-h(t)} \dot{z}^T(\theta)R\dot{z}(\theta)d\theta - \int_{t-h(t)}^{t} \dot{z}^T(\theta)R\dot{z}(\theta)d\theta \\ < \;& -v^T(t)\tfrac{R}{h_m-h(t)}v(t) - w^T(t)\tfrac{R}{h(t)}w(t) \\ < \;& -v^T(t)\tfrac{R}{h_m}v(t) - w^T(t)\tfrac{R}{h_m}w(t) \end{aligned}$$

with

$$\begin{aligned} v(t) &= z(t-h(t)) - z(t-h_m), \\ w(t) &= z(t) - z(t-h(t)). \end{aligned}$$

Therefore, we get $\dot{V}(z_t) < \psi^T(t)\Gamma(\dot{h})\psi(t)$ with

$$\psi(t) = \begin{bmatrix} \dot{z}(t) \\ z(t) \\ z(t-h(t)) \\ z(t-h_m) \end{bmatrix}. \tag{17}$$

$$\Gamma(\dot{h}) = \begin{bmatrix} h_m R & P & 0 & 0 \\ P & T & \frac{1}{h_m}R & 0 \\ 0 & \frac{1}{h_m}R & U & \frac{1}{h_m}R \\ 0 & 0 & \frac{1}{h_m}R & V \end{bmatrix} \tag{18}$$

and

$$\begin{aligned} T &= Q_1 + Q_2 - \tfrac{1}{h_m}R, \\ U &= -(1-\dot{h}(t))Q_1 - \tfrac{2}{h_m}R, \\ V &= -\tfrac{1}{h_m}R - Q_2. \end{aligned}$$

So, the system (11) is asymptotically stable if for all $\psi(t)$ such that $S(\dot{h})\psi(t) = 0$ with

$$S(\dot{h}) = \begin{bmatrix} -E & \bar{A} & \bar{A}_d & 0 \end{bmatrix}, \qquad (19)$$

the inequality $\psi(t)^T \Gamma(\dot{h})\psi(t) < 0$ holds. Using Finsler lemma [28], this is equivalent to

$$\Gamma(\dot{h}) + XS(\dot{h}) + S^T(\dot{h})X^T < 0. \qquad (20)$$

At this stage, assume that $\dot{h}(t)$ is not precisely known but varies between a lower and upper bound, $\dot{h}(t) \in [-r, r]$. Since this uncertain parameter appears linearly in (20), the uncertain set can be described by a polytope [24]. The vertices of this set can be calculated by setting the parameter to either lower or upper limit. The inequality (20) can then be rewritten as follow:

$$\sum_{i=1}^{2} \alpha_i \Gamma^{(i)} + X \sum_{i=1}^{2} \alpha_i S^{(i)} + \sum_{i=1}^{2} \alpha_i S^{(i)^T} X^T < 0, \qquad (21)$$

where $\alpha_i(t) \in [0,1]$, $\sum_{i=1}^{2} \alpha_i(t) = 1$ and $\Gamma^{(i)}$ $(S^{(i)})$, $i = 1,2$ are the two vertices of the uncertain matrix $\Gamma(\dot{h})$ $(S(\dot{h})$ respectively) for $\dot{h}(t) \in [-r, r]$. Considering the quadratic stability framework [25], condition (21) is equivalent to

$$\Gamma^{(i)} + XS^{(i)} + S^{(i)^T} X^T < 0, \ \mathrm{i} = 1,2. \qquad (22)$$

Thus, the inequality (20) has to be verified only on its vertices (22). Finally, the asymptotic stability of system (11) is guaranteed if the two LMI (22) are feasible at the same time. For any initial conditions, the whole state $z(t)$ converges asymptotically to zero. Its components $\varsigma(t)$ converge as well. The original system (8) is thus asymptotically stable.

*Remark 1:* Note that Theorem 1 provides a delay dependent stability condition. It means that if condition (13) holds for $h_{m_1}$ then it still holds for any $h_{m_2} \leq h_{m_1}$ [21].

### B. A first result on synthesis

In the previous section, a general stability condition for time-varying delay systems has been introduced. We aim now at using this latter result for the considered issue, established in Section II. Applying the delayed state feedback (4) (AQM mechanism) on the system (5) (TCP+router), the resulting feedback system can be reduced to a system of the form (8). Hence, given the stability condition (13) and the interconnected system (5) combined with (4) (in this subsection, the disturbance is not taken into account $d(t) = 0$), the following Theorem is obtained.

*Theorem 2:* Given scalars $h_m > 0$ and $r \geq$, if there exist symmetric positive definite matrices $P$, $R$, $Q_1$, $Q_2 \in \mathsf{R}^{2n\times 2n}$, a matrix $X \in \mathsf{R}^{8n\times 3n}$ and a matrix $K \in \mathsf{R}^{1\times n}$ such that

$$\Gamma^{(i)} + XS^{(i)} + S^{(i)^T} X^T < 0 \qquad (23)$$

then, the system (5) (with $d(t) = 0$) is stabilized by the control law $u(t) = Kx(t)$ for any time-varying delay R(t) satisfying $0 \leq R(t) \leq h_m$ and $\dot{R}(t) \leq r$. $\Gamma^{(i)}$ and $S^{(i)}$ (defined in (18) and (25)) for $i = 1,2$ are the two vertices of $\Gamma(\dot{R}) \in \mathsf{R}^{8n\times 8n}$ $(S(\dot{R}) \in \mathsf{R}^{3n\times 8n}$ respectively), replacing the term $\dot{R}(t)$ by $r_i$. $r_i$, $i = \{1,2\}$ corresponding to the bounds of $\dot{R}(t)$: $r_1 = r$ and $r_2 = -r$.

**Proof :** Consider system (5) with $d(t) = 0$ and controlled by the state feedback (4), can be expressed as

$$\dot{x}(t) = Ax(t) + \breve{A}_d x(t - R(t)), \qquad (24)$$

where $\breve{A}_d = A_d + BK$ and $A$, $A_d$ and $B$ are defined as (6). Then, Theorem 1 can be applied on the interconnected system (24). Following the same idea exposed in Section III-A Theorem 2 is derived considering now $S(\dot{R}(h))\xi(t) = 0$ where

$$S(\dot{R}(h)) = \begin{bmatrix} -E & \hat{A} & \hat{A}_d & 0 \end{bmatrix}, \qquad (25)$$

with

$$E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \hat{A} = \begin{bmatrix} A & 0 \\ 0 & A \\ 0 & 1 \end{bmatrix},$$

$$\hat{A}_d = \begin{bmatrix} A_d + BK & 0 \\ 0 & (1 - \dot{R}(t))(A_d + BK) \\ 0 & 0 \end{bmatrix},$$

$$\xi(t) = \begin{bmatrix} \dot{z}(t) \\ z(t) \\ z(t - R(t)) \\ z(t - h_m) \end{bmatrix}.$$

Thus, the stability condition (23) of Theorem 2 enables the design of gains $k_1$ and $k_2$. This condition is formulated as a matrix inequality which can be systematically solved with an appropriate semi-definite programming solver in Matlab [29] and Yalmip [30].

*Remark 2:* Regarding the design problem, since $K$ is a decision variable condition (23) is bilinear with $X$ and a global optimal solution cannot be found. Nevertheless, the feasibility problem can still be tested to provide a solution by either using a BMI solver [31] or developping a relaxation algorithm based on LMI [32].

### C. State feedback with an integral action

The first proposed method, in Section III-B, for the design of an AQM ensures the stability of communications around an equilibrium point (2). However, this equilibrium may be perturbed when non-responsive crossing traffics are introduced. Indeed, these additional non-TCP (so non-modeled) flows fill up the buffer and the AQM may not control the congestion as expected. In order to cope with

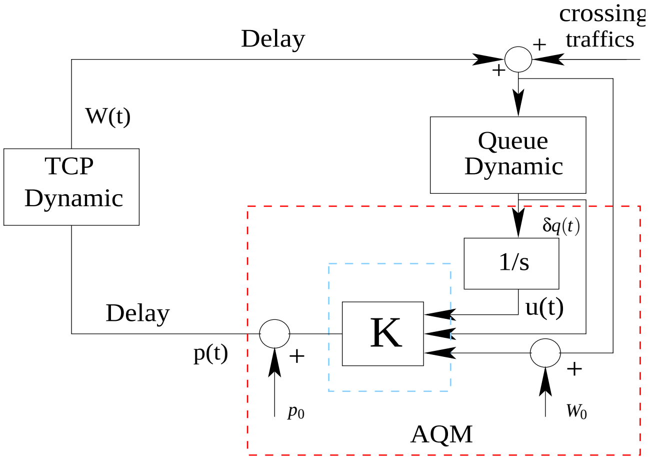


**Fig. 3. Design of an AQM as a dynamic state feedback**

this phenomenon, the AQM is completed with an integral action well known to be able to reject disturbances. The idea is to apply the same design methodology, exposed in Section III-B, over an augmented time delay system composed of the original system (5) and an integrator (see Figure 3). The augmented system has the following form

$$\dot{\eta} = \begin{bmatrix} \begin{matrix} & A & \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 1 \end{matrix} & 0 \end{bmatrix} \eta(t) + \begin{bmatrix} \begin{matrix} & A_d & \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 \end{matrix} & 0 \end{bmatrix} \eta(t-h) + \begin{bmatrix} B \\ 0 \end{bmatrix} \delta p(t-h) + \begin{bmatrix} B_d \\ 0 \end{bmatrix} d(t) \tag{26}$$

with $\eta^T = [\delta W \quad \delta q \quad u]^T$ is the extended state variable. Then, the global control which correspond to our AQM, is a dynamic state feedback

$$\delta p(t) = K \begin{bmatrix} \delta W(t) \\ \delta q(t) \\ u(t) \end{bmatrix} = k_1 \delta W(t) + k_2 \delta q(t) + k_3 \int_0^t \delta q(t) dt. \tag{27}$$

In our problem, non modeled crossing traffics $d(t)$ such as UDP based applications are introduced as exogenous signals (see Figures 2 and 3). The queue dynamic, the second equation of (3), is affected by this additional signals. Considering equations (27) and (3), we obtain the transfer function $T(s)$ from the disturbance $D(s)$ to the queue size (about the operating point) $\Delta Q(s)$:

$$T(s) = \frac{b(s)s}{(s + \frac{1}{R_0})sb(s) + c(s)}, \tag{28}$$

with

$$a(s) = -\frac{R_0 C^2}{2N^2} e^{-hs},$$
$$b(s) = s + \frac{N}{R_0^2 C}(1 + e^{-hs}) + a(s)k_1$$
$$c(s) = \frac{N}{R_0} \left[ \frac{s}{R_0^2 C}(1 - e^{-hs}) + a(s)sk_2 + a(s)k_3 \right].$$

It can be easily shown that for a step type disturbance, the queue size still converges to its equilibrium. The proposed control law (27) is thus suitable to cope with the effects of non responsive traffics. Then, $k_1$, $k_2$ and $k_3$ are designed applying Theorem 2 to the augmented system (26).

## D. Estimation of the congestion window

In these last two parts, a state feedback synthesis has been performed for the congestion control of TCP flows and the management of the router buffer. So far we have considered that the whole state was accessible. However, although the congestion window can be measured in NS, it is not the case in reality. That's why, in this paper it is proposed to estimate this latter variable using the aggregate flow incoming to the router buffer. The sending rate of single TCP source can be approximated by congestion window size over the RTT. This latter approximation is valid as long as the model does not describe the communication at a finer time scale than few round trip time (see [12]). Consequently, the whole incoming rate observed by the router is $r(t) = NW(t)/R(t)$. The measure of the aggregate flow has already been proposed and successfully exploited in [8], [19] for the realization of the AVQ and a PID type AQM respectively. Moreover, other works have also developed tools that enable such measurements in anomaly detection framework (see for example [33], [34]). It is worth to note that queue-based AQMs like RED or PI can be assimilated as output feedbacks according to the queue length [35]. Conversely, AVQ can be viewed as an output feedback with respect to the aggregate flow, belonging thus to the rate-based AQM class. The proposed state feedbacks in Section III-B and III-C combine both informations and we expect thus to apply a relevant control on the communications.

# IV. NS-2 simulations

This section is devoted to elucidate the proposed methodology via an illustrative example and simulations with NS-2 [22]. Consider a network consisting of $N = 60$ communicating pairs through a congested router as illustrated on Figure 1. The transport protocol is TCP-New Reno without ECN marking and the propagation time is $T_p = 200$ ms. Since the link capacity $C = 3750$ packets/s (corresponds to a 15 Mb/s link with average packet size 500 bytes) at the router is shared among all users, there is congestion and the classical *drop tail* mechanism drops packets when buffer overflows (the maximal buffer size is set to 800 pkts). Hence, the queue length at the router shows large oscillations and reaches oftenly the buffer saturation (see Figure 4). Now, we aim at regulating the queue length at the router to a desired level: $q_0 = 175$ packets. To this end, an AQM is embedded into the router in order to control the congestion phenomenon. Different AQMs have been simulated and their configuration parameters are shown in table I. Regarding to the design of the proposed AQM, given the network parameters ($N$, $C$ and $T_p$) and the specification on $q_0$, the equilibrium point (2) can be derived: $W_0 = 15$ packets, $p_0 = 0.008$ and $R_0 = 0.246$ seconds. Secondly, according to the design criteria presented in Section III, the state feedback matrices

$$K_{SF} = 10^{-3} \begin{bmatrix} 0.6103 \\ 0.0209 \end{bmatrix}$$
$$K_{SFI} = 10^{-4} \begin{bmatrix} 26.28 \\ 0.303 \\ 0.464 \end{bmatrix} \tag{29}$$

are calculated based on control laws (4) and (27) respectively.

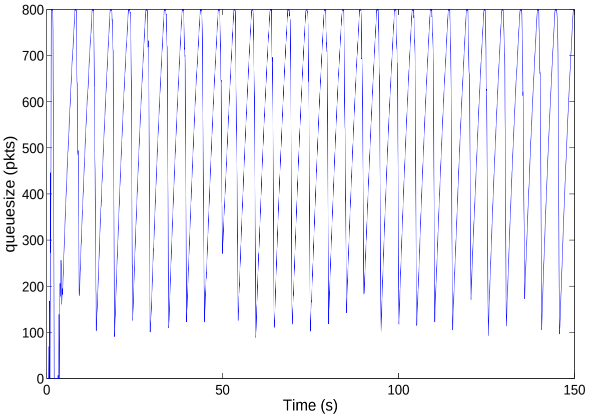


**Fig. 4. Time evolution of the queue length, dropping strategy is the traditional mechanism:** *DropTail*

On Figure 5, the time evolutions of the queue length for different AQMs are shown. In this first simulation, RED, REM, PI and our state feedback (SF: control law (4)) have been tested for congestion control under long-lived TCP flow such that ftp connections. It can be observed that our SF is able to regulate efficiently the buffer:

| | |
|---|---|
| RED | $min_{th}$=150,$max_{th}$=700,<br>$w_Q$=13.3e-06,$max_p$=0.1,$f_s$=160Hz |
| REM | $\gamma$=0.001, $\Phi$=1.001,$q_{ref}$=175pkt |
| PI | a=1.822e-05,b=1.816e-05,<br>$q_{ref}$=175pkt,$f_s$=160Hz |
| SF | gains $K$ (29), equilibrium point (2) |

**TABLE I. Adjustment of parameter setting of each AQM**

- the queue length reaches the steady state fastly,
- it maintains the size close to its equilibrium value $q_0$, ensuring then very low oscillations,
- this control which guarantees a stable queue length, allows to keep a queueing delay with very little variations, thus low delay jitter (see Figure 6).

Then, if non-responsive cross traffics are introduced, the queue is affected and may disturb the AQM control behavior. Considering the previous controller SF ($K_{SF}$ in (29)), we have carried out a new simulation (Figure 7) introducing additional traffics composed of 7 sources (CBR applications over UDP protocol) sending flows of 1Mbytes/s between $t = 50s$ and $t = 100s$. It appears that the queue length is still stable but not regulated at the desired level $q_0$ anymore. That's why, in Section III the first control law (4) has been completed with an integral action to tackle the steady state error in presence of non-responsive CBR traffics. Then, the same simulation is performed using the AQM $K_{SFI}$ (29) from the control law (27). Figure 8 shows the different results for each AQM. In addition, table II summarizes the benefits of the $K_{SFI}$ AQMs providing few statistical characteristics. These characteristics are mean, standard deviation ($Std$) and the square of the variation coefficient ($CV2 = (Std/mean)^2$). This latter calculation assess the relative dispersion of the queue length around its mean. The mean points out the control precision and the standard deviation shows the ability of the AQM to keep the queue size close to its equilibrium. In table II, we can observe that $K_{SFI}$ maintains a very good control on the buffer queue during the whole simulation. Hence, it enable to ensure QoS in terms of RTT (set to a desired value) and delay jitter (see Figure 9). Although PI reject the perturbation quite fast, extensive fluctuations appear during the steady state. Note that RED allows also a good regulation at the steady state but its response time is quite slow. Moreover, this latter is well known to be difficult to tune [36], [37] whereas the proposed AQM can be easily and systematically derived solving the inequality of Theorem 2 with an appropriate semi-definite programming solver (as penbmi [31], sedumi [38] or lmilab in Matlab [29]).

| AQM | RED | REM | PI | $K_{SFI}$ | Period |
|---|---|---|---|---|---|
| Mean | 235.7 | 177.4 | 178.8 | 176.7 | before additional traffic |
| Std | 112.4 | 144.74 | 89.83 | 71.19 | |
| CV2 | 0.227 | 0.665 | 0.252 | 0.162 | |
| AQM | RED | REM | PI | $K_{SFI}$ | Period |
| Mean | 270.3 | 212.4 | 199.4 | 178.3 | during additional traffic |
| Std | 57.39 | 101.5 | 79.05 | 40.42 | |
| CV2 | 0.045 | 0.228 | 0.1972 | 0.051 | |
| AQM | RED | REM | PI | $K_{SFI}$ | Period |
| Mean | 201.4 | 168.4 | 154.1 | 177.8 | after additional traffic |
| Std | 22.24 | 101.02 | 64.95 | 36.64 | |
| CV2 | 0.012 | 0.360 | 0.178 | 0.042 | |

**TABLE II. Statistical characteristics for different AQMs (units are pkts) at different periods (before, during and after the introduction of CBR traffic)**

## V. Conclusion and Future Work

In this work, we have proposed the design of an AQM for the congestion control in communications networks. The developed AQM has been constructed using a state feedback control law. An integral action has been added to reject the steady state error in spite of disturbance, $d(t)$ (non-responsive crossing traffic). Finally, the AQM has been validated using NS simulator. Future works consist in the improvement of control laws and extension to larger networks (with TCP sources at varying hop-distances and using short-term/long-term flows). Validation on emulation platform (experimental part) will be also studied.

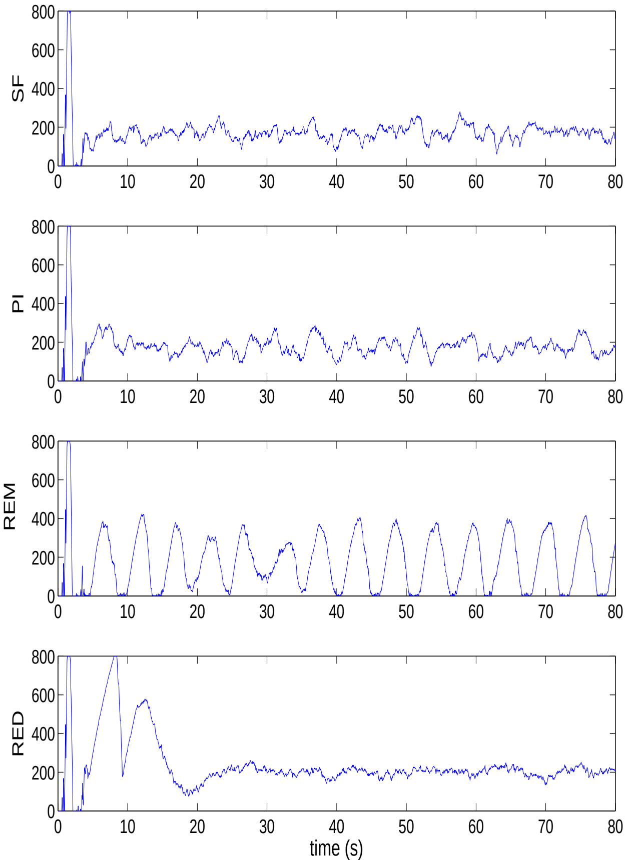


**Fig. 5. Time evolution of the queue length,** $AQM = \{K_{SF}, PI, REM, RED\}$

## References

[1] Y. Ariba, Y. Labit, and F. Gouaisbaut, “Design and performance evaluation of a state-space based aqm,” in *IARIA International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ 2008)*, Jul. 2008, pp. 89–94.

[2] V. Jacobson, “Congestion avoidance and control,” in *ACM SIGCOMM*, Stanford, CA, Aug. 1988, pp. 314–329.

[3] B. Braden, D. Clark, and J. Crowcroft, “Recommendations on queue management and congestion avoidance in the internet,” RFC 2309, Apr. 1998.

[4] L. Le, J. Aikat, K. Jeffay, and F. Donelson Smith, “The effects of active queue management on web performance,” in *ACM SIGCOMM*, Aug. 2003, pp. 265–276.

[5] K. K. Ramakrishnan and S. Floyd, “A proposal to add explicit congestion notification (ecn) to ip,” RFC 2481, Jan. 1999.

[6] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, Aug. 1993.

[7] S. Athuraliya, D. Lapsley, and S. Low, “An enhanced random early marking algorithm for internet flow control,” in *IEEE INFOCOM*, Dec. 2000, pp. 1425–1434.

[8] S. Kunniyur and R. Srikant, “Analysis and design of an adaptive virtual queue (avq) algorithm for active queue management,” in *ACM SIGCOMM*, San Diego, CA, USA, aug 2001, pp. 123–134.

[9] S. Ryu, C. Rump, and C. Qiao, “Advances in active queue management (aqm) based tcp congestion control,” *Telecommunication Systems*, vol. 4, pp. 317–351, 2004.

[10] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, “Analysis and design of controllers for aqm routers supporting tcp flows,” *IEEE Trans. on Automat. Control*, vol. 47, pp. 945–959, Jun. 2002.

[11] V. Misra, W. Gong, and D. Towsley, “Fluid-based analysis of a network of aqm routers supporting tcp flows with an application to red,” in *ACM SIGCOMM*, Aug. 2000, pp. 151–160.

[12] H. S. Low, F. Paganini, and J. Doyle, *Internet Congestion Control*. IEEE Control Systems Magazine, Feb 2002, vol. 22, pp. 28–43.

[13] R. Srikant, *The Mathematics of Internet Congestion Control*. Birkhauser, 2004.

[14] S. Tarbouriech, C. T. Abdallah, and J. Chiasson, *Advances in communication Control Networks*. Springer, 2005.

[15] A. Papachristodoulou, “Global stability of a tcp/aqm protocol for arbitrary networks with delay,” in *IEEE CDC 2004*, Dec. 2004, pp. 1029–1034.

[16] C. Chen, Y. Hung, T. Liao, and J. Yan, “Design of robust active queue management controllers for a class of tcp communication networks,” *Information Sciences.*, vol. 177, no. 19, pp. 4059–4071, 2007.

[17] S. Manfredi, M. di Bernardo, and F. Garofalo, “Robust output

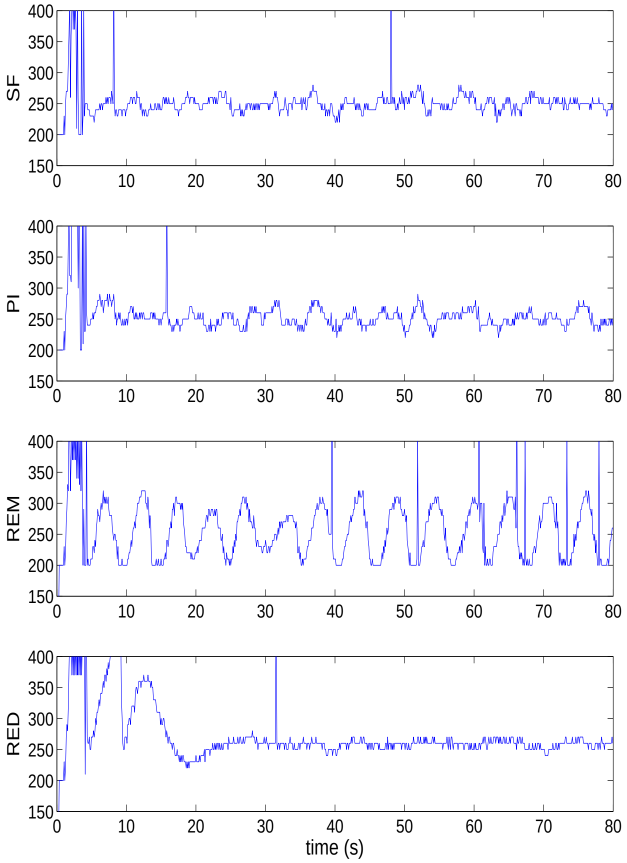


**Fig. 6. Time evolution of the RTT, $AQM = \{K_{SF}, PI, REM, RED\}$**

feedback active queue management control in tcp networks," in *IEEE Conference on Decision and Control*, Dec. 2004, pp. 1004–1009.

[18] D. Wang and C. V. Hollot, "Robust analysis and design of controllers for a single tcp flow," in *IEEE International Conference on Communication Technology (ICCT)*, vol. 1, Apr. 2003, pp. 276–280.

[19] K. B. Kim, "Design of feedback controls supporting tcp based on the state space approach," in *IEEE Trans. on Automat. Control*, vol. 51 (7), Jul. 2006.

[20] W. Michiels, D. Melchior-Aguilar, and S. Niculescu, "Stability analysis of some classes of tcp/aqm networks," in *International Journal of Control*, vol. 79 (9), Sep. 2006, pp. 1136–1144.

[21] Y. Ariba and F. Gouaisbaut, "Delay-dependent stability analysis of linear systems with time-varying delay," in *IEEE Conference on Decision and Control*, Dec. 2007, pp. 2053–2058.

[22] K. Fall and K. Varadhan, "The ns manual," notes and documentation on the software ns2-simulator, 2002, uRL: www.isi.edu/nsnam/ns/.

[23] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, "On designing improved controllers for aqm routers supporting tcp flows," in *IEEE INFOCOM*, vol. 3, Apr. 2001, pp. 1726–1734.

[24] K. Gu, V. L. Kharitonov, and J. Chen, *Stability of Time-Delay Systems*. Birkhäuser Boston, 2003, control engineering.

[25] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, USA: SIAM, 1994, in Studies in Applied Mathematics, vol.15.

[26] Y. Ebihara, D. Peaucelle, D. Arzelier, and T. Hagiwara, "Robust performance analysis of linear time-invariant uncertain systems by taking higher-order time-derivatives of the states," in $44^{th}$ *IEEE Conference on Decision and Control and the European Control Conference*, Seville, Spain, Dec. 2005.

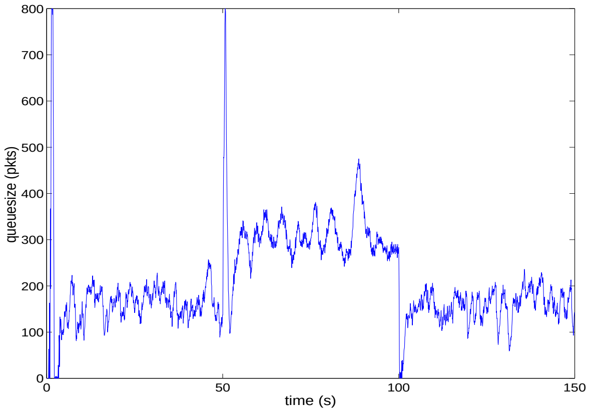


**Fig. 7. Time evolution of the queue length, $AQM = K_{SF}$ under UDP crossing traffic**

[27] Y. He, Q. G. Wang, C. Lin, and M. Wu, "Delay-range-dependent stability for systems with time-varying delay," *Automatica*, vol. 43, pp. 371–376, 2007.

[28] R. Skelton, T. Iwazaki, and K. Grigoriadis, *A unified algebric approach to linear control design*. Taylor and Francis series in systems and control, 1998.

[29] Mathworks, "Matlab - the language of technical computing," http://www.mathworks.fr/products/matlab/.

[30] J. Lfberg, "Yalmip : A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. [Online]. Available: http://control.ee.ethz.ch/ joloef/yalmip.php

[31] M. Kocvara and M. Stingl, "bilinear matrix inequalities: Penbmi," PENOPT GbR, http://www.penopt.com/.

[32] Y. Labit, Y. Ariba, and F. Gouaisbaut, "Design of lyapunov based controllers as tcp aqm," in *2nd IEEE Workshop on Feedback control implementation and design in computing systems and networks (FeBID'07)*, Munich, Germany, May 2007, pp. 45–50.

[33] P. Barford and D. Plonka, "Characteristics of network traffic flow anomalies," Nov. 2001.

[34] S. S. Kim and A. L. N. Reddy, "Netviewer: a network traffic visualization and analysis tool," in *LISA'05: Proceedings of the 19th conference on Large Installation System Administration Conference*. USENIX Association, 2005, pp. 185–196.

[35] D. Supratim and R. Srikant, "Rate-based versus queue-based models of congestion control," *IEEE Trans. on Automat. Control*, vol. 51, pp. 606–619, Apr. 2006.

[36] M. Christiansen, K. Jeffay, D. Ott, and F. Smith, "Tuning red for web traffic," in *ACM/SIGCOM*, 2000, pp. 139–150.

[37] T. Bonald, M. May, and J.-C. Bolot, "Analytic evaluation of red performance," in *IEEE INFOCOM*, vol. 3, Mar. 2000, pp. 1415–1424.

[38] J. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," Optimization Methods and Software 625-653, Special issue on Interior Point Methods, 1999, http://sedumi.mcmaster.ca/.

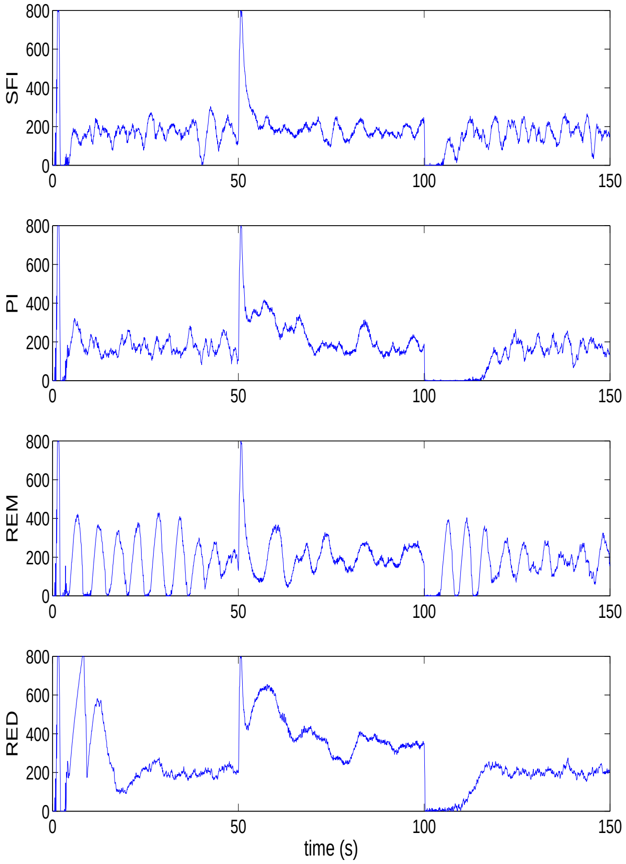


**Fig. 8. Time evolution of the queue length, $AQM = \{K_{SFI}, PI, REM, RED\}$ under UDP crossing traffic**

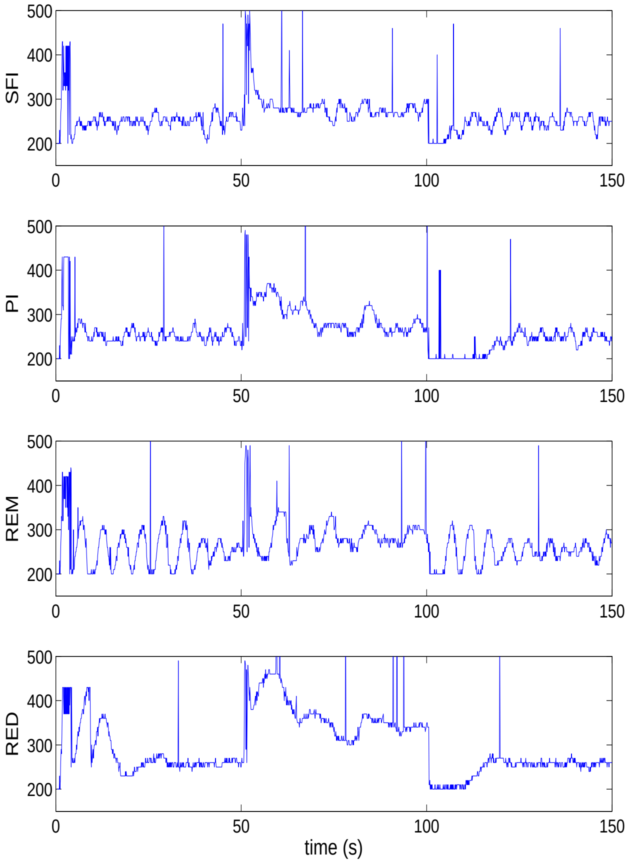


**Fig. 9. Time evolution of the RTT, $AQM = \{K_{SFI}, PI, REM, RED\}$ under UDP crossing traffic**

# A Disaster Aid Sensor Network using ZigBee for Patient Localization and Air Temperature Monitoring

Ashok-Kumar Chandra-Sekaran1, Anthony Nwokafor2, Layal Shammas3, Christophe Kunze3, Klaus D. Mueller-Glaser1

*1 Institute for Information Processing Technology, University of Karlsruhe (TH),Germany. 2 California Institute for Telecommunication and Information technology, University of California San Diego. 3 FZI Research Center for Information Technology, Karslruhe,Germany*

*{chandra, kmg}@itiv.uka.de, {aanwokaf, pjohansson, ikrueger}@ucsd.edu*

*{shammas,Kunze}@fzi.de*

## Abstract

*The mass casualty emergency response involves logistic impediments like overflowing victims, paper triaging, extended victim wait time and transport. We propose a new system based on a location aware wireless sensor network to overcome these impediments and assist the emergency responders (ER) to improve emergency response during disasters. In this paper we focus on the communication aspect, localization aspect and disaster site environment (air temperature) monitoring functionalities of this new emergency response system. We have done ZigBee simulations for investigating the handling of routers, mobility and scalabilty by ZigBee and thereby find out its suitability for our scenario. We have developed an energy-efficient ZigBee-ready temperature sensor node hardware and setup a ZigBee mesh network demonstrator. A RSSI-based localization solution is analyzed to find its suitability for tracking patients at the disaster site. A new algorithm to detect and display the temperature zones at the disaster site is developed and analyzed to find its computation efficiency. The patient tracking and temperature zone detection results show the increase of situation awareness, which can enable fast patient evacuation.*

*Keyword: Emergency response, ZigBee mesh network simulation, Localization, Temperature zone detection.*

## 1. Introduction

During a mass casualty disaster, one of the most urgent problems is to evacuate the patients from the disaster site as quickly as possible [21]. When chemical explosions take place it's difficult to shift the patients to zones free from toxic gases. The emergency response system that currently exists involves manual interpretation which is labor intensive, time consuming and error prone.

A new emergency response system (see Section 3) based on a wireless sensor network (WSN) is proposed by us to solve these problems. ZigBee simulations are undergone to find its suitability for disaster management scenario (see Section 4). A ZigBee mesh network is constructed and we have measured the current consumption results of the ZigBee-ready temperature sensor node (Section 5). We have analyzed a RSSI based localization solution to find out its suitability in patient tracking (Section 6). Finally, we show that our algorithm for detecting the temperature zones at the disaster site is effective in alerting the responders about danger zones (Section 7).

## 2. Disaster Management Scenario

The new emergency response system we propose is based on the disaster management strategy followed for disasters like chemical explosion, fire in building etc. The on-site organization chief (OOC) designates the disaster site into several zones [2] (see Fig. 1).

The hot zone, also referred to as the exclusion zone, is the area where contamination may occur. The warm zone is the area where the Contamination Reduction Corridor (CRC) is located. The cold zone area is chosen for forming the triage zone (TZ), the treatment zone (TTZ) and the transport zone (TRZ). Triaging is a method to classify the patients according to the severity of their injury and prioritize them for evacuation. There are four different classes of triaging:- Red: patients who require immediate attention, Yellow: patients who require delayed attention, Green: patients with light injuries, Blue: patients with no hopes of survival.

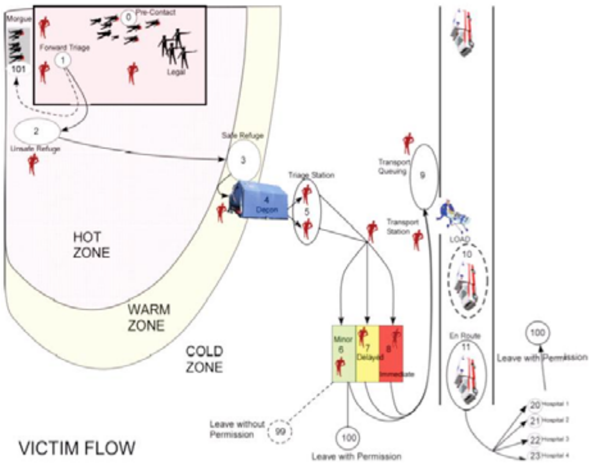


**Fig. 1. Disaster Site Zones**

### 2.1 Field study

A disaster simulation drill was conducted by state fire department Bruchsal, Germany. The on-site organization chief (OOC) drew a map and accounted the details of the number of medical responders, transport vehicles, zones [3]. The manual mapping was time consuming, complex for updating real time changes and the resource estimation was hindered.

With a resource limited response team, patients often wait for an extended period of time before transport. There is no continuous patient vital sign monitoring currently used [5]. The paper based triage is a bottle neck and makes the re-triaging difficult [6]. In addition, patients with minor injuries often depart the scene without notifying the response team, thus creating an organizational headache for OOC.

At the San Diego disaster drill [4] conducted by UCSD/Calit2 (University of California San Diego), the simulation of a car bomb detonation that destroyed the taxi and sent plumes of charcoal-grey smoke containing lethal chemicals was undergone. The emergency officials found it complex to identify the high temperature zones that could harm the patients. The plumes or the colorless gases were heading in the direction of the victim holding area in the cold zones and caused respiratory hazards.

### 2.2 Related Works for Emergency Response Sensor Networks

The Advanced Health and Disaster Aid Network (AID-N) from Johns Hopkins University, Applied Physics Laboratory develops technology-based solutions for time-critical patient monitoring, ambulance tracking, web portals for patient information flow etc. AID-N mainly focuses on critical patient monitoring [4] at the disaster site. But in our emergency response system we have mainly focused on solving logistic problems which are critical at the disaster site.

## 3. Disaster Aid Network (DAN)

An emergency response system is proposed based on the DAN [1] to solve the problems mentioned in Section 2.1.

The DAN architecture (see Figure 2) consists of hundreds of nodes distributed in a disaster site and wirelessly interconnected to form a mesh network. Several standard wireless technologies (ZigBee, WLAN, etc) are investigated and ZigBee is chosen as the wireless technology for DAN, since it's a low power and a standard-based technology for interconnecting large number of nodes [7] [8].

The DAN ZigBee network uses the 2.4 GHz band which operates worldwide, with a maximum data rate of 250kbps [7]. ZigBee network can access up to 16 separate 5 MHz channels in the 2.4 GHz band, several of which do not overlap with US and European versions of IEEE 802.11 or Wi-Fi. It incorporates an IEEE 802.15.4 defined CSMA-CA protocol that reduces the probability of interfering with other users and automatic retransmission of data ensures robustness. Its self-forming feature enables the mesh network to be formed by itself thereby enabling the network to be easily scalable. Its self-healing mesh network architecture permits data to be passed from one node to another node via multiple paths. Its security toolbox ensures reliable and secure networks. The MAC layer uses the Advanced Encryption Standard (AES) as its core cryptographic algorithm and describes a variety of security suites that use the AES algorithm. These suites can protect the confidentiality, integrity and authenticity of MAC frames.

There are three logical device types in ZigBee namely- the coordinators, routers, and end devices. A coordinator initializes a network, manages network nodes, and stores network node information. A Router node is always active and participates in the network by routing messages between paired nodes. The routing is based on the simplified Ad-hoc on demand Distance Vector (AODV) method. An end device is the low power consuming node as it is normally in sleep mode most of the time. It can take 15 ms (typical) to wake up from sleep mode [8].

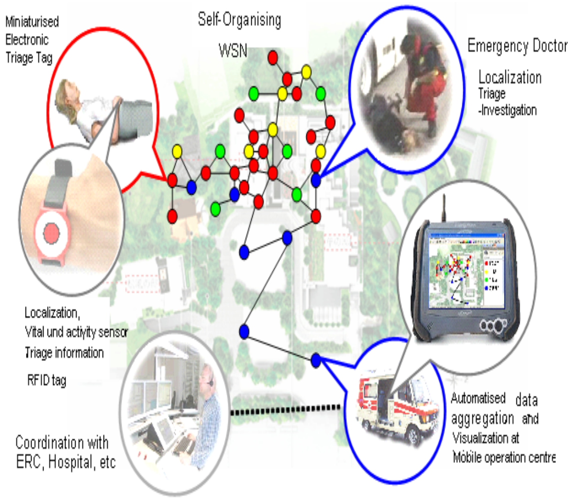


Fig. 2. DAN Architecture

DAN is a heterogeneous network [9] [10] formed with the following type of nodes:

**Patient node**: Minimized electronic triage tag, localization support, ZigBee mote, vital and activity sensors, RFID tag.

**Emergency doctor node**: PDA with GPS and ZigBee mote for patient monitoring and data recording

**Static anchor (router) nodes**: ZigBee motes with known location coordinates, environmental sensors (ex: air temperature) that can be deployed at the site.

**Monitor station** (coordinator): It is a collector node used by the EMC / OOC which supports data aggregation; visualization of inter-zone patient flow, transport capacity indication, patient location, triage information and patient vital signs [11].

At the beginning of the emergency response the OOC classifies the zones of the disaster site. The monitor station which is a portable device (notebook) gets online; the static anchor nodes (reference nodes for patient localization) are deployed manually covering the disaster area; the emergency doctor nodes [6] act as mobile anchor nodes; once a patient is found the doctor provides a wearable patient node. Each patient node (blind node) updates the monitor station with real time patient data.

## 4. ZigBee Simulation

In order to find out the suitability of ZigBee for DAN-Architecture ZigBee mesh network simulations are done. The analysis of statistics like application traffic sent vs. application traffic received, end to end delay, and packet loss ratio is done for every simulation setup to investigate the following features:

- Scalability of the network
- Routing capability of nodes and effect on the number of routers in the network
- Impact of mobility of the nodes and the node heterogeneity

### 4.1 Related Works for ZigBee Simulation

In [17], Gianluigi Ferrari et al. analyze the influence of relaying nodes in ZigBee networks. He shows that the use of relaying nodes degrades ZigBee networks and that the use of acknowledgment messages highly increases the network performance. But the impact of relaying nodes only on a static ZigBee network is analysed and the effect of relaying nodes on static and mobile ZigBee networks is unknown. In [18] Nia-Chiang Liang et. al. investigate the influence of node heterogeneity on ZigBee networks. The behaviour of a ZigBee network with different percentage of end devices is analysed but the analysis is limited to a fixed number of nodes with a fixed range and a fixed transmission rate of 10 packets/sec. Furthermore the mobile nodes in this simulation move according to the random waypoint mobility model which is not realistic for our Disaster Management Scenario. In [19] Ling-Jyh et. al. discuss about the influence of mobility in ZigBee networks. He shows that both the number of mobile nodes and the speed of the nodes severely affect the ZigBee networks. He also tells that the use of an end device as mobile receiver will degrade the network performance. But for investigating the effect of mobility, he uses random waypoint which is not realistic for our scenarios. Therefore we have developed our own self-defined trajectory based mobility models to investigate the effect of mobility if ZigBee is used for the DAN.

### 4.2 Simulator

Several SOA simulators like NS2 [28], OPNET [20] are surveyed and OPNET is chosen for our simulation because OPNET allows simulation of complex networks and includes a ZigBee model library that implements the important features of the ZigBee standard [20]. OPNET supports the mobility models: random waypoint as well as self defined trajectories. The OPNET statistics for analysis are classified into two main groups: local statistics and global statistics. Local statistics describe the behavior of a particular node while the global statistics describe the behavior of the global network.

### 4.3 Simulation Setup and Analysis

In this section, we perform ZigBee simulations using OPNET simulator in static and mobile scenarios (since DAN has both static and mobile nodes) and analyze the results to find out ZigBee's suitability for DAN. Before running simulations ZigBee's stack parameters are configured according to the needs of our application.

#### 4.3.1 ZigBee Stack Parameter

Application layer: The packet size is set to 100 Bytes and the transmission rate is set to 1packet/20sec. In DAN, packet can be delivered either at a particular time interval or when an event occurs (ex: patient movement).

MAC layer: Packet Acknowledgment is enabled. Acknowledgment wait duration is set to 864µsec. The number of retries is set to three.

Network layer: The 'network maximum depth' is set to 5, 'network maximum children' is set to 20, 'network maximum router' is always less than or equal to the count of 'network maximum children' parameter.

#### 4.3.2 Static Scenario

Two static scenarios are developed: offline node scaling scenario and routers influence scenario.

#### 4.3.2.1 Offline Node Scaling Scenario

In this scenario the network area is set to 300m x 300m with the nodes randomly placed over the area. The transmission range of the nodes is fixed to 100 meters. All nodes transmit packets to a single collector node (coordinator) which only receives and doesn't send any traffic. Several simulations are run by varying the total number of nodes (20, 50, 70, 100, 150 and 200). 50% of the total number of nodes acts as router.

Figure 3 compares the application traffic sent (by all the nodes in the network) to the application traffic received (by the single collector node), when the number of nodes is varied from 20 to 200. It is seen that the network scales well up to 150 nodes. From 150 nodes to 200 nodes, the traffic loss is around 400bit/sec. The traffic loss is high due to the increase of network density which inturn increases the channel access attempts and thereby the retransmissions. After the three retransmission attempts, the packet is dropped.

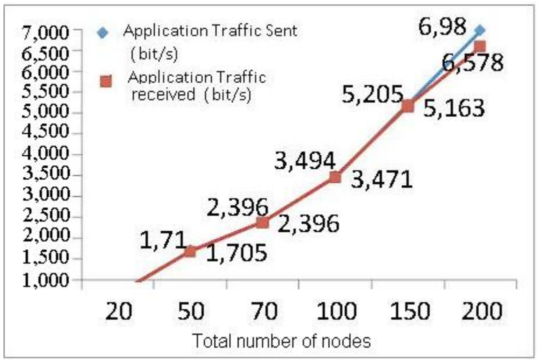


**Fig. 3: Application traffic sent vs. Application traffic received**

Figure 4 shows the packet loss ratio (in percentage) with increase of the number of nodes. The PLR increases linearly between 20 and 150 nodes after which it increases drastically. With more traffic the probability that a packet get lost is higher as the risk of collision through hidden nodes becomes higher.

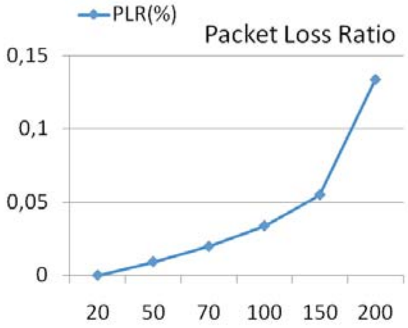


**Fig. 4: Packet Loss Ratio for offline node scaling scenario**

Figure 5 depicts the end-to-end delay (ETE) which is defined as the time elapsed, since a packet is sent by a sending node to a receiver node, and the reception of the Acknowledgment by the sending node. Similar to PLR the ETE also increases with the number of nodes. It increases slowly from 20 to 100 nodes, and then it increases exponentially. When the number of nodes increases, the number of attempt of the application traffic sent also increase. According to the channel access mechanism (CSMA-CA), the nodes listen to the medium before they send. If the medium is busy, they wait for a randomly chosen time and then try again. With more nodes attempting to access the channel, the probability that the channel is busy becomes higher. Also with larger number of nodes, the channel is busier and consequently, the number of retries is increased.

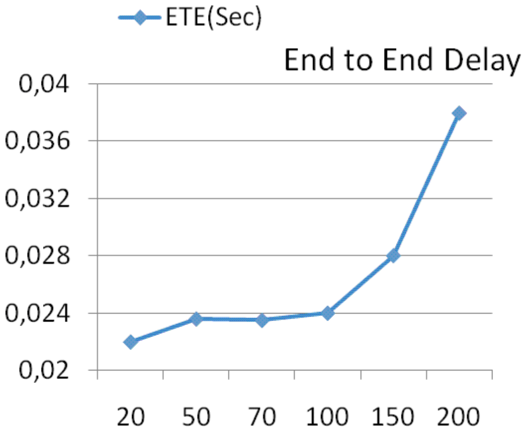


**Fig.5: End to End Delay for offline node scaling scenario**

### 4.3.2.2 Router Influence Scenario

In this scenario the impact of routers on a static network is analyzed. The network area is set to 300m x 300m, the node transmission range is set to 100m, fixed total number of nodes is set to 100. The percentage of routers in the network is varied from 20 to 100 % and the effect of routers is analysed.

In Fig. 6, both the application traffic sent (red line) and the application traffic received are almost the same as the number of routers are increased from 20 to 100%. This shows that the number of routers has no impact on the application traffic received.

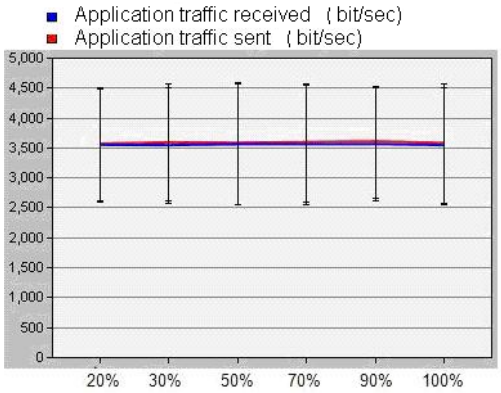


**Fig. 6: Application traffic sent vs. Application traffic received - Router Influence scenario**

In Fig. 7, the PLR remains stable between 20% and 30% after which it increases linearly. Even though the application traffic sent and received are almost the same (see Fig. 6) the PLR is very high. Although we have the same number of nodes, having more routers increases the traffic considerably in the network. When a node needs to send data, it broadcasts a route request message (RREQ). All the neighbouring routers receive this message and broadcast it until the message reaches the destination. The receiver receives the RREQ message from all it surrounding routers and sends back a route reply (RREP) message through the shortest route (in terms of number of hops). So using more routers in a static scenario does not improve the performance.

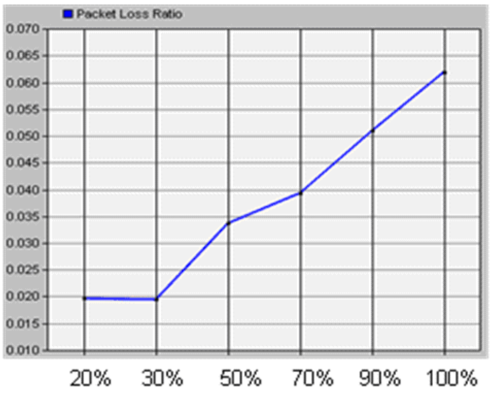


**Fig. 7: PLR – Router Influence scenario**

In Fig. 8 the ETE increases with increase in router percentage. Thus it can be concluded that the number of router has no impact on the application traffic received. However, the PLR and the ETE increases as the number of routers increases. Whether these values of ETE and PLR are critical depends on the application.

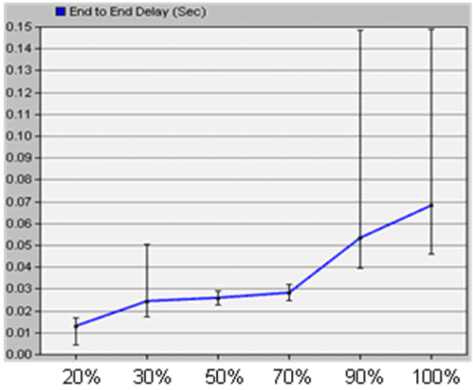


**Fig. 8: ETE - Router Influence scenario**

### 4.3.3 Mobile Scenarios

Two mobile scenarios are defined: random trajectory mobility model and disaster management mobility model. The random trajectory model simulates a general mobile scenario where the movement pattern of the nodes is defined by self-defined trajectories, unlike the ‘random waypoint mobility model’. The Disaster Management Model represents realistic movement pattern of the nodes during emergency response.

#### 4.3.3.1 Random Trajectory Mobility Model

In the random trajectory mobility model, we defined our own trajectories for all the nodes. Each node moves with predefined trajectories throughout the network area. The network area is set to 300m x 300m, simulation duration is set to 300 sec, and the node transmission range is 100m. The total number of end devices is set to 70. Furthermore 10 static routers are distributed to cover the network area. Fig.9 shows the random trajectory mobility model snapshots at the beginning and at the end of the simulation. The nodes are set in groups at the beginning of simulation. The white arrows define the trajectories of the nodes. As the simulation progresses the nodes move to predefined positions within the network area according to their trajectories.

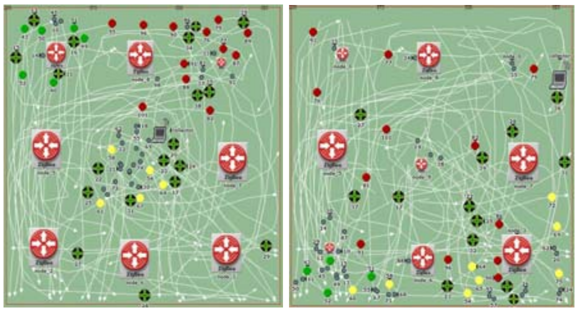


**Fig. 9: OPNET network panel at the beginning and end of simulation**

Figure 10 shows the application traffic sent compared with the application traffic received in the random trajectory mobility model. At the simulation start time, the application traffic sent and received are same. But after about 20 seconds the application traffic lost increases and at the end of the simulation only 1050 bit/sec is received of the 2500 bit/sec sent. In this scenario, only few nodes transmit packets at the simulation start time and as time progresses more nodes transmit and begin to move throughout the network area. The application traffic lost is high because the end devices send packets to their destination through a parent router and in most cases it could be that the node (after sending its packet) moves out of the range of its parent router thereby failing to receive the acknowledgment message and the packet is considered as lost. It can be seen from Fig. 10 that mobility severely affects ZigBee networks.

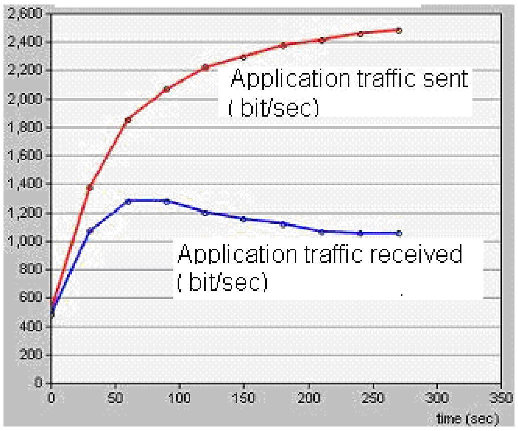


**Fig. 10: Application traffic sent vs. application traffic received for Random Trajectory Model**

#### 4.3.3.2 Disaster Management Mobility Model

In this mobility model the realistic movement pattern of the nodes at the disaster site are setup based on the emergency response process followed (see Section 2). This mobility model uses online node scaling ie. the nodes join the network with increase of time and a few nodes to leave the network toward the end of the simulation. Two models are simulated: 50%-router-model and 100 %-router-model.

#### 50%-Router-Model

In this model 50 % of the total nodes are set as routers. The network area is set to 500m x 500m with a total of 100 nodes, of which 50 nodes are end devices (all patient nodes in DAN are set as end devices). The other 50 nodes are set as routers (15 routers are static representing the static reference nodes and 35 routers are mobile representing the emergency doctors in DAN). Furthermore, the node transmission range is set to 100m and the simulation duration is 2000sec.

In the disaster management mobility model, the network area is divided into four zones: the disaster zone (DZ), the triage zone (TZ), the treatment zone (TTZ), and the transport zone (TRZ). At the beginning of the simulation, 15 static routers are distributed over the area, to make sure that the whole network area is covered. At the simulation start all patient nodes are located at the danger zone and the emergency doctor nodes are progressively entering the network. Some emergency doctor nodes move to the danger zone.

Some doctors move directly to the triage zone and start with the triaging as soon as the first patients arrives the triage zone. The rest of the doctor nodes are moved to the treatment zone and then to the transport zone. By this configuration, it's made sure that there will always be a few doctor nodes in each zone at any time unit.

Fig. 11 shows a snapshot of the network at the beginning of the simulation. The four zones are marked by rectangles and the white lines define the trajectories of the nodes. As the trajectories show, most of the nodes have the following flow: DZ → TZ → TTZ → TRZ. But there are some exceptions, especially for emergency doctors. Also some patient nodes leave the network after they have been transported to the TZ or to the TTZ. The red, green, yellow and blue colour nodes represent the patients. The nodes with a cross represent the emergency doctors. A patient is assisted by at least one doctor from the DZ to the TZ after which the doctor moves back to the DZ. The patients are triaged by the doctors who are already present in the TZ. According to their triage class priority (red first, followed by yellow and green) the patient are shifted to the TTZ and finally to the TRZ.

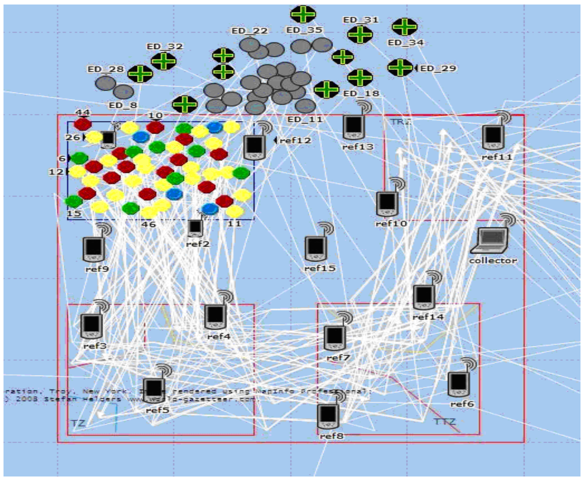


**Fig. 11: 50%-Router-Model snapshot at the start of simulation**

Fig. 12 shows the state of the network after 1000 sec. All mobile nodes have already moved from DZ to TZ and from TZ to TTZ. There are also some patient nodes in TRZ and some have already been transported (evacuated) to the hospital.

At the end of the simulation, the site is almost empty and there are no more patient nodes. There are only doctor nodes and static reference nodes in the network.

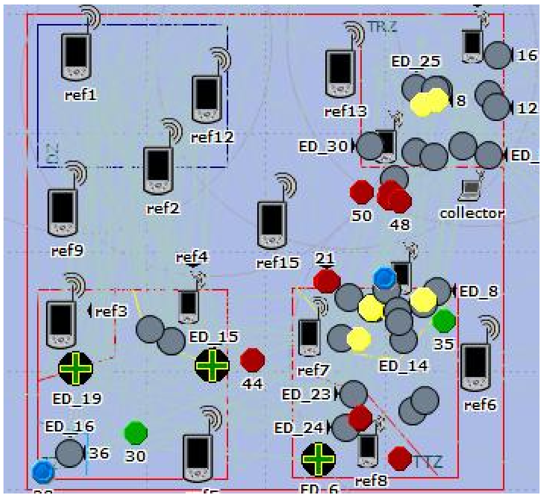


**Fig. 12: 50%-Router-Model snapshot after 1000 sec of simulation**

In Fig.13, at the beginning of the simulation not all nodes are sending and not all nodes are already moving thus the application traffic lost is less. At 500 seconds all the nodes have joined the network. Since all the patient nodes are sending and the network is highly mobile the application traffic lost is high. After 1650s no more patient nodes are present and the network contains only few doctor nodes, static anchor nodes and so almost all the application traffic sent is received. It can be seen that the mobility affects ZigBee network, but in comparison with the results of random trajectory mobility model the performance (in term of application traffic received) is better for the 50%-Router Model which can be due to the varying mobility patterns.

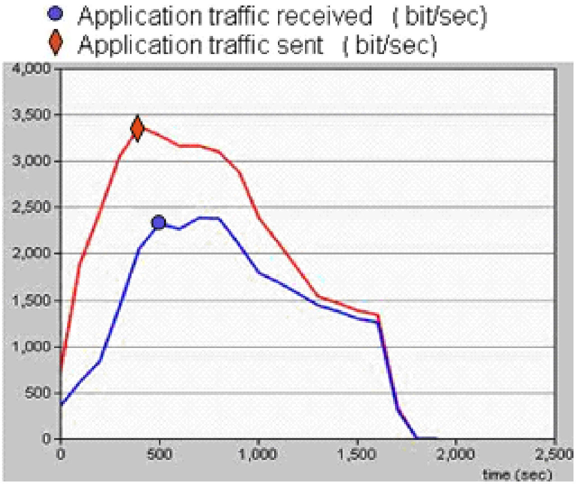


**Fig. 13: Application traffic sent vs. application traffic received for the 50%-Router-Model**

**100%-Router-Model**

The simulation setup for this model remains the same as that of 50%-Router-model except that all patient nodes are now routers (i.e 100% routers in the network).

In Fig. 14, the application traffic received for the 50%-Router-Model and 100%-Router-Model are compared to find the impact of routers in a mobile disaster management scenario. The application traffic received is better when all the patient nodes are set as routers because routers are more robust than end devices against the influence of mobility.

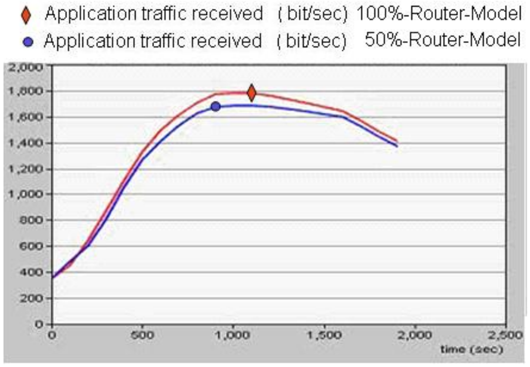


**Fig. 14: Comparison of application traffic received for 50%-Router-model and 100%-Router-model**

In Fig. 15, the PLR when patient nodes act as end devices and the PLR when patient nodes act as routers are shown. With more routers, fewer packets are lost.

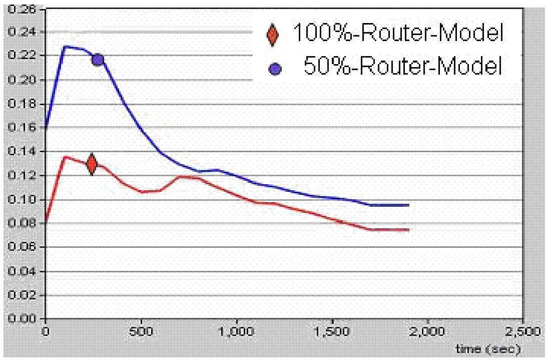


**Fig. 15: Comparisof PLR for 50%-Router-model and 100%-Router-model**

The mobile scenario performs well in presence of routers, which is in contrast to static scenario. Routers dispose of functionalities that make them robust against mobility effects.

### 4.3.4 Results

Scalability: In a static scenario the network scales well, as far as the density of the network is concerned ie. the network scales well up to 150 nodes in a 300m x 300m area. When the number of nodes is increased above 150 the application traffic loss is high (about 400bit/sec). In a mobile scenario, the network does not scale as well as in the static scenario due to the high PLR.

Influence of routers in the network: In a static scenario using more routers doesn't improve the network, rather it degrades the performance (PLR, ETE). But in mobile scenario, the use of more routers considerably improves the performance of the network. So in a static network only the number of routers required to cover the network area, can be used. In a mobile network it may be suitable to use as many routers as possible.

Mobility: The performance of ZigBee network is affected when the nodes are mobile, especially when the mobile node density is high. However, the performance could be improved if the number of routing capable devices is increased.

ZigBee is basically suitable for the DAN, even though more detailed investigations are required. As part of future work the influence of node density and effect of more than one collector node will be investigated.

## 5. ZigBee Mesh Network

In order to perform the RSSI based localization analysis and to setup an air temperature monitoring demonstrator we have developed a 20 node ZigBee mesh network.

### 5.1 ZigBee-ready temperature Sensor Node

The temperature sensor node [21] is designed with a power supply, Texas Instruments (TI) CC2431 System on Chip (SOC) and chip antenna. The CC2431 (see [12], [13]) consists of the location engine, RF transceiver, an enhanced 8051 MCU, and a temperature sensor.

The current consumption of this sensor node (see Figure 16) is shown in Table 1. The ZigBee router is always active, leading to higher current consumption. In the communication deactivated state (8051 core active, RF transceiver off), the end device is in sleep mode leading to lower current consumption. In DAN the nodes may require a battery lifetime of around 5 hours to one week. The current consumption results show that the sensor node can be used as the patient nodes or doctor's node or routers and last at least for few days.

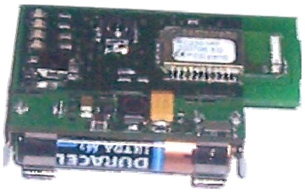


**Fig. 16. ZigBee-ready Temperature Sensor Node**

**Table 1. Current Consumption of Temperature Sensor Node**

| ZigBee Sensor Node (supply voltage = 3.0 V) | Router | End Device |
|---|---|---|
| Data communication activated | 35.8 mA | 27.8mA |
| Data communication deactivated | - | 20.4 mA |

## 6 Patient Localization during Emergency Response

A patient localization solution has to be developed, that provides real time patient's location to the medical / organizational officers and in tandem with the emergency response system facilitates efficient logistics at the disaster site. Each patient node (blind node) runs a localization algorithm and updates the monitor station with its current location information.

The requirements [22] for patient tracking that DAN must comply with are: handle the different environments (both outdoor and indoor); use little or no special infrastructure (static anchor nodes) due to lack of deployment time at the site; track 30-500 patient nodes moving with varying speed (0 to 3 m/s); attain an accuracy of 5 to 10 m with a max latency per node of 5 seconds; be scalable and robust.

The main challenge here is to handle the varying mobility and different environment with adverse RF conditions and also use minimum or no infrastructure.

### 6.1 Related Work for Patient Localization

Localization systems like Active Badge [23], Cricket [24], RADAR [25] required a lot of infrastructure and GPS [26] is not suitable for Indoor. RFID based solutions like SpotON [27] are not suitable for us since they demand high anchor node density, works in short range and needs a fixed infrastructure.

## 6.2 Analysis of a Received Signal Strength Indicator (RSSI) based Localization system

An analysis of the CC2431 hardware based location solution is undergone to find its suitability to the DAN.

The CC2431 Location Engine [12], [13] hardware from Texas Instruments (TI) implements a distributed computation algorithm that uses RSSI values from reference nodes whose coordinates are known to calculate the location of the blind nodes whose coordinates are to be determined. Performing location calculations at the node level reduces network traffic and communication delays otherwise present in centralized computation approach.

### 6.2.1 RSSI based Localization- Functionality

The basis of this radio-based positioning solution is the relation between the distance from the transmitter and the received signal strength (see equation 1) considering the assumption that the propagation of the signal is approximately isotropic [14].

$$RSSI = -\left(10n\log_{10} d + A\right). \qquad (1)$$

The parameters *A* and *N* determine the exactness of the blind node location. *A* is an empirical parameter determined by measuring the absolute RSSI value in dbm of an omni-directional signal at a distance of one meter from the transmitting unit. The parameter *N* is defined as the path loss exponent and describes the rate at which the signal strength decreases with increasing distance from the transmitter [14].

The positioning of blind node is done by averaging at least three and a maximum of eight references nodes. Localization takes place in two steps, which are repeated in cycles. The first step is the Burst-phase, in which the blind node broadcast a sequence of packages, requesting the reference nodes for their position and the averaged received signal strength of the packets sent to them. In the second step the eight best received references will be sorted according to their signal strength and handed over along with the parameters *A* and *N* values to the blind node (localization hardware) which solely calculates its location [14].

### 6.2.2 Analysis

A 120 x 120 meter indoor area is covered by rectangular grid of sixteen reference nodes each separated by 40 meters. ZigBee ready sensor nodes are used as reference and blind nodes. The coordinator is a ZigBee hardware dongle enabled laptop running Location Graphical User interface (GUI) software to display the positions of nodes in the site map. The location (x,y) of the reference nodes are manually configured via Z-location engine a display and control software [14].

The value of *A* is measured as 49 and the value of *N* is selected as 3.875 from the vendor specification, based on the empirical measurement that best fits the environment. Five blind nodes are moved within this grid to 10 different positions (center of grid, corners) at an interval of 20 seconds and the corresponding position coordinates are measured via the Location GUI.

The actual location and the measured location of the blind nodes are compared. The average deviation of the measured values from the actual values, for 10 different readings is calculated for each blind node and an accuracy of 2 meters is obtained. The time (burst phase plus computation phase) for every blind node to calculate its location is measured as 2 seconds. It is noticed that as people/objects moved in the indoor area the blind node location estimation was varying and unstable.

The analysis of the CC2431 localization solution reveals that the blind node location estimation is unstable and needs large number of reference nodes for considerable performance. So this system is also not suitable [21] for our scenario. Therefore we are currently developing a new localization solution for our scenario.

## 7 Air Temperature Monitoring

In DAN, the ZigBee mesh network consists of sensor nodes to sense local weather conditions like air temperature, wind speed and wind direction. The wind speed and wind direction information can enable the emergency responders to identify zones filled with colorless toxic gases. This can allow the responders to quickly shift the patients away from these danger zones and reduce respiratory hazards. In this paper we have focused on air temperature monitoring only.

The air temperature at the different zones of the disaster site varies throughout the disaster management process. We have implemented an air temperature monitoring mesh network that provides real time temperature information at the disaster site. These data are collected by the monitor station which runs the visualization software. The visualization software runs the temperature zone algorithm and displays various temperature zones at the disaster site.

### 7.1 Temperature Zone Algorithm

The temperature zone algorithm [21] is a dynamically responding mechanism [16] based on localized temperature events. The functionality of this algorithm is to estimate and display the various temperature zones present at a disaster site. The algorithm is implemented in Python using NumPy (Numerical Python) for matrix processing and wxWidgets / wxPython for the graphical user interface.

The inputs to this algorithm are:

- The location coordinates of the temperature sensor nodes
- The measured temperature values from the sensor nodes

The output from this algorithm is:

- The estimated temperature zone mapping
- The current algorithm implementation detects only two zones: danger zone (30°C to 50°C), normal zone (20°C to 29°C).

#### 7.1.1 Functionality

The functional block diagram of temperature zone algorithm is as shown in fig.17.

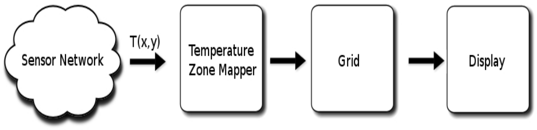


**Fig. 17. Functional Block Diagram of Temperature Zone Algorithm**

**Temperature Zone Mapper**

The location of the temperature sensors and their corresponding temperature readings are given as input into the temperature zone mapper. For every location coordinate received, the temperature zone mapper uses the corresponding temperature reading to calculate a cosine probability distribution of the temperature centered at the given location coordinate and weighted with the temperature value for that location coordinate.

For example, assuming the algorithm gets a temperature value of 30°C at a location coordinate (20,20) then the temperature zone mapper plots a probability distribution function (as shown in fig. 18) centered at (20,20). As the distance increases away from the location (20, 20) the probability for the temperature to be 30°C is less.

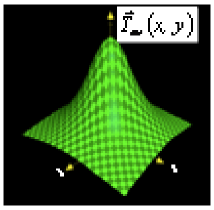


**Fig. 18 Temperature Probability Vector**

**Grid block**

The Grid block is a three dimensional representation of the display block where time is the third dimension. It generates a three dimensional $\vec{T}(x,y,t)$ vector as output for every location coordinate given as input and stores this value in the grid. The value of $\vec{T}(x,y,t)$ in the grid is calculated by adding the output of the temperature zone mapper to the decayed previous grid value at that location (see equation 2). The values of the grid are instantaneously calculated and updated at each time step for all location coordinate inputs. This enables the tracking of temperature variation at a particular location. The grid functionality is mathematically shown below.

$$\vec{T}\,(x,\,y,t\,) = \sum \vec{T}_{ev}\,(x,\,y\,)\cdot decay\;\;(t-t_{ev}\,) \qquad (2)$$

For efficient implementation, a decreasing exponential function is used for delay.

$$decay(x) \;\; = \; a^{\,x}\,,0 \;<\; a \;<\; 1 \qquad (3)$$

By substituting the decay function in $\vec{T}(x,\,y,t)$ and expanding the summation, we get

$$\vec{T}(x,\,y,t) \,=\, \vec{T}_{ev_{\,1}}(x,\,y)\cdot a^{\,t-t_{ev_{\,1}}} \,+\, ... + \vec{T}_{ev_{\,n}}(x,\,y)\cdot a^{\,t-t_{ev_{\,n}}} \qquad (4)$$

Extraction of $a$ summands between *1* and $n-1$ yields

$$\vec{T}(x,y,t)=a\left(\vec{T}_{ev_1}(x,y)\cdot a^{\,t-1-t_{ev_1}}+...+\vec{T}_{ev_{n-1}}(x,y)\cdot a^{\,t-1-t_{ev_{n-1}}}\right)+\vec{T}_{ev_n}(x,y)\cdot a^{\,t-t_{ev_n}} \qquad (5)$$

By substitution of $\vec{T}(x,\,y,t-1\,)$ for events from 1 to $n-1$, we get

$$\vec{T}(x,\,y,t) \,=\, a\cdot\vec{T}(x,\,y,t-1\,) + \vec{T}_{ev_{\,n}}(x,\,y)\cdot a^{\,t-t_{ev_{\,n}}} \qquad (6)$$

When $t \,=\, t_{ev_{\,n}}$ , this is further simplified to

$$\vec{T}(x,\,y,t) \,=\, a\cdot\vec{T}(x,\,y,t-1\,) + \vec{T}_{ev_{\,n}}(x,\,y) \qquad (7)$$

**Display**

Temperature zones are displayed by assigning a color to each location of the output of the grid based on temperature range. Danger zone is displayed in red and normal zone is displayed in blue. The resulting color of each output is calculated by multiplying the output of the grid with the assigned color for its corresponding temperature range.

### 7.2 Demonstrator

A 20 node ZigBee mesh network is set up covering an indoor area of 120 x 120 meters. Each temperature sensor node (see Section 5.1) was displaced by around 40 meters to form a rectangular grid of static routers and end devices. The location coordinates of the nodes were manually configured. The temperature values with its corresponding location coordinates are periodically transmitted to the monitor station running the visualization software. This visualization software consists of a display and summary panel. The display panel shows the various temperature zones in different colors over a map of the site where the mesh network was deployed. The summary panel provides the temperature values of the nodes at its corresponding location textually.

When the mesh network was started the nodes sensed a room temperature value which falls under the normal zone and so the display panel indicates the entire deployment site in blue color (see figure 19-a). But as the mesh network was running we gradually heated the nodes at the bottom half of the grid above 30°C using the heat gun. We were able to see in real time, the bottom half of the deployment sites map (in the display panel) changing to red color indicated a danger zone (see figure 19-b).

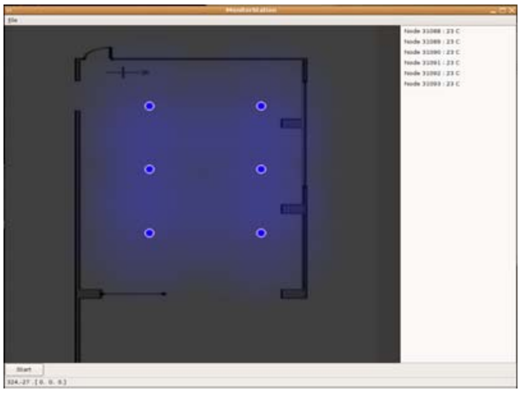
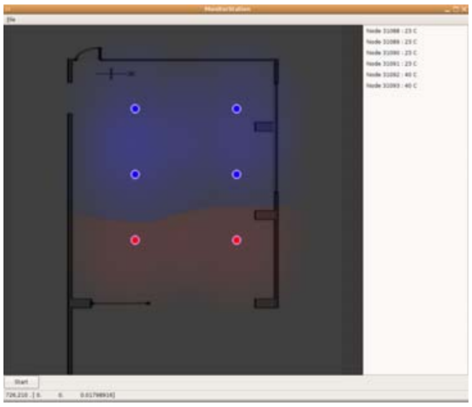

Node 31088 : 23 C
Node 31089 : 23 C
Node 31090 : 23 C
Node 31091 : 23 C
Node 31092 : 23 C
Node 31093 : 23 C

**Fig.19 a) Normal Zone Visualization**



```
Node 31088 : 23 C
Node 31089 : 23 C
Node 31090 : 23 C
Node 31091 : 23 C
Node 31092 : 40 C
Node 31093 : 40 C
```

```
726,210 : [ 0.      0.      0.01798916]
```

**Fig.19 b) Danger Zone Visualization**

### 7.3 Computation analysis of Temperature zone algorithm

The temperature zone algorithm was only evaluated using a mesh network of 20 nodes. In order to find the computation efficiency of this algorithm in large scale mesh network the algorithm is analyzed by giving an event log file containing the events (temperature values at different location coordinates) as input (see fig. 20).

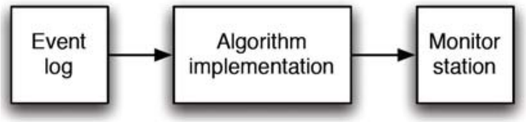


**Fig. 20 Analysis Model for Temperature Zone Algorithm**

1006 randomly generated events were fed as input to the algorithm at once, to estimate the performance of the temperature zone algorithm. The analysis was run on an Intel Pentium 1.6GHz processor and took 484.65 seconds of CPU time (396.7s user and 33.7s system) to complete its run. From this, we can infer that on a comparable or better processor, the temperature zone algorithm will be able to process 2 or more events per second.

## 8 Conclusion

A new emergency response system based on the location aware DAN is proposed for assisting the ER's at the disaster site. The ZigBee simulation results for scalability, mobility and number of routers show that ZigBee is basically suitable for DAN even though detailed investigations will have to be done. The current consumption results of ZigBee-ready temperature sensor node indicate that they can have a lifetime of at least few days as patient wearable nodes or temperature sensor nodes in DAN. The analysis of RSSI based localization solution shows that it's not suitable to DAN due to its need for large infrastructure and unstable blind node location. So a new localization solution for DAN will have to be developed as future work. The result of the temperature zone algorithm shows its computational efficiency and its effectiveness in alerting the responders about danger zones. The patients can therefore be quickly evacuated from the disaster site. Further expansion of the system and its testing with large scale networks is part of the future work.

## 9 References

1 Chandra-Sekaran, A. and Nwokafor, A. and Johansson, P. and Mueller-Glaser, K. D., and Krueger, I. ZigBee Sensor Network for Patient Localization and Air Temperature Monitoring During Emergency Response to Crisis. The Second International Conference on Sensor Technologies and Applications, SENSORCOM 2008, France.

2 Hazardous material management http://www.epcra.state.mn.us/hazmat_info/scene_safety.html

3 "Gesetz über den Rettungsdienst sowie die Notfallrettung und den Krankentransport durch Unternehmer (Rettungsgesetz NRW - RettG NRW) Vom 24. November 1992"- Emergency Response Law in the German state NRW.

4 San Diego Disaster Drill http://www.calit2.net/newsroom/article.php?id=745

5 Gao, T. and Greenspace, D. and Welsh, M. and Radford, R. J. and Alm, A. Vital sign monitoring and patient tracking over a wireless network. Johns Hopkins University, Applied Physics Laboratory.

6 Gao, T. and White, D. A next generation electronic triage to aid mass casualty emergency medical response. Johns Hopkins University, Applied Physics Laboratory.

7 "ZigBee Specification 2006" ZigBee Alliance, Tech. Rep.Document 053474r13, 2006.

8 ZigBee Alliance website http://www.zigbee.org/

9 Hac, A. Wireless Sensor Network Designs. University of Hawaii at Manoa, Honolula, USA.

10 Zhao, F., and Guibas, L. Wireless sensor networks, an Information processing approach.

11 Chandra-Sekaran, A. and Mueller Glaser, K. D. and Stork, W. and Picioroaga, F. and Brinkschulte, U. Towards a self-organizing wireless hospital area network. World Congress in Medical Physics and Bio-Medical Engineering, Seoul, South Korea, 2006.

12 Texas Instruments CC2431 - System-on-Chip for 2.4 GHz ZigBee/ IEEE 802.15.4 with Location Engine: Data sheet

13 Texas Instruments CC2430 - System-on-Chip for 2.4 GHz ZigBee/ IEEE 802.15.4: Data sheet

14. Texas Instruments CC2431 Location Engine: Application Note AN042

15 “NRW, Behandlungsplatz-bereitschaft: Konzept BHP-B 50 NRW. Innenministerium des Landes Nordrhein-Westfalen, April 2006”-NRW state treatment zone concept BHP-B 50 NRW.

16 Tatomir, B. and Rothkrantz, L. Ant based mechanism for crisis response coordination, Proceedings of Ant Colony Optimization and Swarm Intelligence, ANTS 2006

17 Ferrari, G. and Medagliani, P. and Martaló, M. Performance analysis of ZigBee wireless sensor networks with relaying. Wireless Ad-hoc and Sensor Networks (WASN) Laboratory, Department of Information Engineering University of Parma, Parma, Italy.

18 Nia-Chiang Liang, Ping-Chieh Chen, Tony Sun, Guang Yang, Ling-Jyh Chen, and Mario Gerla, Impact of Node Heterogeneity in ZigBee Mesh Network Routing. 2006 IEEE International Conference on Systems, Man and Cybernetics, October 8-11 2006, Taipei, Taiwan.

19 Ling-Jyh, Tony Sun, Nia-Chiang Liang, An Evaluation Study of Mobility Support in ZigBee Networks. Institute of Information Science, Academica Sinica.

20 OPNET online documentation. http://www.opnet.com/,

21 Ranjan, G. and Kumar, A. and Rammurthy, G. and Srinivas, M. B. A natural disaster management system based on location aware distributed sensor networks. MASS2005, 0-7803-944-6/05/.

22 Lechtleutner, A. Disaster Management process Investigation- University of Applied Sciences, Rescue Engineering Department.

23 Want, R. et al., The active badge location system, ACM Trans. Inf. Syst., pages 91-102, 1992

24 Priyantha, N. B. and Chakraborty, A. and Balakrishnan, H. The Cricket Location Support System, ACM Press, Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom’00). pages 32-43, 2000

25 Bahl, P. and Padmanabhan, V. RADAR: An inbuilding RF based user location and tracking system, Proceedings of IEEE Infocom. vol. 2, pages 775-784, 2000

26 Global Positioning System http://www.gps.gov/

27 Hightower, J. and Borriello, G. SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength

28 ns-2 Simulator http://www.isi.edu/nsnam/ns/

# Securing Wireless Sensor Networks: Introducing ASLAN - A Secure Lightweight Architecture for WSNs

Michael Collins[1], Simon Dobson[2], Paddy Nixon[2]
*Systems Research Group, School of Computer Science and Informatics,*
*UCD Dublin, Ireland*

[1]*michael.collins@comp.dit.ie*
[2]*{simon.dobson, paddy.nixon}@ucd.ie*
[1]*http://www.comp.dit.ie/mcollins*
[2]*http://www.csi.ucd.ie/Staff/AcademicStaff/{sdobson, pnixon}*

***Abstract*—Wireless sensor networks consist of many small, inexpensive devices that have constraints in coverage, bandwidth, storage resources, communications ability and processing power. Therefore security issues are a critical concern due to possible exposure to malicious activity and potential threats. As a result of the physical constraints in sensor nodes, traditional cryptographic techniques are not suitable to operate on such networks where security requirements are of crucial importance. This raises serious concerns on finding methods to protect sensor nodes from adversaries, to quickly segregate those that have been attacked, and allow the network to reform. To address the security vulnerabilities in a wireless sensor network, this paper proposes a secure lightweight architecture (ASLAN) that takes account of the constraints of sensor networks. With the aid of a base station, a hierarchical network topology is formed allowing end-to-end communication between sensor nodes. ASLAN also supports identifying and isolating aberrant sensor nodes.**

***Keywords*—Wireless Sensor Network; Architecture; Security; Protocol***

## 1. Introduction

Wireless sensor networks have emerged as a technology that are being quickly adopted due to their flexibility and use in a variety of environments. However, they consist of small, inexpensive devices or nodes that have severe constraints such as limited bandwidth, limited processing power, short battery life, small storage capability and are physically prone to external threats [1]. Even with all the advantages that wireless sensor networks provide such as fast deployment and configuration, the constraints of the sensor nodes makes them extremely vulnerable to various security threats [2]. These include attacks that target a specific node with endless communication in order to exhaust its limited battery life and also the physical vulnerability of the sensor nodes within a hostile environment, e.g., a military battlefield. Unfortunately, cryptographic techniques such as Public Key Infrastructure (PKI) [3], which is widely used in traditional wired networks, is not suitable to operate on sensor networks to enable secure data communication. Therefore, this makes sensor networks susceptible to attack and also very difficult to identify and deal with nodes that act maliciously.

In this paper, the authors propose a secure lightweight architecture for wireless sensor networks (ASLAN) that provides the desired security mechanisms to address the identified security threats. ASLAN employs the notion of a base station that is used as a base class in a hierarchical network configuration. The paper describes how this network topology is formed.

The structure of this paper is as follows: Section 2 discusses some of the various security issues associated with wireless sensor networks, Section 3 gives a summary of previously related research in this area, Section 4 introduces our proposed architecture (ASLAN) which includes the network topology formation, details of our security protocol for identifying and isolating aberrant nodes, and the secure routing mechanism of data communication in the sensor network, Section 5 discusses our future work to implement and evaluate ASLAN, and finally, Section 6 concludes this paper.

## 2. Security issues in Wireless Sensor Networks

Wireless sensor networks have innate constraints compared to traditional wired networks that prevent many security mechanisms being able to operate. This section gives examples of such constraints and discusses the threats and issues typically encountered.

### 2.1 Introduction

Traditional security mechanisms normally require high processing capability, and large memory and storage requirements. Such resources are not available in nodes in a wireless sensor network. As a result of these constraints, designing effective security mechanisms is more difficult than for a wired network. Examples of these constraints include:

(a) Small memory

The memory in a wireless sensor node is very limited memory with small storage capacity. As a result, any security mechanism to be designed and run within a sensor network will have limitations and not be as robust as one for a wired network.

(b) Reduced energy levels

Designing security mechanisms for wireless sensor networks must consider the reduced energy levels that are implicit with sensor nodes. When a sensor node is deployed, its energy source is usually a battery so it is critical to design security features that are not memory or power intensive in order to prevent the battery life being exhausted quickly. However, security features will consume extra energy that that required for normal operation, for example cryptographic techniques, and this may be detrimental to the sensor node's time to live.

(c) Communication problems

There is an inherent problem with wireless communication in that data can get intercepted, lost and is generally prone to attack. Since sensor nodes are usually deployed in mass numbers and form a sensor network, lots of data will be transmitted and received between sensor nodes resulting in heavy network traffic. This makes it likely that some data packets will be damaged or lost. Unlike traditional wired networks where protocols deal with such situations, the nodes in a wireless sensor network do not have the resources available to resend data packets.

(d) Physical security

Sensor nodes are generally small devices that are not very robust. This makes them prone to damage and vulnerable to attack in harsh and hostile environments where an attacker can potentially capture or damage a node. After a sensor node has been deployed, they are susceptible to issues such as weather conditions, undesired natural phenomena, deliberate attack by an adversary, and power exhaustion. It is almost impossible to have any control of these issues.

### 2.2 Threats and Issues in Wireless Sensor Networks

Most of the threats and attacks against security in wireless networks are almost similar to their wired counterparts while some are exacerbated with the inclusion of wireless connectivity. In fact, wireless networks are usually more vulnerable to various security threats as the unguided transmission medium is more susceptible to security attacks than those of the guided transmission medium [4].

Attacks can also be launched at any point in the network and that certain attacks may be more effective at different layers of the communications protocol. Table 1 depicts the various attacks that can be launched at different layers of the communications stack [5].

Table 1. Attacks on Communications Stack

| Layer | Attack |
|---|---|
| Physical Layer | DOS – Jamming, Tampering Sybil |
| Data-link Layer | DOS – Collision, Exhaustion, Unfairness<br>Interrogation<br>Sybil – Data aggregation, Voting |
| Network Layer | DOS – Neglect & Greed, Homing, Misdirection (Spoofing), Black Holes, Flooding<br>Sybil<br>Wormhole Attack |
| Transport Layer | DOS – Flooding, De-synchronization |

### 2.2.1 Denial of Service (DoS) attack

A DoS attack tries to exhaust the resources available to the victim node by sending unnecessary data packets and therefore prevents legitimate network users from accessing services or resources they desire [6, 7]. They take many forms as shown in Table 1 that prevent the network performing its expected functions. There are several types of DoS attacks in a wireless sensor network that include jamming, power exhaustion, service greed, and network flooding. Mechanisms that attempt to prevent a DoS attack may include payment for network resources, and robust authentication.

### 2.2.2 Sybil Attack

A Sybil attack is one in which a sensor node mimics the identity of more than one other legitimate nodes [8, 9]. The Sybil attack specifically targets situations where large tasks are divided into subtasks and distributed among several sensors in order to complete the task. The Sybil attack operates by attacking the distributed storage, routing mechanism, data aggregation, voting, fair resource allocation and misbehaviour detection of nodes [9]. All peer-to-peer networks are susceptible to a sybil attack. However, the detection of sybil nodes is difficult [9].

### 2.2.3 Sinkhole Attack

In a sinkhole attack, the adversary's goal is to lure nearly all the traffic from a particular area through a compromised node, creating a metaphorical sinkhole with the adversary at the center. Since nodes on, or near the path that packets follow have many opportunities to tamper with application data, sinkhole attacks can enable many other attacks (for example, selective forwarding) [10].

Sinkhole attacks operate by trying to attract network traffic and pass their data through a compromised node. For example, a compromised node could falsely advertise that it offers an efficient route from one point of the network to another. Due to either the real or imagined high quality route through the compromised node, it is likely each neighbouring node of the adversary will forward packets through the adversary, and also propagate the attractiveness of the route to its neighbours. Effectively, the adversary creates a large "sphere of influence", attracting all traffic from nodes several (or more) hops away from the compromised node [10].

One reason for mounting a sinkhole attack is that it offers an easy process for performing selective forwarding. Thus, most of the traffic in the vicinity of the compromised node will flow through it and the compromised node can then select whichever packets it desires to modify or suppress.

### 2.2.4 Wormhole Attack

A wormhole attack is one whereby an attacker tunnels messages received in one part of the network over a low latency link and replays them in a different part. A simple example of this attack is a single node situated between two other nodes forwarding messages between the two of them. However, they more commonly involve two distant malicious nodes colluding to understate their distance from each other by relaying packets along an out-of-bound channel available only to the attacker [11]. Wormholes may also be used simply to convince two distant nodes that they are neighbours by relaying packets between the two of them.

### 2.2.5 Attack on transit information

When sending data in a wireless sensor network, the information may be spoofed, modified, replayed or removed. An attacker can monitor the traffic being routed through the network and may interrupt, intercept, modify or fabricate data packets thereby sending inaccurate information to the recipient [12]. Due to the resource constraints of nodes, adequate security mechanisms to deal with such issues are difficult to implement.

There has been much research carried out to identify all the possible threats and issues to wireless sensor networks. However, these threats evolve over time with new security concerns emerging frequently that need to be addressed.

## 3. Related work

Research into security in wireless sensor networks has been conducted over recent years. This section summarizes some of this research.

Chen et al. [13] were among the early proposers of a security model for communication between a base station and the sensor nodes in a wireless sensor network. The model consists of two security protocols for the deployment of sensor networks. The first protocol is called "base station to mote confidentiality and authentication" and describes how an efficient shared-key algorithm be used to guarantee authenticity and privacy of information passing on the network.

The reason they use a shared-key algorithm is because of its low consumption of resources which is ideal for use on small, resource-constrained sensor nodes. The second protocol is called “source authentication” that implements a hash chain function to achieve mote authentication.

Perrig et al. [14, 15] proposed a model called SPINS which is a collection of protocols for sensor networks. It integrates SNEP (Secure Network Encryption Protocol) and μTESLA (micro-Time Efficient Streamed Loss-tolerant Authentication). SNEP supports end-to-end security by providing data confidentiality and two-way data authentication with minimum overhead. μTESLA, a micro version of TESLA, provides authenticated streaming broadcast and keeps computation costs low by using only symmetric cryptography.

However, the SPINS model leaves some unresolved security questions such as the security of compromised nodes, Denial-of-Service (DoS) issues, and network traffic analysis issues. Furthermore, this protocol assumes the static network topology ignoring the ad hoc and mobile nature of sensor nodes [16].

Undercoffer et al. [17] proposed a light weight security protocol operating in the base station of a sensor network framework. In this model, the base station can detect and remove a sensor node if it behaves anomalously or becomes compromised. However, there are no security measures specified on dealing with an attack such as the interception of communication between nodes.

Eschenauer et al. [18], proposed a key pre-distribution model where each sensor in the network receives a random subset of keys from a large key pool before they are deployed. This key pool is held by a base station. In order for communication to take place between nodes, a common key must be selected from each node’s subset of keys and to use this as their shared key.

Chan et al. [19] extended the model in [18] in which they developed three key pre-distribution schemes; q-composite, multipath reinforcement, and random-pairwise keys schemes. Each of these schemes enabled the base station to pre-distribute keys to the nodes on deployment.

Du et al. [20, 21] introduced two different schemes. The first scheme proposed using pairwise key pre-distribution. Under this scheme, there would be a much higher payoff for an attacker to spend the large amount of time and resources required to compromise nodes in a large-scale sensor network than a smaller scale network. The second scheme proposed a key management mechanism whereby keys are issued to sensor nodes based on deployment knowledge which stores the position of sensors prior to their deployment. However, since neighbouring nodes must use the same key (symmetric cryptography) for communications, the problem exists in that there is no way to know the exact locations of neighbour nodes due to the randomness of node deployment. However, it is feasible to know a set of likely neighbouring nodes so the use of a random key pre-distribution technique is possible using [18].

Undercoffer et al. [17, 22] proposed a system whereby the base station in the sensor network was used to authenticate the sender of data packets. However, this model makes the assumption that the base station operates under perfect conditions and can detect anomalous nodes or nodes acting maliciously. This is done by storing statistics of node activity. The model also implemented security mechanisms at the packet level where each data packet is encrypted with shared keys to ensure data integrity and source authentication.

There are assumptions made in these protocols that have been replicated in the proposal of ASLAN in this paper. It includes the assumption that the base station is always dependable and that all data stored in a sensor node’s memory is secure.

## 4. Architecture

This section describes the proposal of ASLAN. The proposed architecture incorporates the following:

1. Network topology organisation
   a. Formation
   b. Inserting additional nodes into the network
2. Key management
3. Identifying and isolating aberrant nodes
4. Secure routing

### 4.1 Network topology organisation

ASLAN supports a network topology that is organised into two distinct roles.

#### 4.1.1 Formation

The architecture considers that the network is composed of sensor nodes, cluster leaders and a base station. The base station is the only interface between the sensor network and the outside. Similar to Undercoffer et al. [17, 22], it is assumed to operate under perfect conditions and also have sufficient power and resources to communicate securely with all nodes and outside the network. Before deployment, each sensor node is assigned a unique ID that is recorded in the base station. After deployment, sensor nodes self-organise into clusters by broadcasting their unique IDs and listening for IDs being broadcast by neighbouring nodes. Upon receiving a broadcast ID, each node adds this ID to its routing table. Nodes that share IDs with each other then form a cluster. Each cluster then elects one sensor node to act as cluster leader and all communication between different clusters must be routed through the respective cluster leader. Similarly, all communication between a node and the base station must also pass through the node's cluster leader.

Since the volume of communication routed through the cluster leader will be significantly larger than that of other sensor nodes in the network, this will increase the cluster leader's power consumption. However, a sensor node's energy supply is very limited so in order to enable consistent power consumption between all nodes in a cluster, the role of cluster leader changes periodically. This provides each node the opportunity of becoming cluster leader.

In this model, when a cluster leader cannot route data accrued by one of its sensor nodes directly to the base station, it may do so by inter-cluster communication and reaching the base station by routing the data via other cluster leaders.

Once the network is deployed, the base station builds a table containing the unique IDs of all the nodes in the network. After the self-organizing process has completed, the base station will then know the topology of the sensor network. Using this hierarchical topology, nodes will collect data, pass this to their respective cluster leader who will aggregate the packets and send them either directly to the base station or via one or more cluster leaders.

### 4.1.2 Inserting additional nodes into the network

Additional nodes may be inserted into the network at any time. Before a node is inserted, the base station records and stores its unique ID and will insert the node into a cluster having the least number of nodes. This will help minimise the event of a cluster monopolising bandwidth if it contains a greater number of nodes than other clusters who are communicating. The node will then self organise within its cluster.

## 4.2 Key management

Establishing secure key management in sensor networks is a difficult issue to solve. However, security techniques such as asymmetric cryptography that use keys are impractical due to the sensor node's resource constraints and the network's ad hoc environment where nodes are randomly joining and leaving. One common key management technique employed in wireless sensor networks is a key pre-distribution scheme where key information is embedded in sensor nodes before they are deployed. This is an energy efficient key management mechanism for resource constrained nodes [23].

The key management scheme in this architecture uses two keys similar to that proposed in [23]:

- $K_n$ (network key) – Generated by the base station, pre-deployed in each sensor node, and shared by the entire sensor network. Nodes use this key to encrypt the data and pass onto the next hop.
- $K_s$ (sensor key) – Generated by the base station, pre-deployed in each sensor node, and shared by the entire sensor network. The base station uses this key to decrypt and process the data and the cluster leader uses this key to decrypt the data and extract nonce values.

The base station uses $K_n$ to encrypt and forward data. When a sensor node receives the message, it decrypts it by using its own $K_s$.

A cluster leader amasses any messages received from nodes within its cluster and forwards them to the next level cluster leader or directly to the base station itself if it is one-hop away. If a cluster leader receives a data packet from a node within its cluster, it will first add its own unique ID and TimeStamp to the packet before forwarding it. All cluster leaders who are not one-hop away to the base station add their own ID to packets they receive from a sending cluster leader.

When the base station receives a packet, it checks the ID of the sending cluster leader. It authenticates the cluster leader who sent the packet and also the packet's integrity.

## 4.3 Identifying and isolating aberrant nodes

Sensor nodes that do not function as specified must be identified and isolated in order to continue the desired operation of the sensor network. An aberrant node may be the result of an attack or may act maliciously due to unexpected network behaviour. According to Hu et al. [24], an aberrant node is one that is not functioning as specified and may cease to function as expected for the following reasons [17]:

- It has exhausted its power source.
- It is damaged by an attacker.
- It is dependant upon an intermediate node and is being deliberately blocked because the intermediate node has been compromised.
- An intermediate node has been compromised and is corrupting the communication by modifying data before forwarding it.
- A node has been compromised and communicates fictitious information to the base station.

Therefore, the security of the sensor network can be maintained by identifying an aberrant node quickly and isolating it from the sensor network. ASLAN includes a protocol that is used to identify and isolate aberrant nodes. This is divided into two algorithms:

a. node-to-node
b. cluster leader-to-node

In order to describe the functionality of the protocol, it will be assumed that node A wishes to communicate with node B whom are both located within the same cluster. The protocol also assumes that a secure, end-to-end communications channel between node A and node B has been established. It is also assumed that an attacker is not capable of accessing the contents of packets received by the attacked node.

a. node-to-node

Node A will send data (i.e. packets) to node B. Before node A sends a packet, it generates a nonce, appends it to the packet and saves a copy of it in memory. A different nonce is generated for each packet. Due to memory constraints in sensor nodes and the possible large number of nonce values that may need to be generated, the nonce value will be a combination of a random, medium-size prime number and a time stamp. Node A also sends a copy of the nonce value associated with the packet to the cluster leader.

When node B receives a packet, it will be required to send an acknowledgement (ACK) back to node A within a specified time period. This ACK must contain the same nonce that it received. Node B also sends a copy of this nonce value to the cluster leader. Since the protocol assumes that an attacker cannot access the contents of received packets, the attacker cannot access the nonce and therefore append it to the ACK. Therefore, only a genuine node that has not been attacked is capable of sending an ACK containing the correct nonce back to the original sender of the packet.

When node A receives the ACK from node B, it will compare the nonce it receives with that it has saved in memory. If they are the same, this verifies that node B is not an aberrant node. Otherwise, if they are different or if no ACK has been received within the specified time period, it will assume node B is aberrant and node A then sends an alert to the cluster leader. Node A terminates all communication with node B and deletes the nonce value saved in memory.

Likewise, if node A receives an alert from the cluster leader indicating that node B is an aberrant node before receiving the ACK, it will immediately terminate communication with node B and delete all nonce values saved with respect to node B.

b. cluster leader-to-node

When node A sends packets to node B, node A will send the cluster leader a copy of each nonce value for each packet. When node B sends an ACK back to node A containing the nonce value, it also sends a copy of the nonce to the cluster leader. The cluster leader will compare the two nonce values. If they are the same, it will verify that node B has not been compromised and deletes the nonce values saved in memory it received from node A and node B that correspond to the packet.

If the two nonce values are different, the cluster leader issues an alert to all nodes in the cluster that node B is an aberrant node and should be ignored. This alert is also issued to cluster leaders in all other clusters who in turn notify the nodes in their respective cluster. The base station is also alerted and can take measures to isolate or remove node B from the sensor network. Similarly, if the cluster leader receives an alert from node A about node B, it carries out the same procedures.

In a situation when the cluster leader is the sender or receiver of data with another node, then it cannot act as the independent party to receive nonce values and compare them to check for differences. This means that its role of cluster leader must pass to another node that is not currently involved in direct communication. This ensures that the role of cluster leader does change periodically and is shared between all nodes in the cluster.

Table 2. Notation

| | |
|---|---|
| Timer | *timer* |
| Node A | *node_a* |
| Node B | *node_b* |
| Sent packet | *sent_packet#* |
| Received packet | *recd_packet#* |
| Cluster Leader | *cluster_leader* |
| Base Station | *base_st* |
| Authentic node | *auth_node* |
| Aberrant node | *abb_node* |

**Algorithm 1**. Node-to-node

```
1. node_a sends packet to node_b

2. node_a saves sent_packet# sent to node_b

3. node_a sends sent_packet# to cluster_leader

4. if timer not expire then
       node_b send ACK to node_a with recd_packet#
   end if

5. node_a receives ACK from node_b
   if recd_packet# B = sent_packet# A then

       delete sent_packet# in node_a
       node_b = auth_node AND communication continue
   end if

6. if recd_packet# B NOT = sent_packet# A then

       send ALERT to cluster_leader AND
       terminate communication with node_b
   end if
```

**Algorithm 2**. Cluster leader-to-node

```
1. if sent_packet# node_a = recd_packet# node_b then

       node_b = auth_node
       delete sent_packet# node_a AND
           recd_packet# node_b
   end if

2. if sent_packet# node_a NOT = recd_packet#
   node_b OR ALERT received from node_a then

       node_b = comp_node AND send ALERT to
       all nodes in cluster that node_b = abb_node
   end if

3. if sent_packet# node_a NOT = recd_packet#
   node_b then

       send ALERT to cluster_leader in all clusters
       AND base_st that node_b = abb_node
   end if
```

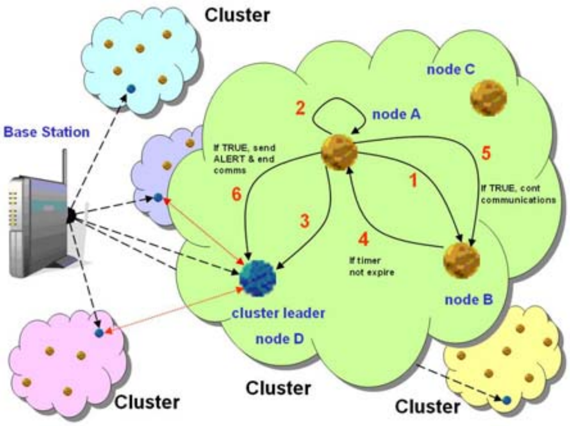


Figure 1. Node-to-node

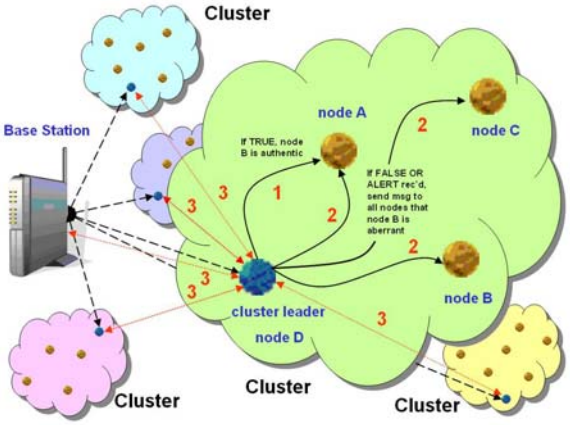

Cluster
Base Station
node A
node C
If TRUE, node B is authentic
If FALSE OR ALERT rec'd, send msg to all nodes that node B is aberrant
1
2
2
2
3
3
3
3
3
3
node B
cluster leader
node D
Cluster
Cluster
Cluster

Figure 2. Cluster leader-to-node

The algorithm presented takes into consideration the nodes and cluster leaders that are not the sender or intended recipient of data or are involved in aggregating nonce values. These nodes forward the data packets without applying any further cryptographic operation, thus further saving the nodes' processing power and memory.

### 4.4 Secure routing

ASLAN achieves secure data transmission by complimenting the energy efficient secure data transmission algorithm in [25] and adding extra security mechanisms by integrating the proposed algorithm to identify and isolate aberrant nodes in a cluster. The following two algorithms are proposed to achieve secure data communications from node to base station and vice versa.

<u>Sensor node algorithm</u>

1. Node A wishes to send data to another node. The recipient node may/may not exist within the same cluster.

2. Node A generates a nonce value and saves this value temporarily in memory

3. Node A encrypts the data it is sending using the encryption key $K_n$ (assigned at its deployment) and appends its ID, the current TimeStamp, and the nonce value to the encrypted data.

4. Node A sends the encrypted data packet to the cluster leader.

5. Cluster leader receives the encrypted data and makes a copy. It adds its own ID and TimeStamp to the original data packet and forwards this packet to the next higher-level cluster leader or directly to the base station itself if it is one-hop away.

6. Cluster leader decrypts the copy of the data packet it made using it's key $K_s$ (assigned at its deployment) and extracts the nonce value. It stores this temporarily in memory. It discards the copy of the data.

7. If the cluster leader receives incoming data destined for a node within its cluster, then make copy. The cluster leader decrypts the copy using it's $K_s$ and checks if it is an ACK (contains a nonce value). If the data is an ACK, then proceed to step #8, otherwise step #9.

8. Cluster leader compares the ACK nonce value with the original nonce value stored in memory for the original data packet sent. If equal, then delete the nonce value stored in memory and proceed to step #9. Otherwise, send alert to cluster leader that the sender node (node $ID_{1..n}$) to be considered compromised. Discard the original data packet, the copy and delete the nonce value stored in memory.

9. Node A receives data forwarded to it by its cluster leader. Decrypt the data using $K_s$ and check if it is an ACK (containing a nonce value). If not, proceed to step #10. Otherwise, compare the ACK with the original nonce value stored in memory. If same, continue as normal and delete the nonce value in memory. If different, assume sending node is compromised and send alert to cluster leader. Delete nonce value in memory.

10. Reply to sending node. Create a new data packet containing the ACK value (nonce value) extracted in step #9 and encrypt the data using it's encryption key $K_{n.}$ Send the ACK data packet. Process the received data accordingly. Return to step #1.

<u>Base station algorithm</u>

1. If a data packet has been received from a cluster leader that is needed to be forwarded, encrypt it using $K_n$.

2 If no data packet is needed to be forwarded, check if any incoming data from any cluster leaders. If not, return to step #1.

3. If there is incoming data to the base station, then decrypt the data using $K_s$. Extract the node ID and the TimeStamp.

4. If the data does not decrypt correctly, discard the packet and proceed to step #6.

5. Extract the message from the decrypted packet and process accordingly.

6. If necessary, send a request to the sensor node that transmitted the original packet to retransmit the data. Return to step #1.

## 5. Future work

ASLAN is a proposed architecture which assumes that the base station is always dependable and that the data stored in a sensor node's memory is secure. Therefore, the architecture is designed to address situations where an attacker will breach security and cause disruption to the sensor network such as interrupt, intercept, modify or fabricate data packets. Examples of these types of attack include attacks on information in transit [26] and blackhole/sinkhole attacks [27].

Consequently, we will implement and evaluate our architecture with a focus on the following criteria:

- Accuracy of intruder detection
    - Success rate
    - False-positive rate
    - False-negative rate
- Communication overhead (cost)
- Energy consumption

Results for these criteria will then be collated and evaluated.

## 6. Conclusion

Security is a primary concern in the design of a wireless sensor network. ASLAN needs to be as lightweight as possible in order to reduce the overhead burden placed on sensor nodes that have very limited resources. In this paper, we presented ASLAN: a lightweight architecture that aims to secure a wireless sensor network against deliberate and hostile attack. The proposed architecture consists of phases that involve a model for a self-organising network topology, a secure key management scheme and a secure routing system allowing data to traverse the network securely. The secure key management scheme is based on the pre-deployment of keys to sensor nodes.

ASLAN also incorporates a protocol for the identification and isolation of aberrant nodes. This protocol consists of two sections: node-to-node and cluster leader-to-node, both of whom work in tandem with each other. This protocol will identify any node that has become compromised and isolates it. ASLAN has also been designed so that nodes and cluster leaders, who are involved in forwarding data packets, do not apply any further cryptographic operations and thereby aid in enabling the architecture to be as lightweight as possible.

## 7. Acknowledgements

The work described in this paper was supported in part by a grant "Secure and Predictable Pervasive Computing" from Science Foundation Ireland (SFI).

## 8. References

[1] M. Collins, S. Dobson, and P. Nixon (2008), "A Secure Lightweight Architecture for Wireless Sensor Networks". Proceedings of the Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2008), Valencia, Spain

[2] J. M. Kahn, R. H. Katz, and K. S. J. Pister (1999), "Mobile networking for smart dust". In ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 99), Seattle, WA, USA.

[3] Matt Blaze, Joan Feigenbaum, and Angelos D. Keromytis "Keynote: Trust Management for Public-Key Infrastructures", SpringerLink, Lecture Notes in Computer Science, Security Protocols book, Volume 1550/1999, p625

[4] Al-Sakib Khan Pathan, Hyung-Woo Lee, and Choong Seon Hong (2006), "Security in Wireless Sensor Networks: Issues and Challenges", Proceedings of the 8th IEEE International Conference on Advanced Communication Technology (ICACT), Vol II, pp 1043 – 1048.

[5] D. Boyle, and T. Newe (2008), "Securing Wireless Sensor Networks: Security Architectures". Proceedings of the Journal of Networks (JNW), Vol. 3, No. 1, ISSN : 1796-2056

[6] Blackert, W.J., Gregg, D.M., Castner, A.K., Kyle, E.M., Hom, R.L., and Jokerst, R.M. (2003), "Analyzing interaction between distributed denial of service attacks and mitigation technologies", Proc. DARPA Information Survivability Conference and Exposition, Volume 1, pp. 26 – 36.

[7] Wang, B-T. and Schulzrinne, H. (2004), "An IP traceback mechanism for reflective DoS attacks", Canadian Conference on Electrical and Computer Engineering, Volume 2, 2-5 May 2004, pp. 901 – 904.

[8] Douceur, J. (2002), "The Sybil Attack", 1st International Workshop on Peer-to-Peer Systems.

[9] Newsome, J., Shi, E., Song, D, and Perrig, A (2004), "The sybil attack in sensor networks: analysis & defenses",

Proc. of the third international symposium on Information processing in sensor networks, ACM, 2004, pp. 259 – 268.

[10] Chris Karlof and David Wagner (2003), "Secure routing in wireless sensor networks: attacks and countermeasures", ScienceDirect, Ad Hoc Networks, Volume 1, Issues 2-3, pp. 293 – 315.

[11] Yih-Chun Hu, Adrian Perrig, and David B. Johnson (2002), "Wormhole detection in wireless ad hoc networks," Tech. Rep. TR01-384, Department of Computer Science, Rice University, June 2002.

[12] Pfleeger, C. P. and Pfleeger, S. L. (2003), "Security in Computing", 3rd edition, Prentice Hall 2003.

[13] Mike Chen, Weidong Cui, Victor Wen, and Alec Woo (2000), "Security and Deployment Issues in a Sensor Network", http://www.cs.berkeley.edu/wdc/classes/cs294-1-report.pdf

[14] Adrian Perrig, Robert Szewczyk, J. D. Tygar , Victor Wen, and David E. Culler (2001), "SPINS: Security Protocols for Sensor Networks", Proceedings of the Seventh Annual International Conference on Mobile Computing and Networks, MOBICOM 2001

[15] Adrian Perrig, Robert Szewczyk, J. D. Tygar , Victor Wen, and David E. Culler (2002), "SPINS: Security Protocols for Sensor Networks", Wireless Networks, Vol. 8, 521-534

[16] Tanveer A. Zia (2006), "An overview of Wireless Sensor Networks and their Security Issues", LNCS – Secure Data Management in Reactive Sensor Networks, SpringerLink, Volume 4332

[17] J. Undercoffer, S. Avancha, A. Joshi, and J. Pinkston (2002), "Security for Sensor Networks", CADIP Research Symposium

[18] L. Eschenauer and V. Gligor (2002), "A Key-management Scheme for Distributed Sensor Networks", Proceedings of the 9th ACM conference on Computer and Communication Security", Washington DC, USA

[19] H. Chan, A. Perrig, and D. Song (2003), "Random Key Predistribution Schemes for Sensor Networks". Proceedings of the IEEE Symposium on Security and Privacy, Oakland, California, USA

[20] W. Du, J. Deng, Y. S. Han, and P. K. Varshney (2003), "A Pairwise Key Pre-Distribution Scheme for Wireless Sensor Networks", ACM CCS

[21] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney (2004), "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge", IEEE InfoCom

[22] Sasikanth Avancha, Jeffery Undercoffer, Anupam Joshi, and John Pinkston (2003) "Secure Sensor Networks for Perimeter Protection," Computer Networks: The International Journal of Computer and Telecommunications Networking, 43(4), 421–435

[23] Tanveer Zia and Albert Zomaya (2006), "A Security Framework for Wireless Sensor Networks", Proceedings of the IEEE Sensors Applications Symposium, February, 2006

[24] F. Hu, J. Ziobro, J. Tillett, and N. Sharma, "Secure Wireless Sensor Networks: Problems and Solutions", Rochester Institute of Technology, Rochester, New York, USA

[25] H. Cam, S. Özdemir, D. Muthuavinashiappan, and P. Nair (2003), "Energy Efficient Security Protocol for Wireless Sensor Networks", IEEE

[26] Al-Sakib Khan Pathan, Hyung-Woo Lee, and Choong Seon Hong (2006), "Security in Wireless Sensor Networks: Issues and Challenges", Proceedings of the 8th IEEE International Conference on Advanced Communication Technology (ICACT), Vol II, pp 1043 – 1048

[27] Edith C.H. Ngai, Jiangchuan Liu, and Michael R. Lyu (2007), "On the Intruder Detection for Sinkhole Attack in Wireless Sensor Networks" Computer Communications, Vol 30, Issue 11 – 12, pp 2353 – 2364

**Michael Collins** (B.Sc., M.Sc.) is a part-time Ph.D candidate and member of the Systems Research Group in University College Dublin, Ireland. His research is under the joint supervision of Dr. Simon Dobson and Prof. Paddy Nixon and is focusing on security and authentication in wireless sensor networks. His research interests include ubiquitous computing, security, and I.T. education.

**Simon Dobson** (B.Sc., M.A., D.Phil) is a lecturer in the School of Computer Science and Informatics, University College Dublin, Dublin, Ireland. His research centres around adaptive pervasive computing and novel programming techniques. He has an extensive record of published work (including papers in CACM, TAAS, JPDC, EHCI and ECOOP) and primary authorship on grants worth over €3M (and further involvements grants worth over €28M) feeding around €1.5M directly into his own current research programme. He is National Director for the European Research Consortium for Informatics and Mathematics, a board member of the Autonomic Communication Forum (at which he chairs the semantics working group), and a member of the IBEC/ICT Ireland standing committee on academic/industrial research and development.

**Paddy Nixon** (B.Sc., M.A., Ph.D) is Professor of Distributed Systems in the School of Computer Science and Informatics, University College Dublin, Dublin, Ireland. He is the lead of the Systems Research Group located in the UCD Complex and Adaptive Systems Laboratory (CASL). Prof. Nixon is also a SFI (Science Foundation of Ireland) Professor of Ubiquitous Systems. His funding of EUR €2.5M from the SFI is helping to seed the formation of the SRG with an initial core activity in Secure and Predictable Pervasive Systems. He has published numerous articles and proceedings in the areas of pervasive computing, adaptive information, autonomic computing, middleware and applied formal methods.

# QoS for Wireless Mesh: MAC Layer Enhancements[*]

Mathilde Benveniste, Ph.D.
*En-aerion*
*benveniste@en-aerion.com*

## Abstract

*Wireless mesh networks present MAC design challenges beyond those of WLANs. Abundant hidden nodes increase the number of collisions. This, combined with the correlated access needed when forwarding a multi-hop flow, degrades QoS. MAC enhancements for meshes are presented in this paper that reduce latency for mesh traffic while promoting co-existence with nearby WLANs. Wider contention windows for backoff lower the risk of repeated hidden-node collisions, a spatial extension of the TXOP concept called 'express forwarding' clears multi-hop flows sooner, and a new mechanism called 'express retransmission' reduces collisions on retransmission. Simulation results show the potential benefit of the proposed enhancements. The issue of fairness is addressed, as well as preservation of QoS in nearby WLANs.*

## 1. Introduction

A wireless mesh network is a network that accommodates forwarding of packet traffic on a wireless medium over one or more hops. Enabling multiple-hop communication, which gives rise to a mesh network, extends the range of a wireless LAN (WLAN). A mesh may furnish wireless connections either to access points (APs) serving different WLANs, or simply to devices supporting peer-to-peer wireless communication. A gateway, the *portal*, facilitates communication of the users of the mesh with users on other networks. A wireless mesh shares many of the challenges encountered in mobile and *ad hoc* networks, also known as MANETs [2] – [4].

Wireless mesh is useful both in environments where wired network infrastructure is unavailable and where eventual connectivity to the available wired network is desirable. The first, commonly known as *ad hoc* mode meshes, are useful for the ability to be quickly deployed with low cost where there is no wired infrastructure. The second type of mesh, known as *infrastructure* mode meshes, help extend connectivity range without additional wiring. Examples of mesh usage include emergency early response, public Internet access, metropolitan hotspot coverage, and enterprise and campus wireless networks.

MAC design for wireless meshes must account for a variety of features, such as the number of physical channels used in the mesh. Low volume meshes linking devices through peer-to-peer single- or multi-hop wireless connections can perform well on a single channel. Meshes providing wireless backhaul for a collection of APs, however, would require greater channel capacity than any one of those APs. Multiple channels would be needed to backhaul traffic of multiple APs. Finally, different MAC protocols are needed when multiple channels are used in a mesh with a mix of multiple radios per device. The IEEE 802.11s Task Group is currently addressing the standardization of a wireless mesh MAC that will be compatible with the IEEE 802.11 WLAN MAC protocol [5].

### 1.1 QoS in Wireless Networks

QoS objectives can be pursued on different ISO layers. QoS metrics such as end-to-end latency can serve as the optimization criterion in routing. Routes may vary in time due to mobility and topology changes [6] - [8]. Routes can also be adapted to traffic-trend changes over time, but routes do not change on a per-packet basis. The excessive control load needed to change routes would defeat an attempt to use routing to resolve collisions. Cross-layer interactions and their implications for QoS have also been considered [9] - [11].

Since much of the latency experienced in a wireless network occurs in accessing the shared

[*] Some of this work was done while the author was affiliated with Avaya Labs – Research.

medium, MAC protocol design is important in meeting QoS requirements. Whether transporting packets for backhaul to/from APs or linking wireless devices over multiple hops, applications with limited latency tolerance should be delivered within the required delay bounds. In addition, MAC protocols must be compatible with existing wireless networks operating on the same RF spectrum. Interoperability implies fair behavior toward other users of the RF spectrum, and especially not destroying the QoS expected by such users. The underlying problem is that of accessing the wireless medium in a fair, efficient, and distributed manner.

Latency restrictions for QoS are meaningful end-to-end. International Telecommunications Union document G.114 recommends a limit for end-to-end delay of 150 milli-seconds for real-time voice [12]. After subtracting from this total delay budget 50 to 60 milli-seconds for encoding, packetization, decoding and jitter buffering delays, the delay allowed for a wireless mesh carrying real-time traffic will depend on other delays experienced outside the wireless mesh. If the wireless mesh stands alone it will have a greater delay budget than if it interfaces with other network infrastructure. Voice over IP packets traversing wired networks experience IP network delays of about 50 milliseconds, which include propagation, table lookup and queuing delays. The delay budget would typically leave voice traffic between 40 and 50 milliseconds for network access/egress. The mesh latency limit applies on a per flow basis. Hence, in a wireless mesh, the allowed delay restriction applies to the entire multi-hop path.

If one extrapolated from experience with WLANs, meeting the above latency limit would not appear difficult for any but the longest multi-hop flows. If latency for single-hop access was less than 10 milliseconds, a five-hop path could be completed within the allowed time limit. We find, however, that wireless meshes have novel collision behavior that imposes latency increases on both mesh and co-channel WLANs beyond what non-mesh experience suggests.

## 1.2 MAC protocol design for real-time traffic over mesh

A key contributor to latency in a wireless network is the contention occurring when accessing the shared medium. Hence, the design of the MAC protocol is an important consideration in meeting the requirements of real-time traffic. The mutual RF interference experienced at nodes sharing the same channel, which is added to ambient noise, can prevent correct decoding on the receiving node.

Several MAC protocols exist for both single-channel and multi-channel meshes. For single-channel meshes, the IEEE 802.11 distributed MAC protocol for WLANs, known as EDCA, [5], [13] is the MAC protocol most commonly used [4]. For meshes using multiple channels, access can be combined with channel assignment. In addition, if the number of transceivers on a node is smaller than the number of channels employed in the mesh, access can be combined with scheduling radio and channel use on different links [14] – [23]. Of the multi-channel protocols, some employ EDCA as the underlying MAC protocol and interoperate with the IEEE 802.11 MAC and some do not.

EDCA enables WLANs to meet QoS requirements through the TCMA (Tiered Contention Multiple Access) protocol for prioritized channel access [13], [24]. In the absence of low priority traffic, however, prioritized access does not offer any benefit. Consequently, EDCA results in comparable latencies with the basic CSMA/CA protocol [25], [26]. The following question thus arises:

Considering distributed MAC protocols that are compatible with WLANs operating on the same channel as the mesh, does the CSMA/CA MAC provide the best QoS performance for a wireless mesh, or can another MAC protocol perform better?

The single-channel mesh is of special concern, for a variety of reasons. Single-channel meshes are expected to gain acceptance rapidly once standardized, and through the flexibility they offer, will provide the technology toward which future WLANs will evolve. Though not appropriate for backhaul of multiple fully loaded WLAN APs, they can be used as a means of extending the range of an infrastructure wireless network and for data rate improvement. By replacing the WLAN AP with a mesh portal as the distribution network interface, wireless devices will be able to reach the wired network from a longer distance away, on multiple hops. Multi-hop transmission will also increase the realizable data rate. Devices situated on the edge of a WLAN’s coverage area are limited to transmit on a single hop at low data rates. With one or more devices situated in between, the edge device’s traffic would be forwarded on multiple yet shorter hops, which would be capable of higher rates.

Although channel assignment and radio scheduling problems do not arise in single-channel meshes, MAC design is more challenging. The short channel re-use distances encountered in single-channel meshes cause a prevalence of hidden nodes. The prevalence of hidden nodes increases collision rates and retransmissions, and leads to higher channel utilization per attempted transmission and ultimately to dropped frames. In the rest of the paper, we are concerned with meshes employing a single channel throughout the mesh and a single radio per mesh node. This paper is based on the author's presentation on this subject at MESH 2008 [1].

Before exploring how hidden nodes impact QoS performance of single-channel meshes, we describe in Section 2 the distributed MAC protocol for IEEE 802.11 WLANs and the remedy for hidden node collisions in WLANs. Section 2 describes how the mesh topology impacts the effectiveness of the 802.11 MAC protocol. A new MAC protocol and other remedies for removing the deleterious effects introduced by mesh topology are described in Section 4. In Section 5, we compare the performance of different MAC protocol options for static routing conditions. Section 6 contains conclusions.

## 2. The existing IEEE 802.11 MAC protocol

The IEEE 802.11 standard for WLANs employs a distributed MAC protocol, CSMA/CA. A combination of prioritized access and admission control offer satisfactory QoS in IEEE 802.11 WLANs. Prioritized access is achieved through service differentiation. Higher priority packets have a higher probability of accessing the channel before lower priority frames. Fairness among devices with one or multiple types of traffic is ensured through the use of different EDCA queues for different types of traffic, each queue contending independently [5], [13].

The CSMA/CA protocol has been designed to avoid collisions through carrier sensing, backoff, and handshake. A device transmits only when the channel is determined idle. Each device listens to the channel and, if busy, postpones transmission and enters into the 'backoff procedure'. This involves deferring transmission by a random time, determined by the backoff value drawn randomly. Backoff facilitates collision avoidance between multiple stations that would otherwise attempt to transmit immediately after completion of the current transmission. The backoff value expresses, in time slots, the cumulative time the channel must be idle before access may be attempted.

IEEE 802.11 WLANs use TCMA, an enhanced version of CSMA/CA, to prioritize access among different traffic types [24]. A station engaged in backoff countdown must wait while the channel is idle for time interval equal to DIFS before decrementing its backoff delay immediately following a busy period, or before attempting transmission. According to the TCMA protocol, variable lengths of this time interval, which is called Arbitration-Time Inter-Frame Space (AIFS), lead to varying degree of accessibility to the channel. A shorter AIFS will give a station an advantage in contending for channel access. Differentiation between different access categories is achieved by assigning a shorter AIFS to a higher priority access category.

Prioritized distributed channel access mechanisms like TCMA meet packet latency requirements when the WLAN is reasonably loaded. The challenge is to meet similar end-to-end QoS requirements with a distributed MAC protocol on a per flow basis for a reasonably loaded mesh.

## 3. Using the existing MAC in mesh

Prioritized access increases the probability of higher priority traffic transmitting before lower priority traffic. However, that alone is not sufficient to meet the latency restrictions for QoS. The end-to-end delay experienced in a mesh multi-hop path is not always a simple multiple of the delay experienced for a single hop in a non-mesh environment. A single hop flow in a mesh may experience a longer delay than non-mesh experience would suggest. The prevalence of hidden nodes and the interaction of contention-based access with multi-hop flows can impose latency increases on both single and multi-hop flows beyond what non-mesh experience suggests.

### 3.1 Hidden node collisions

Collisions in wireless mesh networks occur for two reasons. One type of collision is caused by simultaneous transmissions by two or more devices located sufficiently close that their signals result in signal to interference plus noise ratio (SINR) at the receiver that is too low for proper decoding. Typically such a collision occurs if the backoff delay of two or more such devices waiting to transmit expires simultaneously. Obviously, the higher the concentration of active devices in the vicinity of a transmitter-receiver pair, the higher the collision rate observed.

Another way collisions arise is from 'hidden nodes' [27]. A hidden node is one that cannot sense an ongoing transmission, but if it transmits, it can interfere with the decoding of such transmission at the receiver. An example of a hidden node is illustrated in Figure 1, where node F, which is outside the sensing range of node A, is a hidden node when node A transmits to node B. The sensing range of a transmitter refers to a range within which any node can sense the received signal, whose power level exceeds a sensing threshold. Collisions can result from hidden nodes as follows. If node B is within the interference range of node F and nodes A and F engage in overlapping transmissions, B will be unable to decode a transmission from A. This is known as a 'hidden node collision'.

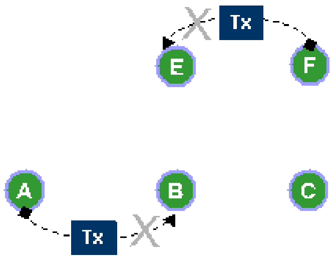


**Figure 1. Hidden node collision**

The IEEE 802.11 MAC protocol offers RTS/CTS and TXOPs as possible remedies for hidden node collisions [5]. RTS/CTS involves the use of a multiple-frame handshake between the transmitter and receiver, which comprises short control frames – namely, RTS (Request to Send) and CTS (Clear to Send) frames [25], [28]. The RTS, which is sent by the source of the pending transmission to head off a collision, includes the period of time for which the channel is reserved. The receiver returns a CTS control frame if the channel is clear to send. This frame notifies also the neighboring nodes of the channel reservation as it carries a field with the duration of the channel reservation.

The RTS/CTS handshake protects against hidden node collisions in two ways. If the sender of a frame cannot sense an ongoing transmission that its intended recipient hears, the CTS will not be sent; using the RTS will preempt a hidden node collision involving the frame. Once the frame transmission starts, any hidden nodes would refrain from transmission because they received the CTS, thus averting hidden node collisions.

A TXOP (Transmission Opportunity) also provides protection against hidden nodes. The frame initiating the TXOP carries a field indicating the TXOP duration, which is the period of time for which the channel is reserved. The receiver returns this information in the acknowledgement frame, which notifies the neighboring nodes of the channel reservation as it carries a field with the duration of the channel reservation.

It must be noted that neither RTS/CTS nor TXOPs reserve the channel for the transmission on the next hop of a multi-hop transmission. In the discussion that follows, we explain how this can be done through 'express forwarding', and how this capability can be used for the transmission of the RTS and for a TXOP, thus combining their respective benefits.

There is a tradeoff in using RTS/CTS. The penalties include the increased bandwidth taken by the control frames. Additionally, collisions are not entirely avoided. Both the RTS and CTS may be involved in collisions. The RTS may be involved in a regular collision or a hidden node collision, just like any other frame. The CTS may cause a hidden node collision to an ongoing transmission its sender cannot hear. This notwithstanding, the use of RTS/CTS pairs is advantageous if they avert collisions involving longer frames. A tradeoff exists, therefore, between the increased bandwidth taken by the control frames and the decrease in channel time lost to collisions.

Hidden node collisions are more prevalent in mesh networks than in WLANs. Hidden node collisions arise in WLANs, but not with the same frequency as in the mesh. In infrastructure WLANs -- that is, WLANs where stations communicate typically through the AP -- hidden node collisions occur only on uplink transmissions, as all devices can hear the AP. WLANs with overlapping coverage areas can avoid cross collisions by selecting different channels. Thus, co-channel WLANs could be separated by longer distances than possible for nodes of a single-channel mesh, avoiding cross collisions. Figure 2 illustrates hidden node collisions in WLANs. Simultaneous transmissions by nodes A and E to their serving AP at node B will fail, because A and E, although part of the same WLAN, cannot hear one another. On the other hand, simultaneous transmissions by nodes A and F (or by E and F) to their serving APs, at nodes B and D, respectively, will be received successfully if the two APs use different channels. If the nodes in Figure 2 represented a mesh, all operating on the same channel, simultaneous transmissions by nodes A and

F (or by E and F) to nodes B and D, respectively, would fail because of hidden node collisions.

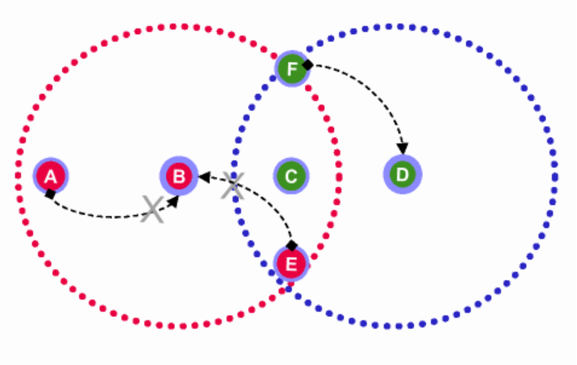


**Figure 2. Hidden node collisions in WLANs**

Hidden nodes are most prevalent in mesh networks used for range extension because the proportion of nodes that can hear each other is small. A node cannot typically decode the transmissions by neighbors of neighbor nodes. While long separation between communicating nodes gives rise to channel re-use potential across a mesh, the derived benefit disappears in a mesh using a single channel. Between a pair of potentially non-interfering nodes in a connected mesh, lies a third node that can cause interference to both pair members, operating on the same channel. This gives rise to hidden nodes and the potential of hidden node collisions. A grouping where all nodes can hear one another, i.e. a grouping without hidden nodes, will perform better because of the effectiveness of collision avoidance. Such a grouping, however, might be covered as well by a single WLAN.

Hidden nodes arise also when a single-channel mesh is located near a WLAN that uses the same channel. The mesh neighbors of a mesh node within sensing range of a WLAN device may be outside the sensing range of the same WLAN device and therefore become hidden nodes. Similarly, other WLAN devices would be hidden nodes for the mesh node closest to the WLAN.

The prevalence of hidden nodes increases collision rates and retransmissions, leading to higher channel utilization per attempted transmission and to dropped frames. Figure 1 illustrates common topologies in wireless meshes that cause repeated collisions and dropped frames. Nodes A and F cannot hear one another while node B and E can hear both A and F. The transmissions A and F to B and E, respectively, overlap in time. As a consequence, both B and E experience collisions. These collisions are likely to repeat on re-transmission because nodes A and F cannot hear each other. The backoff delay of each is decremented in time regardless of whether the other is transmitting, and transmission is likely to be attempted while the other is transmitting, simply because they cannot hear each other. Repeated collisions increase latency. If the retry limit is reached, their frames are dropped. With adjustable data rates, high dropped-frame rates lead to data rate reduction and low throughput.

## 3.2 Multi-hop flows

Multi-hop flows in a single-channel mesh may experience repeated hidden node collisions along several of their hops, causing the end-to-end delay to build up. In addition, their interaction with contention-based access can cause latency increases on other single and multi-hop flows beyond what non-mesh experience suggests. This novel behavior of meshes can impact nearby WLANs as well.

Longer delays can be caused by multi-hop flows because of special features of the contention-based access mechanism. When a transmission is involved in a collision, it is at a disadvantage relative to transmissions attempted for the first time. According to the IEEE 802.11 MAC protocol, a device attempting a failed transmission must draw a random backoff from a wider range – known as the contention window. A retransmit backoff delay would typically be longer than the backoff delay drawn by a forwarding device, immediately following the successful receipt of a frame of a multi-hop flow. Therefore, if a re-transmitting device is in the vicinity of a multi-hop flow, this device may have to wait for the completion of multiple hops of that flow before retransmission is possible because of its longer retry backoff.

Collisions are often caused by a multi-hop flow as it advances along its path. Transmissions near a multi-hop path are vulnerable. The acknowledgement of successful receipt of a frame on one hop may collide with a transmission further down near the path, which will likely have to wait for the entire multi-hop flow to complete. Figure 3 illustrates how a multi-hop flow may delay a transmission near its path. The acknowledgement from node C to node D causes a collision for the transmission to node B. Node A will probably have to wait for nodes D and E to forward the frame they receive before it can retransmit because of its longer backoff.

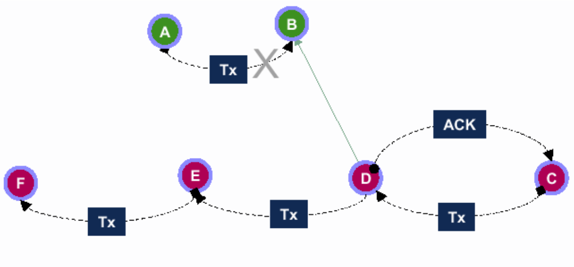


**Figure 3. Collision due to multi-hop flow**

Applications with short frame inter-arrival times (e.g. HDTV) risk going unstable if situated near multi-hop flows.  The sooner the multi-hop flow completes the sooner retransmission will succeed.

## 4. MAC remedies for wireless mesh

Three measures are proposed to improve the QoS performance of single-channel wireless meshes.  They are: (1) use of wider contention windows for transmission retry following a collision, (2) ‘express forwarding’ and (3) ‘express retransmission’.

### 4.1 Wide Retry Contention Window

By increasing the contention window on transmission retry, according to the first measure, the likelihood of averting a repeat collision increases for two nodes whose transmissions collided because they cannot hear each other. In this case, the backoff delay represents the clock time -- not the cumulative channel idle time – each such node will wait before transmitting, as the transmission of the other node is not heard.  Therefore, increasing the retry contention window increases the probability that the transmissions of the two nodes will not overlap in time.

This measure can be implemented simply when using the IEEE 802.11 MAC protocol by allowing the contention window size for backoff delay to increase more.  The default values for CWmax, the contention window size in the IEEE 802.11 Standard such that once it is reached, the window size is no longer doubled after a collision can be raised.

### 4.2 Express Forwarding

‘Express forwarding’ is an enhancement of the CSMA/CA protocol designed to reduce the latency experienced end-to-end by a multi-hop wireless mesh. Because it uses carrier sense functions and the collision avoidance backoff mechanism, it can interoperate with WLANs using the same channel. A high-level overview of express forwarding was first given in a presentation to the IEEE 802.11s task group [29].

According to express forwarding, multi-hop transmissions are expedited by reserving the channel via the transmitted frame on each leg of the multi-hop path for the next hop.  The notion of an Express Forwarding TXOP (EF-TXOP) thus arises, which is a time-space extension of the IEEE 802.11 TXOP. Transmit opportunities (TXOPs) enable a source to transmit multiple frames following a single successful channel access attempt, without having to contend for the channel. That is, a source transmits consecutive frames from the same access category without the need to contend (i.e. engage in backoff) more than once.   In an EF-TXOP, consecutive linked transmissions of a multi-hop flow are made without the need to contend more than once.  Reservation is done the same way as for TXOPs.  In a TXOP, the right to transmit contention-free following the initial successful channel access attempt remains with the source of the transmission.  With the EF-TXOP, the right to access the channel contention-free is handed over to the next node on a multi-hop path.

Reservation of the channel for an EF-TXOP is done through the virtual carrier sense mechanism used in IEEE 802.11 devices, as in the case of the TXOP, Virtual carrier sense is one of two mechanisms that enable a device to keep track of the activity level of the channel.  Physical carrier sense is based on the receiver detecting energy in the channel. Virtual carrier-sense relies on a timer, referred to as the network allocation vector (NAV), which indicates how long the medium will be busy.  A node is not allowed to transmit while its NAV timer is set. The NAV is set and updated based on the Duration field value contained in transmitted frames.  The response frame, which is the acknowledgement to a data frame or the CTS sent in response to an RTS frame, contains a Duration value derived from the value in the frame for which it is returned, adjusted for elapsed time.  Thus the duration field and the NAV timer provide a means for channel reservation.

The channel is reserved for a TXOP by setting the duration value of a frame long enough to cover at least one additional frame and its response frame, and by waiting a shorter time between transmissions than any other source contending for the channel. The Duration field of the response frame thus indicates the length of the following frame in the TXOP or the remaining TXOP duration. Because all but one frame in a TXOP is transmitted without contention, TXOPs help reduce the frequency of collisions. This increases channel use efficiency.

When a frame is express forwarded, the channel is reserved by extending the Duration field value of the frame long enough to silence all neighboring nodes and give the receiving node the opportunity to seize the channel and forward the frame. As illustrated in example of a three-hop flow in Figure 4, the NAV timer at neighboring nodes is set according to the Duration field value on a frame that is to be express-forwarded on the next hop. The Duration field value is longer than the time period the channel is occupied by the transmission and acknowledgment of the frame for the first two of the three hops of the path illustrated. This way, following the contention for the transmission on the first hop, an express-forwarded frame is transmitted quickly on the second and the third hop without contention, causing the multi-hop end-to-end delay to decrease. As in the case of a TXOP, EF-TXOPs help reduce collisions and thus increase channel use efficiency. As in the case of TXOPs, a limit can be imposed on the maximum length of an EF-TXOP, in order to avoid excessive delay jitter for non-express-forwarded traffic.

The time interval added to the duration field to reserve the channel for express forwarding should be one time slot plus the shortest time necessary to ensure that IP processing of the transmitted frame is complete at the receiving node. The additional reservation time gives the forwarding node the opportunity to seize the channel before any of its neighbors, as their NAV is set according to the received frame duration field value. If processing of an incoming frame commences as soon as it is received, and in parallel with the acknowledgement, the time increment added to the duration field is the time by which the processing time exceeds the time it takes to send an acknowledgement, if any, plus one time slot. The duration field value of an express-forwarded frame is not extended on the last hop of a multi-hop transmission.

Express forwarding can be used for the transmission of the RTS and for a TXOP, thus combining their respective benefits.

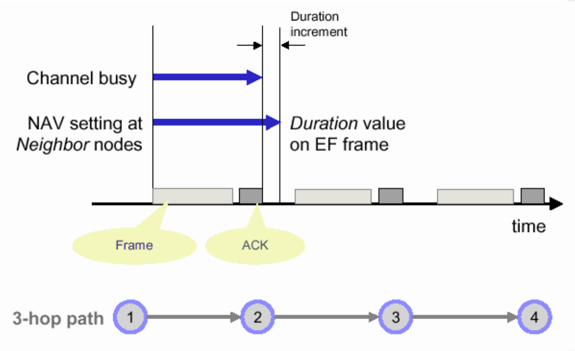


**Figure 4. Express Forwarding reservation**

### 4.2.1. Combining TXOPs and EF-TXOPs

EF-TXOPs can be combined with TXOPs in several ways. An express-forwarded frame can be transmitted along a hop as part of a TXOP. In order to enable the receiving node to seize the channel without contention for the next hop, the channel must be reserved beyond the end of the TXOPs transmission and acknowledgement. This can be achieved through the duration field of any of the frames in the TXOP. It suffices to extend the duration field of the last frame transmitted in the TXOP. If multiple express-forwarded frames are part of a TXOP, they can all be express-forwarded by the receiving node only if they are all going on the same link next. If the express-forwarded frames of a received TXOP request different next-hop destinations, the receiving node will have to select one mesh neighbor for its upcoming EF-TXOP. It may have to initiate other EF-TXOP(s) for the remaining packets that requested express forwarding.

When the received express-forwarded frame must be forwarded, if other frames queued at the receiving node can be sent in the same TXOP (that is, meets existing TXOP restrictions), the entire TXOP may go contention free. Its transmission may start immediately after the receiving node sends the last acknowledgement and following the appropriate AIFS idle period, even though the backoff delay of the frames included in this TXOP may not have expired. Transmission may thus start before the received express-forwarded frame is fully processed.

Embedding TXOPs within EF-TXOPs increases the efficiency of channel utilization. Express forwarding requires all nodes in the vicinity of the source to wait for the received frame to be processed at the IP layer and returned to the MAC layer for forwarding. This may cause the channel to remain unused if the time required for processing is longer than the time for transmitting the acknowledgment, assuming processing and acknowledgment is done in parallel. Transmitting more frames in the same TXOP following an express-forwarded frame allows the channel to be used while the express-forwarded frame is processed at the receiving node. Transmitting frames that are queued at the receiving node ahead of a received express-forwarded frame, in the same TXOP, allows the channel to be used while the received express-forwarded frame is processed for forwarding. Using either approach to place express-forwarded frames in a TXOP can prevent the channel from sitting unused.

#### 4.2.2. RTS/CTS with Express Forwarding

The RTS/CTS handshake is unlikely to benefit performance of wireless networks operating on fast channels, like IEEE 802.11a/g/n, for the reasons given earlier. For slower channels, the handshake can improve performance. RTS/CTS helps in a different way than express forwarding. The two mechanisms complement each other and can be used together.

Express forwarding can be used to send an RTS along each of the legs of a multi-hop path. The duration field of the RTS will reserve the channel not only for the protected transmission by the source of the RTS, but also for the next RTS transmitted by the forwarding node. This way RTS/CTS can reduce the penalty from forward hidden node collisions, while express forwarding will expedite the multi-hop flow and reduce the contention experienced by the RTS along the multi-hop path.

### 4.3. Express Retransmission

The retransmission of an express-forwarded frame that has been involved in a collision can also be expedited. Retransmission of a failed transmission typically involves contending with a backoff delay drawn from a wider contention window than the initial transmission attempt. An expedited retransmission, referred to as ‘express-retransmission’, can be sent contention free if the source retransmits as soon as the acknowledgment timer expires. If collision is experienced for an express-retransmitted frame, further attempts to transmit this frame will involve backoff from a widened contention window.

Express retransmission helps shorten the end-to-end latency of a multi-hop flow. An express-retransmitted frame will not collide with transmissions from neighbors as they have their NAV still set according to the duration field of the express-forwarded frame. If the collision that prompted the retransmission was due to a hidden node, the collision is less likely to repeat than in the case where both re-transmissions are attempted with backoff. It is less likely for the two retransmissions to overlap in time since express retransmission occurs without backoff, while other retransmissions must use a long backoff delay. An exception occurs if the hidden node collision involves another express-forwarded frame. Collision is likely then on the first retransmission attempt, but less likely on the subsequent attempt, since the backoff procedure is invoked with contention windows widened by a factor of four.

## 4 Performance evaluation

The performance benefits of express forwarding and express retransmission have been demonstrated in several studies for a range of scenarios [30], [31].

### 5.1. Description of study

The objective of these studies was to compare the QoS performance of a lightly loaded mesh, co-located with WLANs using the same channel for various channel access scenarios. We present here results from one of the studies, which deals with three scenarios, as described in Table 1. In the first scenario, all traffic accesses the channel through the IEEE 802.11 EDCA mechanism. Single-hop flows use EDCA for all scenarios. In the second scenario, express forwarding is employed for the multi-hop flows. In the third scenario, the multi-hop flows use express forwarding and express retransmission.

The network configuration consists of three WLANs and a wireless mesh, all operating on the same channel. The network traffic consists of constant flows between specified end points. The traffic flows simulated are three multi-hop flows, with three hops each, and a collection of single-hop flows. The multi-hop flows, which are part of the mesh, carry VoIP calls outside the mesh through a gateway device, the mesh portal. The single-hop flows belong either to the WLANs or to the mesh. The traffic of these flows is VoIP, low-resolution video, or high-resolution video, as indicated in Figure 5. The IP phones generate bi-directional streams communicating either with mesh peers or with the outside world through Node 0, which is the mesh portal. There was no node mobility; hence, static routing is employed. Table 2 presents the key traffic and MAC parameters.

Table 1. Scenario description

| Scenario | Description |
|---|---|
| 1. EF Disabled | Express Forwarding disabled |
| 2. EF Enabled | Express Forwarding enabled for multi-hop flows |
| 3. EF-ERTX Enabled | Express Forwarding & Express Retransmission enabled for multi-hop flows |

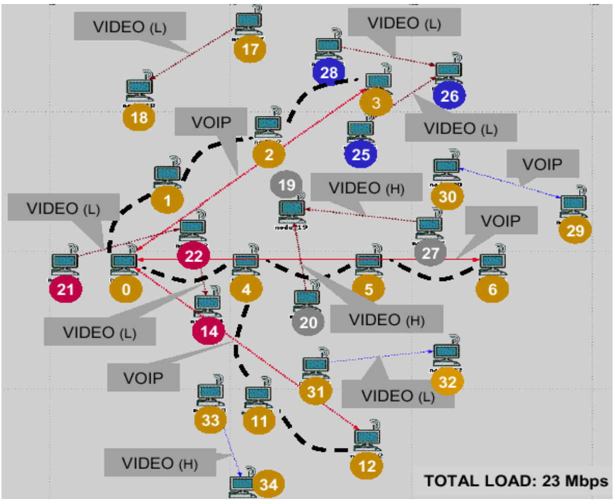


Figure 5. Network layout

Table 2. Key traffic and MAC parameters

| Traffic Type | Payload (bytes) | Frame Spacing (ms) | CWmin* | CWmax** WLAN/ Mesh |
|---|---|---|---|---|
| VoIP call | 200 | 20 | 7 | 15/ 1023 |
| Low-resolution Video | 1464 | 8 | 15 | 31/ 1023 |
| High-resolution Video | 1464 | 2.83 | 15 | 31/ 1023 |

*CWmin+1 is the contention window size used to draw a backoff delay when a transmission is first attempted

**CWmax+1 is the maximum size the contention window may assume when retransmission is attempted following a collision

All nodes were equipped with a single 802.11a radio. The channel was assumed to be noise free. Application data traffic was transmitted at 54 Mbps and acknowledgments at 24 Mbps. A 50 μsec IP processing delay was assumed at each node, typical delay for for processors in devices now being implemented. Processing of a frame starts as soon as it is received and in parallel with the transmission of an acknowledgement.

Simulations were conducted by using the OPNET Modeler modeling platform [32]. Statistics were computed over a simulation time of two minutes, starting when steady state was reached. Repeated experiments, obtained by varying the starting time of the flows randomly, showed negligible change in the measured statistics.

## 5.2. Results

Table 3 presents the mean end-to-end delays for all the flows under the three scenarios described in Section 5.1. The table indicates the network to which each flow belongs and whether it is a multi-hop flow – marked as (M) – or a single-hop flow – marked as (S). Figures 6 and 7 present, respectively, the normalized number of retransmissions and dropped frames by transmitting node. Normalization was done by dividing by the number of frames for which a transmission attempt was made at a given node.

Of the three multi-hop flows, only one – the call to Node 3 – meets the latency requirements for QoS when EDCA is the access mechanism. The other two multi-hop flows experience excessive delays and retransmissions. On some nodes, the average number of attempts needed exceeds two per frame. Retransmissions cause frames to be dropped; as many as 4 per cent of the frames are dropped at Node 11.

Table 3. Mean end-to-end delay (msec)

| Scenario | | EF Disabled | EF Enabled | EF-ERTX Enabled |
|---|---|---|---|---|
| Flow | Network | | | |
| Node 0 – Node 3 (M) | Mesh | 22 | 5 | 2 |
| Node 3 – Node 0 (M) | Mesh | 19 | 3 | 2 |
| Node 0 – Node 6 (M) | Mesh | 2,698 | 8 | 3 |
| Node 6 – Node 0 (M) | Mesh | 2,562 | 4 | 3 |

| | | | | |
|---|---|---|---|---|
| Node 0 – Node 12 (M) | Mesh | 3,583 | 17 | 6 |
| Node 12 – Node 0 (M) | Mesh | 3,448 | 16 | 7 |
| Node 17 – node18 (S) | Mesh | 12 | 4 | 3 |
| Node 29 – Node 30 (S) | Mesh | 9 | 3 | 3 |
| Node 30 – Node 29 (S) | Mesh | 4 | 3 | 2 |
| Node 31 – Node 32 (S) | Mesh | 8 | 4 | 3 |
| Node 33 – Node 34 (S) | Mesh | 28 | 14 | 7 |
| Node 20 – Node 19 (S) | WLAN 1 | 6 | 4 | 4 |
| Node 27 – Node 19 (S) | WLAN 1 | 8 | 5 | 5 |
| Node 21 – Node 22 (S) | WLAN 2 | 4 | 3 | 3 |
| Node 22 – Node 14 (S) | WLAN 2 | 3 | 2 | 2 |
| Node 25 – Node 26 (S) | WLAN 3 | 3 | 2 | 2 |
| Node 28 – Node 26 (S) | WLAN 3 | 3 | 2 | 2 |

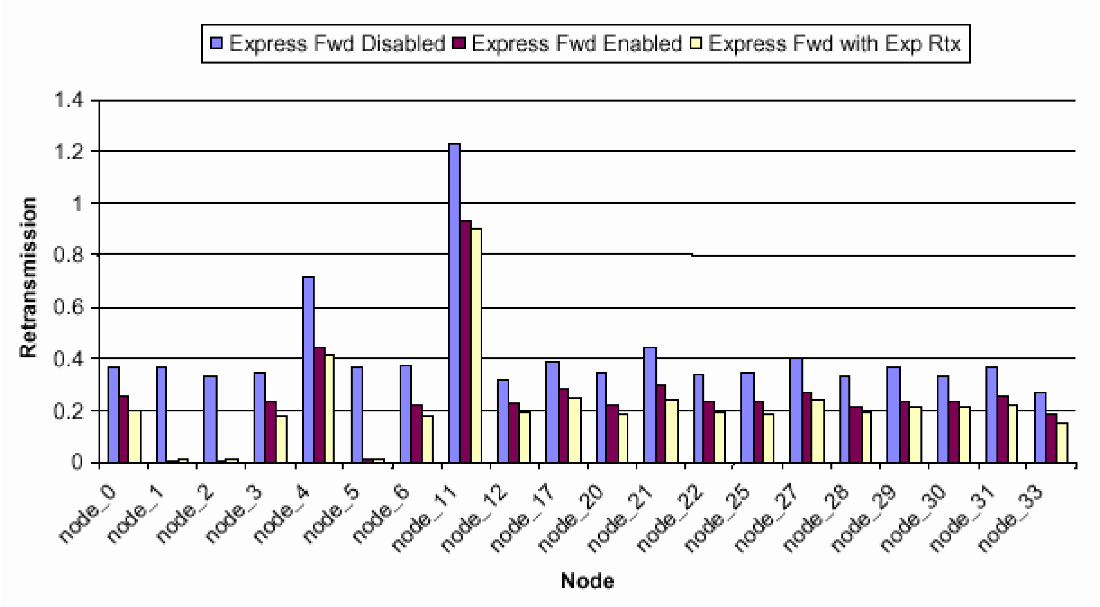


**Figure 6. Normalized retransmissions by node**

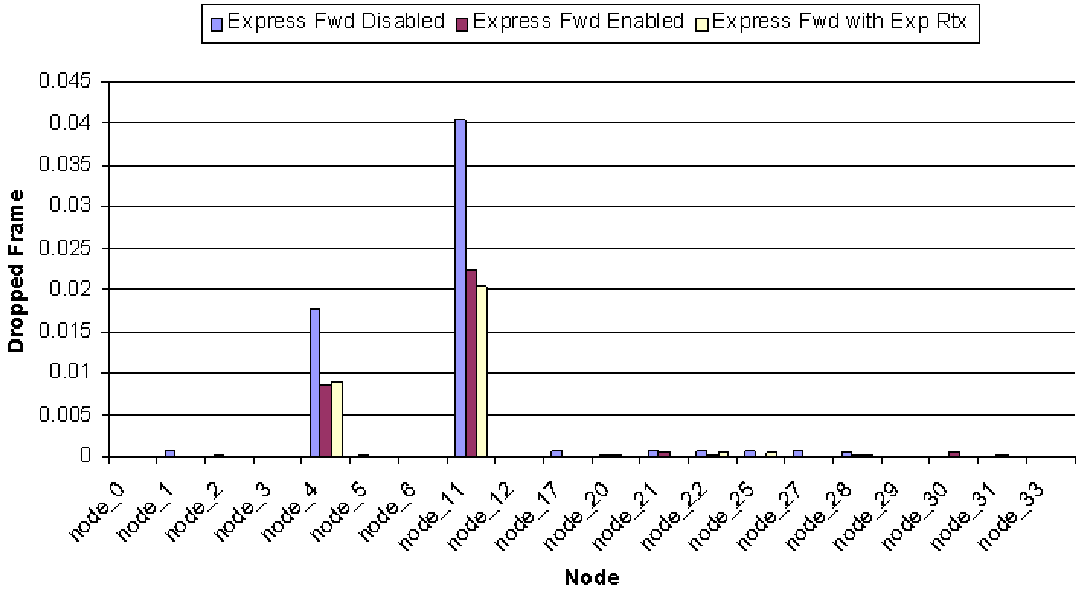


**Figure 7. Normalized dropped frames by node**

When express forwarding is applied to the multi-hop flows, the latency on all flows is reduced, whether they are express-forwarded or not. The latency reduction is greater for the flows that are express forwarded, but the other flows benefit as well. The number of retransmissions declines and the number of dropped frames is halved. All calls can meet QoS requirements with express forwarding. Express retransmission, combined with express forwarding, further improves MAC performance.

These results, as well as the other performance studies cited here, suggest that when packets are transmitted on a reserved channel, rather than contend for the channel on every leg of a multi-hop path, total contention is reduced considerably. As a consequence, both multi-hop and single-hop flows benefit from use of express forwarding for the multi-hop flows.

## 6. Summary and Conclusions

This paper deals with meshes using a single channel for all mesh nodes, and a single radio per mesh node. It describes a novel MAC protocol for mesh, called Express Forwarding, which represents an enhancement of the CSMA/CA protocol. Express Forwarding can be further enhanced through Express Retransmission. Express Forwarding can coexist with WLANS using the standard IEEE 802.11 MAC protocols to access the same channel as the mesh.

The performance of the new protocol was examined for a single channel mesh that is co-channel with several nearby WLANs. The combined traffic load was similar to that seen in a WLAN, and the multi-hop paths were of moderate length. It was observed that express forwarding was able to deliver delay performance that meets the QoS requirements for real-time applications. The standard IEEE 802.11 EDCA access mechanism could not meet these requirements.

Simulations confirmed that both types of frames (express-forwarded frames and non-express forwarded frames) enjoy shorter latencies when express forwarding is used for multi-hop transmissions. Paradoxical as this may seem, giving preferential treatment with express forwarding to nodes forwarding multi-hop traffic over nodes that transmit traffic for a single hop, has helped both types of transmissions. This is because, as with a TXOP, the EF-TXOP reduces contention on the channel and thus decreases the collision probability. Fewer collisions imply shorter latencies for all traffic. As in the case of TXOPs (where single-frame latencies may increase as a result of TXOP use), there may be some non-express forwarded traffic whose short delays will increase somewhat. According to our simulations, such increases are small and the resulting single-hop latencies are far shorter than the multi-hop latencies. As an added precaution, however, one can impose a limit on the maximum length of an EF-TXOP, very much the way we limited the maximum length of a TXOP.

The simulation studies involved VoIP and video traffic only, for the transmission of which the channel is accessed with the same AIFS. No Best Effort (lower priority) traffic was included. Had lower priority traffic been included, EDCA would have prioritized access accordingly. Express forwarding and prioritized access are orthogonal mechanisms that can be used together.

Express forwarding is a fair MAC protocol. When analyzing fairness in channel access on a per-node basis, express forwarding gives preferential treatment to nodes forwarding multi-hop traffic over nodes that transmit traffic for a single hop. Since the user's experience is tied to the end-to-end latency, however, fairness should be considered on a per-flow basis. Express forwarding is fairer than EDCA as it helps reduce multi-hop flow latencies and prevents single hop flows from experiencing longer delays than multi-hop ones. Regardless of the criterion used to establish fairness, however, it is important to note that the traffic disadvantaged with express forwarding – namely, the single-hop traffic – enjoys better performance when express forwarding is employed than when it is not. In general, all traffic enjoys better QoS performance with express forwarding than with EDCA.

Express forwarding can be extended to apply to multi-channel meshes. The benefit derived from it will depend on the MAC protocol used for channel assignment and scheduling radio use. This would be the subject of future investigation.

## References

[1] M. Benveniste, "A Distributed QoS MAC Protocol for Wireless Mesh", Sensor Technologies and Applications, 2008. SENSORCOMM '08. Second International Conference, Aug. 2008, pp. 788 – 795.

[2] C. Perkins, editor. *Ad Hoc Networking*. Addison-Wesley, 2001.

[3] I.A.Akyildiz and X. Wang, "A survey of wireless mesh networks", *IEEE Radio Communications*, September 2005.

[4] H. Zhai, J. Wang, X. Chen and Y. Fang, "Medium access control in mobile ad hoc networks: challenges and solutions", Wireless Communications and Mobile Computing, Vol 6, 2006, pp. 151 –170.

[5] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical (PHY) Layer Specifications, ANSI/IEEE Std 802.11, 2007 Edition

[6] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing", in *Proceedings of the* 2nd *IEEE Workshop on Mobile Computing Systems and Applications*, (New Orleans, LA), February 1999, pp. 90-100.

[7] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers", in *ACM SIGCOMM' 94 Conference on Communications Architectures, Protocols and Applications*, 1994, pp. 234.244.

[8] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks", in *Mobile Computing* (Imielinski and Korth, eds.), vol. 353, Kluwer Academic Publishers, 1996.

[9] S. Toumpis and A. J. Goldsmith, "Performance, Optimization, and Cross-Layer Design of Media Access Protocols for Wireless Ad Hoc Networks," in Proc. IEEE International Conference on Communications, May 2003, pp. 2234-2240.

[10] J. Chen and S.Hsia, "Joint Cross-Layer Design for Wireless QoS Video Delivery," in Proc. International Conference on Multimedia and Expo, July 2003, pp. 197-200.

[11] C. L. Gwee, Y. Qin and W. K. G Seah, "Bundled Virtual Circuit: A Proposed Cross Layer Design of Routing and Scheduling for QoS Services in MANETs" VTC Spring 2006, pp. 1283-1287.

[12] *Recommendation ITU-T G.114, One-Way Transmission Time,* Int'l Telecommunication Union, Geneva, 1996.

[13] M. Benveniste, "WLAN QoS", in *Emerging Technologies in Wireless LANs* (B. Bing, ed.), Cambridge Univ. Press, 2008.

[14] A. Nasipuri, and S. Das, "A Multichannel CSMA MAC Protocol for Mobile Multihop Networks." In Proc. of IEEE WCNC '99.

[15] S-L Wu, C-Y Lin, Y-C Tseng, and J-P Sheu, "A new multi-channel MAC protocol with on-demand channel assignment for mobile ad hoc networks", International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN) 2000, pp. 232– 237.

[16] Y-C Tseng, S-L Wu, C-Y Lin, and J-P Sheu. A multi-channel mac protocol with power control for multi-hop mobile ad hoc networks. In *Proceedings of 21st*

*International Conference on Distributed Computing Systems Workshops*, April 2001, pp. 419–424.

[17] A. Tzamaloukas, and J. J. Garcia-Luna-Aceves; .A Receiver-Initiated Collision-Avoidance Protocol for Multi-channel Networks.; Infocom '01.

[18] L. Bao, J. and J. Garcia-Luna-Aceves, “Hybrid channel access scheduling in ad hoc networks”, in Proc. IEEE ICNP’02, November 2002.

[19] N. Choi, Y. Seok , and Y. Choi, “Multi-channel MAC protocol for mobile ad hoc networks”, in Proc. IEEE VTC 2003-Fall, October 2003.

[20] K. Liu, T. Wong, J. Li, L.Bu, and J. Han, “A reservation-based multiple access protocol with collision avoidance for wireless multihop ad hoc networks” in Proc. IEEE ICC’03, May 2003.

[21] T. You, C. Yeh, and H. Hassanein, “A new class of collision prevention MAC protocols for wireless ad hoc networks”, in Proc. IEEE ICC’03, May 2003.

[22] Y. Liu, and E. Knightly, “Opportunistic Fair Scheduling over Multiple Wireless Channels”, in Proc. IEEE INFOCOM '03.

[23] J. So, and N. Vaidya, “Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver”, in Proc. MobiHOC 2004.

[24] M. Benveniste, “Tiered Contention Multiple Access (TCMA), a QoS-Based Distributed MAC Protocol”, in Proc. PIMRC, Lisboa, Portugal, September 2002.

[25] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical (PHY) Layer Specifications, ANSI/IEEE Std 802.11, 1999 Edition.

[26] L. Kleinrock and F. A. Tobagi. Packet switching in radio channels: Part–I - carrier sense multiple access modes and their throughput-delay characteristics. *IEEE Transactions in Communications*, COM-23(12), 1975, pp.1400–1416.

[27] F. A. Tobagi and L. Kleinrock. Packet switching in radio channels: Part–II - the hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *IEEE Transactions in Communications*, COM- 23(12), 1975, pp.1417–1433.

[28] P. Karn. MACA: A new channel access method for packet radio. In *Proceedings of ARRL/CRRL Amateur Radio* 9th *Computer Networking Conference*, 1990.

[29] M. Benveniste, “‘Express’ Forwarding for Single-Channel Wireless Mesh”, IEEE Doc 802.11-07-2452r2.

[30] M. Benveniste and K. Sinkar, “Performance Evaluation of ‘Express Forwarding’ for a Single-Channel Mesh”, IEEE Doc 802.11-07-2454r1.

[31] M. Benveniste and K. Sinkar, “More on Performance Evaluation of 'Express Forwarding' for Mesh”, IEEE Doc 802.11-08-0142r0.

[32] OPNET Modeler, http://opnet.com/solutions, May 2008.

# GTS Attack: An IEEE 802.15.4 MAC Layer Attack in Wireless Sensor Networks

Radosveta Sokullu
Dept. of Electrical and Electronics Eng.
Ege University
Izmir, Turkey
radosveta.sokullu@ege.edu.tr

Ilker Korkmaz
Dept. of Computer Eng.
Izmir University of Economics
Izmir, Turkey
ilker.korkmaz@ieu.edu.tr

Orhan Dagdeviren
Dept. of Computer Eng.
Izmir Institute of Technology
Izmir, Turkey
orhandagdeviren@iyte.edu.tr

***Abstract*—In the last several years IEEE 802.15.4 has been accepted as the major MAC layer protocol for wireless sensor networks (WSNs). It has attracted the interest of the research community involved in security issues because the increased range of application scenarios brings out new possibilities for misuse and taking improper advantage of sensor nodes and their operation. As these nodes are very resource restrained such possible attacks and their early detection must be carefully considered. This paper surveys the known attacks on wireless sensor networks, identifies and investigates a new attack, Guaranteed Time Slot (GTS) attack, taking as a basis the IEEE 802.15.4 MAC protocol for WSN. The GTS Attack is simulated with different scenarios using ns-2 and the results are evaluated both from the point of view of the attacked and the attacker.**

*Keywords*—*IEEE 802.15.4 MAC*; *wireless sensor network attacks*; *Guaranteed Time Slot*; *GTS attack*

## I. Introduction

Through the developments on micro electro-mechanical systems (MEMS) to be used as sensor devices [2], many ad-hoc network researchers have been focusing on Wireless Sensor Networks (WSNs). WSNs have many potential applications [3]–[7]. In the ubiquitous environment, WSNs enhanced with actuator capabilities can materialize the interface between people and the environment by establishing a context for a great variety of applications ranging from environmental monitoring to assisted living and emergency measures. In many of these scenarios, WSNs are of interest to adversaries and are easily prone to attacks as they are usually deployed in open and unrestricted environments. In many cases single nodes might be unattended and can be even physically destroyed or reprogrammed to work in a way different than their usual operations.

An attack on a WSN in general is defined as a defective action on the efficient operation of the whole system or a malicious invasion on a specific part of the network [11]. The attacker, as Wood et. al. [8] adapted from the National Information Systems Security Glossary [9], is mainly the originator of an attack and is used synonymously with the term adversary. The attacker can be an adversary within the network that attacks with the aim of damaging some nodes or gaining more selfish benefits on the provided services than the other legitimate users of the WSN. On the other hand the attacker may exploit protocol weaknesses to obtain network resources to his own benefit by depriving others or may simply try to cause disrupt in the operation of the network. The basic feature of attacks and misbehavior strategies is that they are entirely unpredictable [12]. Early definition and investigation of possible attacks and misbehavior patterns can provide valuable insight into reliable and timely detection which is a main prerequisite for ensuring proper operation and minimization of performance losses in WSNs. These issues motivated us to research on WSN attacks. Our goals are to survey the important WSN attacks categorized according to their target layers and to identify possible new attack types.

This paper extends the work in [1] and dissects the Guaranteed Time Slot (GTS) attack. The sequence of communication for realizing a GTS attack is presented, four different possible attack scenarios are defined and their ns-2 implementation results are presented and evaluated. From here on the paper is organized as follows: Section II covers the related work on attacks in WSN and their definitions, Section III discusses the IEEE 802.15.4 MAC layer security issues and Section IV identifies the new attack and presents the evaluation from the point of view of the attacker and the attacked taking into consideration both incurred damage and related energy consumption. Finally Section V concludes the paper.

## II. Related work

The known attacks in IEEE 802.15.4 WSNs can be classified into different categories according to different taxonomical representations. In this section the attacks for wireless sensor networks are categorized with regards to the different OSI layers whose operation and functions are attacked, destroyed or damaged. Chan et. al. [13] made this categorization mainly based on physical layer attacks, MAC layer attacks, and routing layer attacks; in addition to this, Raymond et. al. [14] has surveyed the denial-of-service attacks based on all protocol layers including transport and application layers.

### A. Physical Layer Attacks

Physical layer attacks cover mainly the *radio jamming* or *signal jamming* modifications aiming to corrupt the communication within the channel due to frequency interferences. If jamming is carried out by emitting just signals instead of sending packets, it is called radio jamming at the physical layer.

Another physical layer attack is *node tampering* [8], [14], [15]. An attacker, who has a physical direct access to the nodes, may tamper with the nodes. In this way, the attacker can interrogate a node's memory, can capture private information including the cryptographic data, can compromise the node's function, or can totally destruct the hardware [8], [14], [15].

Regarding the physical security concerns including physical accesses to sensor nodes or other network resources, wireless sensor networks are very vulnerable. Since sensor nodes are generally distributed in a wide area or are used in great numbers to realize a fault tolerant application, destructive physical accessibility to some single sensors, due to its perceivability, is not considered as very harmful for the whole network operation.

### *B. MAC Layer Attacks*

MAC layer attacks have attracted a lot of interest and there are a number of studies in this respect [11], [12], [17], [29]. IEEE 802.15.4 MAC layer attacks target the data link layer specifications to achieve mainly denial of service (DoS). Attackers generally aim to disrupt the specified IEEE 802.15.4 procedures for channel use or to consume the channel resources unfairly through modifying the IEEE 802.15.4 protocol definitions in a selfish and malicious manner. In the following we present a brief description of some IEEE 802.15.4 MAC layer attack types.

Jamming is basically constructing radio interference to cause a DoS on transmitting or receiving nodes. Xu et. al. [18] classified the jammers as constant, deceptive, random, and reactive according to their radio jamming strategies. *Link layer jamming* is fundamentally creating collision at the link layer by jamming packets rather than signals. An intelligent jammer that knows the link layer protocol logics intentionally misinterprets the channel use rules to deprive the legitimate users from gaining access to the medium. Rather than a blind jammer that emits signals or useless packets randomly without knowing the protocol logics, an intelligent jammer, from the point of the energy usage, aims to attack at specific times to preserve its energy [19]. *Back-off manipulation* is defined as selfishly and constantly choosing a small back-off interval in IEEE 802.11 Distributed Coordination Function (DCF) rather than applying the rules of the protocol for choosing a random back-off period [17]. Back-off manipulation is applicable to both IEEE 802.11 wireless networks and IEEE 802.15.4 wireless sensor networks due to their similar CSMA-CA based protocols. *Same-nonce attack* is related to the access control lists (ACL) identifying the nodes that data can be received from [20]. In order to be used in an encrypted transmission, ACL entry includes the destination address, the key, the nonce and option fields. If the sender uses the same key and nonce pairs within two transmissions, an adversary obtaining those ciphertexts may retrieve useful information [21]. *Replay-protection attack* targets the replay protection mechanism provided in IEEE 802.15.4 specification. This mechanism is used to accept a frame by checking whether the counter of the recent message is larger than the previous one. If an adversary sends many frames with large counters to a legitimate node, the legitimate user using the replay protection mechanism will reject the legitimate frames with small counters from other nodes [20]. *ACK attack* [20] can be accomplished by eavesdropping the channel. An eavesdropper, firstly, may block the receiver node from taking the transmitted packet, then, can mislead the sender node by sending a fake ACK that it comes from the receiver. *PANId conflict attack* [11] creates a fake conflict within a Personal Area Network (PAN). The members of a PAN know the PAN coordinator's identifier (PANId). If there exist more than one PAN coordinator operating in same Personal Operating System (POS), a PANId conflict occurs [10]. An adversary may send fake PANId conflict notification messages to PAN coordinator in order to make PAN coordinator execute conflict resolution procedure, which delays the communication between the PAN coordinator and the legitimate nodes [11].

### *C. Routing Layer Attacks*

Routing layer attacks are usually designed to hinder the route selection mechanism or routing strategy. A routing layer attacker possibly attacks the operation at the network layer at route discovery time, or at route selection time, or after the establishment of the routes [29]. For a wireless sensor network, an example of a routing layer attack on the route discovery process is the *fake route information attack*, which provides incorrect routing data to the network [22]. Some attacks on routing selection processes are i.) *HELLO flood attack* [23] in which the receiving node is convinced that the attacker is within one-hop transmission range when in fact the attacker is carrying out high-power transmission and is far away, ii.) *sinkhole attack* [23]that convinces the attacker's neighboring nodes to forward their packets through the attacker, iii.) *wormhole attack* [24], realized by at least two negotiating attackers using tunneling the packets through a low-delay path established between them to fool the legitimate users for relaying the packets earlier, iv.) *sybil attack* [25] in which the attacker provides more than one different identifications to the network in order to increase his probability of being selected on many routes. An example of the attacks on established routes is *blackhole attack* [26] causing the node to drop all or selectively some received packets. More details about routing layer attack types can be found in [22].

### *D. Transport Layer Attacks*

According to the OSI protocol functions, transport layer provides the data transfer through the management of end-to-end connections. In this manner, Wood et. al. [8] describe the flooding and the desynchronization approaches as two important denial of service attacks.

Based on the classical *TCP SYN flood* [27] approach, an attacker may send many connection requests to a legitimate node.The node's resources, mainly its memory, shall be consumed to maintain those unnecessary connections unless there is a defense mechanism specified in the protocol.

An active connection established between two legitimate nodes can be deteriorated by the *desynchronization attack* [8]. An attacker listening to the connection between two end points can forge messages to either of them in order to make the receiver node request retransmission of related messages from the sender. If the attacker can attack the messages carrying connection specific control data at proper times, the synchronization between the two end points might be lost.

### E. Application Layer Attacks

An interesting case for the application specific sensor networks are the attacks targeting the application itself, including the application data as well as the privacy concerns of the nodes/devices participating in the application. The wireless sensor network attacks targeted at the application layer may be viewed in many and various aspects as the sensor applications constitute a very large variety, from environmental monitoring to medical, military and target tracking applications. Among those attacks, Raymond et. al. [14] discussed the *overwhelming attack* and the *path-based DoS attack*, which aim denial of service.

The overwhelming attack is related to event-based monitoring applications, such as motion detection, in which sensors trigger an action upon detection of an event. An attacker or a group of attackers may try to overwhelm the sensor nodes, which will cause the network to forward a huge amount of traffic to the sink [14].

A path-based DoS attack [28] feeds some replayed packets into the network at the leaf nodes. Through the forwarding of these packets to the sink node, valuable network resources, mainly the bandwidth, would be consumed. The attacker may also decrease the lifetime of the network by making the nodes consume energy via forwarding irrelevant relayed packets.

Furthermore, other attacks can also be constructed within application specific sensor network scenarios. Therefore, various attack types can be modified and specialized to the network application area. In relation to this issue, Misic et. al. [29], for example, analyzed some possible security attacks on healthcare related WSNs, whose sensors are usually deployed on the patients' body.

Figure 1 summarizes above mentioned sensor network attacks with their target protocol layers.

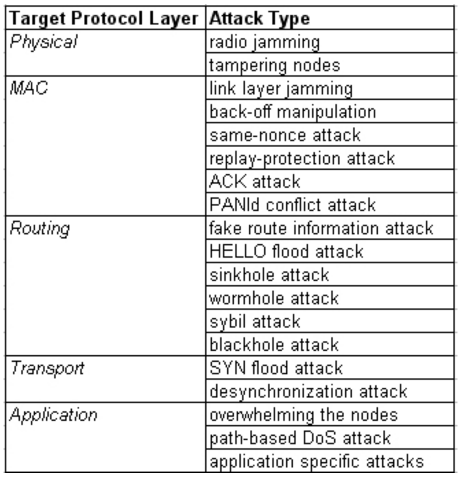
## III. IEEE 802.15.4 Security

In this section some details on security requirements and security modes of IEEE 802.15.4 MAC layer are presented.

### A. Requirements

*Access control*, *confidentiality*, *frame integrity*, *sequential freshness* are the four security requirements specified for IEEE 802.15.4 [20]. Definitions and additional explanations are briefly presented below:

- *Access Control*: Legitimate nodes must be protected from frames of unauthorized nodes. This security requirement is achieved by maintaining an ACL of valid devices [20].



| Target Protocol Layer | Attack Type |
|---|---|
| *Physical* | radio jamming |
| | tampering nodes |
| *MAC* | link layer jamming |
| | back-off manipulation |
| | same-nonce attack |
| | replay-protection attack |
| | ACK attack |
| | PANId conflict attack |
| *Routing* | fake route information attack |
| | HELLO flood attack |
| | sinkhole attack |
| | wormhole attack |
| | sybil attack |
| | blackhole attack |
| *Transport* | SYN flood attack |
| | desynchronization attack |
| *Application* | overwhelming the nodes |
| | path-based DoS attack |
| | application specific attacks |

Fig. 1. Sensor network attacks and target layers.

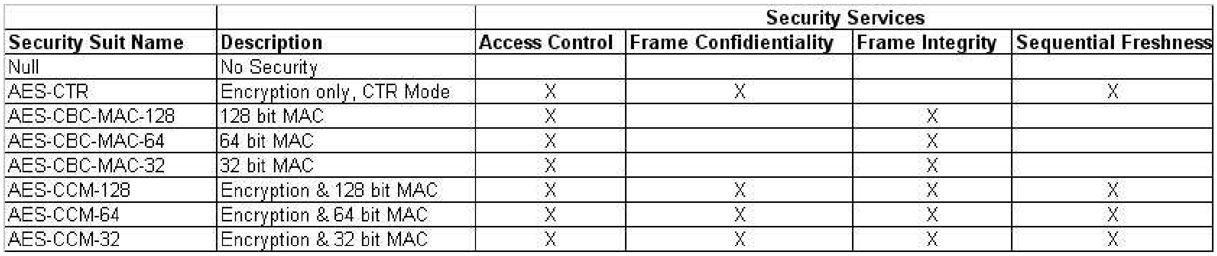
- *Frame Confidentiality*: To make information confidential, only the legitimate nodes must share the secret information [21]. This is done by encryption. Only the legitimate devices that share the secret key can decrypt frames for communication.
- *Frame Integrity*: The frames generated by legitimate nodes must not be manipulated by adversary nodes. The frame integrity is provided by message authentication code (MAC).
- *Sequential Freshness*: Legitimate nodes must not accept old messages (previously replayed). A simple message counter is provided to ensure sequential freshness.

More details on these definitions and requirements can be found in [20], [21].

### B. Modes

There are three security modes to cover the security requirements of different types of application [20]. An ACL includes multiple entries. Each entry is composed of an address (source, destination), a security suit, a shared key, a last initial vector, and a replay counter. The last initial vector is used by the source, while the replay counter is used by the destination for sequential freshness. The modes are listed as follows:

- *Unsecured Mode*: In this mode, no security service is provided. It is used for low cost applications that do not require any security.
- *ACL Mode*: In ACL mode, each node maintains its ACL. In this mode, devices only receive message from those devices in its ACL. No other cryptographic protection is provided.
- *Secured Mode*: All the security requirements (access control, frame confidentiality, frame integrity and sequential freshness) are provided in this mode according to defined security suits. It uses all the fields in the ACL entry format. According to [20], [21], the security suits are summarized in Figure 2.



| | | Security Services | | | |
|---|---|---|---|---|---|
| Security Suit Name | Description | Access Control | Frame Confidientiality | Frame Integrity | Sequential Freshness |
| Null | No Security | | | | |
| AES-CTR | Encryption only, CTR Mode | X | X | | X |
| AES-CBC-MAC-128 | 128 bit MAC | X | | X | |
| AES-CBC-MAC-64 | 64 bit MAC | X | | X | |
| AES-CBC-MAC-32 | 32 bit MAC | X | | X | |
| AES-CCM-128 | Encryption & 128 bit MAC | X | X | X | X |
| AES-CCM-64 | Encryption & 64 bit MAC | X | X | X | X |
| AES-CCM-32 | Encryption & 32 bit MAC | X | X | X | X |

Fig. 2. Security suits.

## IV. GTS ATTACK

This section explains the use of Guaranteed Time Slots in WSN communication. After introducing the communication sequences of GTS allocation and deallocation schemes, the section identifies GTS attack through illustrating various scenarios. Besides the attacks stated in Section II, our GTS attack scenarios contribute WSN attack literature as categorized in MAC layer attack type.

### A. Guaranteed Time Slots of IEEE 802.15.4 MAC Layer

In IEEE 802.15.4 MAC Standards, a superframe structure is allowed to manage the services with and without contention. The superframe is managed by the PAN coordinator. The IEEE 802.15.4 generic superframe structure is shown in Figure 3 [10]. The PAN coordinator sends *BEACON* messages at the beginning of each superframe thus the superframe interval is also called the *beacon interval*. Each *BEACON* message includes the network identifier, beacon periodicity and superframe structure in order to help other network devices to synchronize. The superframe is divided into 16 slots as shown in Figure 3. The network devices communicate with the PAN coordinator in the superframe interval. This duration is called contention access period (CAP) for the generic superframe shown in the figure.

The structure of the superframe can be configured by the PAN coordinator to meet the needs of various applications. For nodes running applications with relaxed latency requirements, the superframe can be partitioned into active and inactive portions as shown in Figure 4. The nodes sleep in the inactive portion. The length of the active and inactive portions are determined in accordance with the application's requirements. The inactive portion of the superfame prevents idle listening thus helps preserving the energy of the battery constrained nodes.

According to the IEEE 802.15.4 standards the PAN coordinator can assign dedicated slots to one or more separate network devices [10]. A slot assigned by the coordinator for communication only with a given device is defined as a Guaranteed Time Slot (GTS). GTSs support applications with particular bandwidth requirements or ones with relax latency requirements. Each GTS can contain a single or an integer multiple of time slots each one being equal to 1/16 of the beacon interval. The superframe structure with the contention free period (CFP), which includes GTSs is shown in Figure 5. There are 7 slots provided for GTS transmission in CFP of the superframe. GTSs are located after the CAP.

A device must track beacons in order to request and get an allocation for a GTS. The PAN coordinator decides whether to accept a GTS allocation request of a device and may give more than one slot if there are available slots. The GTS allocation policy is first-come-first serve. Figure 6 shows the usual communication sequence of a GTS slot allocation procedure.

First of all, the node must receive the beacon successfully in order to synchronize with the coordinator. After receiving the beacon, the node can communicate with the coordinator in CAP. Secondly, the node sends a GTS Allocation request to the PAN coordinator. The GTS request message includes the length and the direction. The GTS direction can be defined as either transmit or receive. On receipt of this command, the PAN coordinator may send an ACK to indicate the successful reception of the GTS request. Then, the PAN coordinator checks for available slots in the current superframe within *aGTSDescPersistenceTime* superframes time. If there are available slots, new GTS information is included in the following beacon. The GTS requesting node receives the beacon and extracts the GTS transmission time if it is inserted by the PAN coordinator. In this case, the GTS transmission is successfully achieved as seen in Figure 6. If no GTS descriptor is found in the superframe, the node notifies the next upper layer of failure. The device can deallocate its GTS in the same way as shown in Figure 7.

The above mentioned GTS management including request, allocation and deallocation is based on the IEEE 802.15.4 explicit procedure/algorithm [10]. In addition to this procedure, some modified GTS allocation schemes have also been proposed. Ji et. al. [30] proposed an efficient GTS allocation algorithm for IEEE 802.15.4 that is capable of traffic analysis. Their GTS allocation scheme is based on packet arrival rate and number of devices in the network. When devices are transmitting, the ones with the higher packet transmission rate can cause more collisions and longer delay compared to the ones with the lower rate. So, their scheme allocates the GTS slots to devices with the higher packet rates. The proposed GTS algorithm also takes into consideration the number of nodes because there are at the most 7 GTS slots available for allocation. Ji et. al. [30] constructed a 17-node IEEE 802.15.4 star topology in order to compare their proposed GTS allocation mechanism with the standard one.

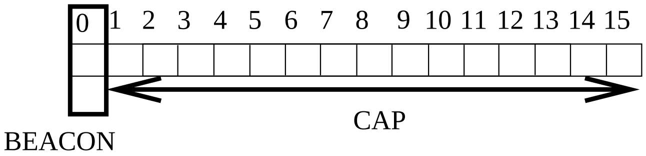


Fig. 3. IEEE 802.15.4 generic superframe structure.

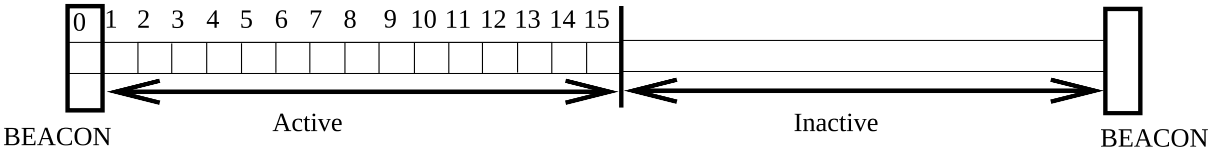


Fig. 4. IEEE 802.15.4 Superframe structure with active and inactive portions.

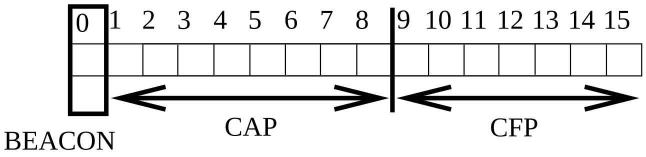


Fig. 5. IEEE 802.15.4 superframe structure with GTS.

By tracing the packet delivery rates, it is shown that their proposed scheme achieves 16 % higher throughput than the standard one. Additionally, the amount of dropped packets caused by collisions is decreased significantly. By tuning the algorithm's parameters, they reach a 18 % improvement on average throughput.

One of the basic disadvantage of the standard GTS management scheme is that the number of nodes having GTS slots is limited to 7. So, the GTS slots can be quickly consumed by a few number of nodes and devices with low data rates can cause the underutilization of the GTS resources. To overcome these problems, Koubaa et. al. [31], [32] proposed a GTS allocation approach, which is based on the idea that a slot can be used by more than one node. By considering the arrangement of GTS request arrivals with traffic specifications and the delay parameters, their algorithm makes a decision about the slot sharing policy among the nodes sending requests. They provide a kind of round-robin scheduling mechanism to prevent starvation, however they indicate that some modified scheduling schemes can be used. They implemented the proposed GTS algorithm with nesC on micaZ platforms. Their experimental test bed includes 1 PAN coordinator and 7 motes which are located within the transmission range of the PAN coordinator. The experiment results show that this implicit GTS management mechanism, i-GAME, is more efficient in bandwidth utilization than the explicit one defined in IEEE 802.15.4 standard.

### B. Identified GTS attack

As described in [1], GTS attack is based on the inherent properties of the IEEE 802.15.4 superframe organization in beacon-enabled operational mode for WSNs. GTS slots create a vulnerable point which can allow an attacker to disrupt the communication between a device and its PAN coordinator. A possible attack scenario using the GTS interval is illustrated in Figure 8. Assume that all the nodes as well as the adversary, which is an intelligent attacker device, have achieved synchronization with the coordinator by receiving beacon messages. A legitimate node may request a GTS slot by sending a GTS request command to the PAN coordinator including the GTS descriptor. The PAN coordinator may respond with an optional ACK for this GTS request. Meanwhile the coordinator handles the GTS request. The coordinator may accept the GTS request and allocate demanded GTS slot(s) or may reject it. The accepted requests are announced in the following beacon message broadcasted to all nodes. The adversary can learn the GTS slot times by extracting the GTS descriptor(s) from the beacon frame. After obtaining the allocated GTS times, the adversary can create interference at any of these moments. This interference will cause collision and corruption of the data packets between the legitimate GTS node and the coordinator. The collision occurring during the GTS period can be considered as a kind of DoS paradigm since these slots are assumed to provide collision-free communication.

### C. Evaluation

We have simulated the proposed GTS attack implementation on ns-2.31 [33]. ns-2.31 comes with IEEE 802.15.4 MAC layer protocol in which GTS data structures are defined but GTS management methods are not implemented [34], [35]. In the simulations, we have implemented and used the explicit GTS management mechanism defined in IEEE 802.15.4 MAC layer standard [10].

Two types of attackers are defined in the simulations: intelligent attacker and random attacker. An intelligent attacker aims at corrupting the communication in the GTS slot with

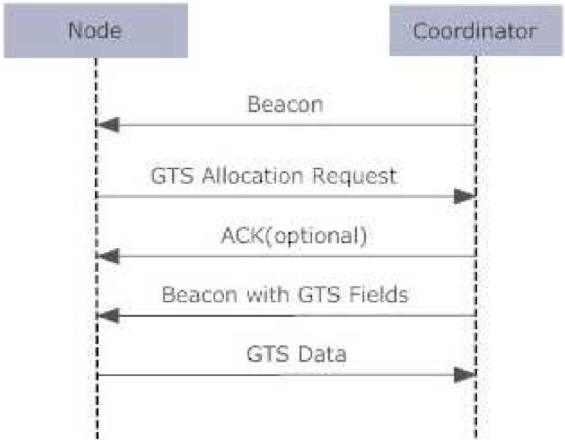


Fig. 6. Communication sequence in GTS allocation.

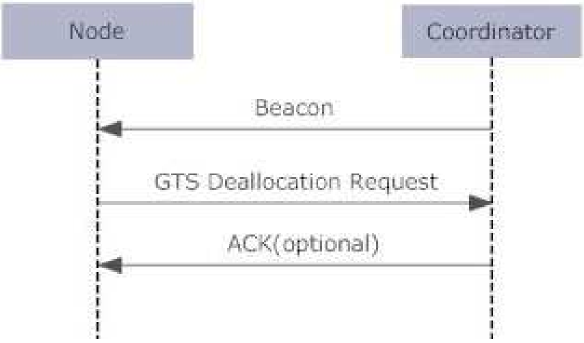


Fig. 7. Communication sequence in GTS deallocation.

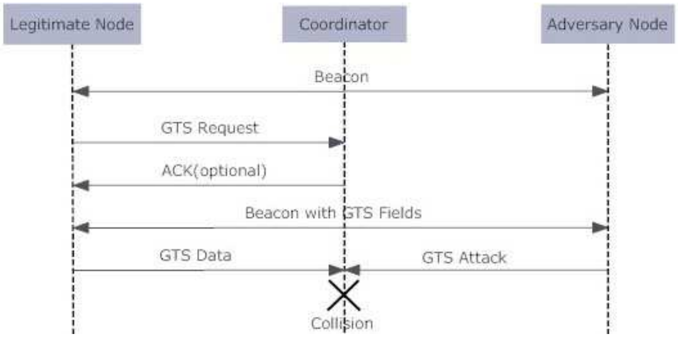


Fig. 8. Communication sequence in GTS attack scenario.

maximum length in the CFP, whereas a random attacker randomly chooses a GTS slot to be attacked. Attacking a slot, which is allocated for communication between the PAN coordinator and a legitimate user, can be achieved by creating a collision through jamming or sending messages in that slot. In our simulations, both attackers corrupt the communication by sending a message to the coordinator at the starting time of the selected GTS slots.

A star network with ten nodes has been simulated, of which at most two attackers are on duty. Four types of scenarios are defined: ”one intelligent attacker” (OIA), ”one random attacker” (ORA), ”two intelligent attackers” (TIA), and ”two random attackers” (TRA). It is expected that, for the ORA scenario, the adversary attacks the allocated slot of an average length communication. In the case of TRA scenario, two attackers may attack two different communications or may attack the same communication, in which case the energy of the attackers is consumed ineffectively to corrupt the same node communication. In contrast to this, an intelligent attacker can use its energy in a more efficient manner. It can attack the first slot of the communication with maximum slot length thus destroying the whole communication. For the TIA scenario, the adversaries can cooperatively attack the nodes with one of them attacking the maximum length communication (with the maximum number of slots allocated) and the other attacking the communication with the second maximum slot length. For this last scenario, the common goal of the attackers is to cause maximum possible decrease in bandwidth utilization within the CFP period. Table I summarizes the definitions of the attack scenarios used in simulations.

TABLE I
ATTACK SCENARIOS

| *No* | *Name* | *Definition* |
|---|---|---|
| 1 | OIA | One Intelligent Attacker |
| 2 | ORA | One Random Attacker |
| 3 | TIA | Two Intelligent Attackers |
| 4 | TRA | Two Random Attackers |

In the simulations we have used the predetermined GTS request schedule of the nodes presented in Table II. According to this, the request of node 8, which is for 5 slots in length, can not be granted due to the remaining capacity of 4 out of 7 CFP slots after the reservation of 3 slots for node 7. The requests of node 6, node 4, and node 5 are granted for the communication within the remaining free slots sequentially. It is observed that the accepted requests are announced in the GTS field attribute of the following beacons as shown in Figure 9.

TABLE II
GTS REQUEST SCHEDULE

| *NodeID* | *Request Length(slots)* | *Request Time(s)* |
|---|---|---|
| 7 | 3 | 25 |
| 8 | 5 | 28 |
| 6 | 2 | 31 |
| 4 | 1 | 35 |
| 5 | 1 | 40 |

In the attack scenario experiments, node 0 is the PAN coordinator, nodes 1 and 2 are selected as intelligent attackers, nodes 3 and 9 are selected as random attackers, and the rest are the ordinary nodes. The simulation results are gathered for 60 s where the beacon interval is set to 0.98304 s. The number of total attack messages sent, and corrupted slots for the four different scenarios OIA, ORA, TIA and TRA respectively are given in Table III.

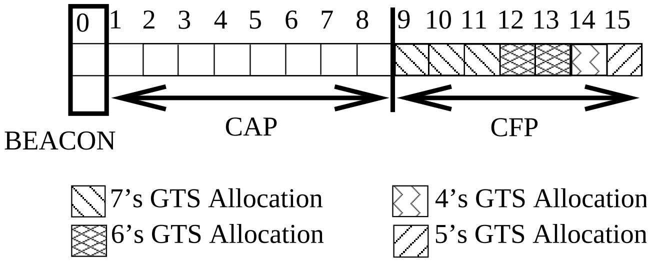


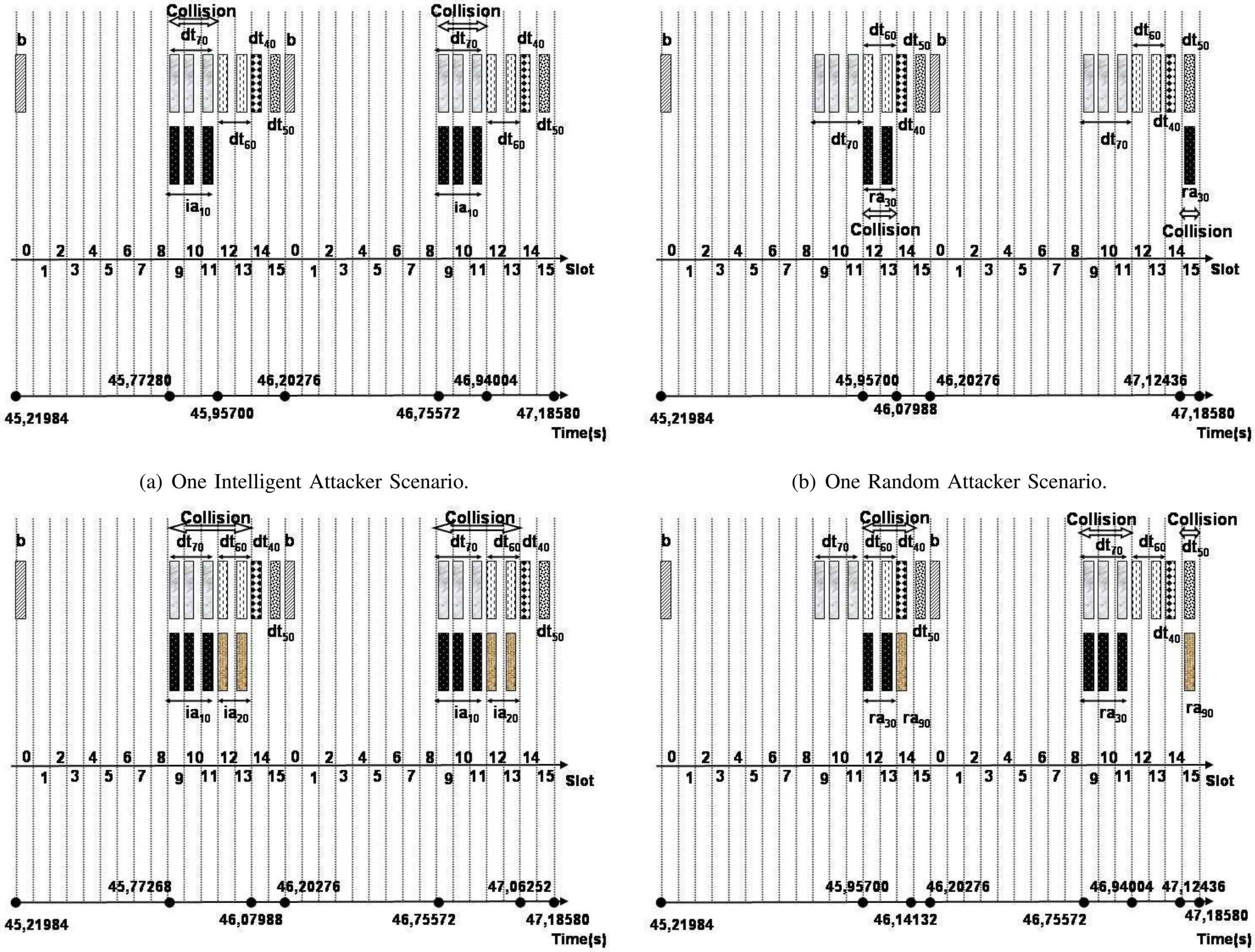
Fig. 9. Granted GTS allocation.

TABLE III
THE NUMBER OF ATTACK MESSAGES AND CORRUPTED SLOTS

| *Scenario Name* | *Attack Messages* | *Corrupted Slots* |
|---|---|---|
| OIA | 35 | 105 |
| ORA | 35 | 69 |
| TIA | 64 | 163 |
| TRA | 70 | 92 |

Figure 10 illustrates the collisions on the related communication between legitimate nodes and the PAN coordinator for the relevant scenarios. The figures indicate the details of the transfers of 2 sequential superframe structures on given times measured in simulation experiments. In all subfigures, the first frame transfer starts at 45,21984 s, which is the 46th beacon transmission time in the experiments. According to our simulation settings in Figure 9, the last GTS request is made at 40 s, which corresponds to the 41st ($\lceil 40/0.98304 \rceil$) beacon transmission. It is clear from the relation between the data presented in Table II and Figure 9 that all sequential frames transmitted after the 42nd beacon shall include the same communication pattern. As an example, the 46th beacon at 45,21984 s in the figures is chosen to be the beacon *b* of the first frame. The slots between 9 and 15 correspond to the slots of the CFP periods for the related superframes. The data transfer, *dt*, in those slots corresponds to the guaranteed amount of data communication between the nodes that have been granted the requested GTS. For example, $dt_{70}$ refers to the data communication from node 7 to the PAN coordinator (node 0). The attack messages sent by the attacker are shown as *ia* for the intelligent attacker(s), and *ra* for the random attacker(s). For example, $ia_{10}$ refers to the attack message sent from the intelligent attacker node 1 to the PAN coordinator. When an attacker sends its attack messages concurrently with the data communication between a node and the PAN coordinator, a collision occurs. In the OIA scenario given in Figure 10.a, the communication of node 7, shown as $dt_{70}$, is corrupted by node 1, shown as $ia_{10}$, in between 9th and 10th time slots. In the ORA scenario given in Figure 10.b, TIA in Figure 10.c, and TRA in Figure 10.d, the same notation is used to demonstrate the relevant collisions.

In the OIA scenario, node 1 corrupts 35 different data transfers each of 3 slot length belonging to node 7 causing all together 105 slots to be corrupted. It means that, the data of 105 slots out of 208 slots is affected by the attack. Assuming all other parameters equal, this attack results in 105/208 (50.48 %) decrease in bandwidth utilization during the CFP period. Node 3 corrupts 35 different data transfers with random slot lengths leading to 69 slot corruptions in the ORA scenario. So, the utilization decrease in the second case is 33.17 %. In the third case, two attackers totally broadcast 64 attack messages that result in 163 corruptions leading to a 78.37 % decrease in utilization. The two random attackers in the fourth scenario totally corrupt 92 slots using 70 attack messages and decrease the utilization by 44.23 %. In order to numerically evaluate



(a) One Intelligent Attacker Scenario.

(b) One Random Attacker Scenario.

(c) Two Intelligent Attackers Scenario.

(d) Two Random Attackers Scenario.

Fig. 10. The collisions on different attacker scenarios

the damaging effects of the attacker and compare the different scenarios in the following we introduce two new variables. The first one, related to the attacker's behavior, is called *corruption strength* and is defined as the ratio of the number of damaged slots to the total number of slots of data. The *transmission strength* on the other hand describes the node's behavior and is defined as the ratio of the number of slots with successfully completed transmission to the total number of slots. The corruption strength and the transmission strength are visualized in Figure 11. Depending on the corrupted slots per unit time, the best scenario from the attacker's point of view is the TIA, the worst scenario is ORA as seen in Figure 12. Consequently, the intelligent attack method causes more damage to the sensor network communication than the random attack, and cooperating attackers decrease bandwidth utilization in CFP period more than a single attacker.

To evaluate the effectiveness of the atttacks we introduce another parameter - the energy consumed by the attacker for achieving a certain degree of damage. ns-2 supports the simulation of energy use of the sensor nodes, therefore the energies of the attackers have been traced within the simulations. Using the scenarios in Table II, the energy consumptions of one intelligent attacker, one random attacker, two intelligent attackers, and two random attackers during their 60-second attack period is plotted in Figure 13. Figure 13 includes the consumed energies of the attackers for corrupting the communication slots. The energy exhaustion for each corrupted communication is calculated and recorded at the attacker node by subtracting the current traced energy level from their previous values after each attack. As seen in Figure 13, the slopes of the intelligent attackers' energy consumption curves are lower than the ones of the random attackers'. Therefore, intelligent attackers consume less energy per corrupted slot than random attackers.

Neither the intelligent attacker nor the random attacker can be easily detected in GTS attack cases. Since the attackers are synchronized with the PAN coordinator in a fine-grained manner, the attack messages, which reveal collisions in the channel, cannot be received by the coordinator. Therefore, the coordinator can not perceive the ID of the attacker.

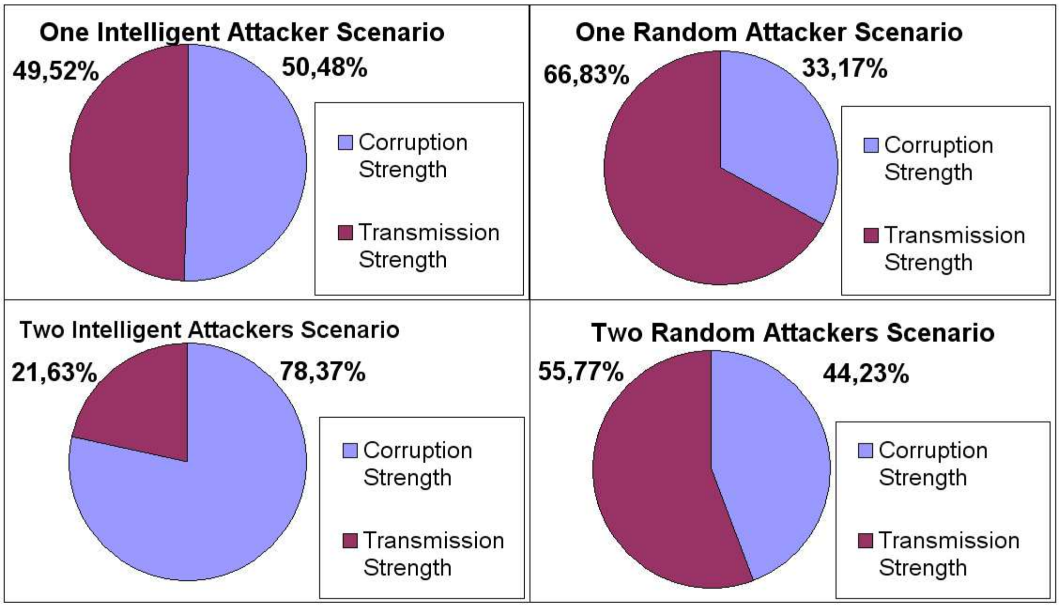


Fig. 11. Transmission and corruption strength.

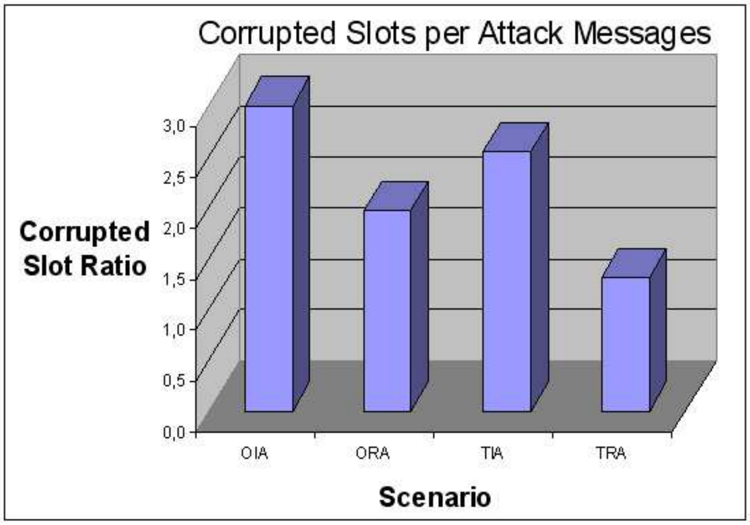


Fig. 12. Corrupted slots per attack messages.

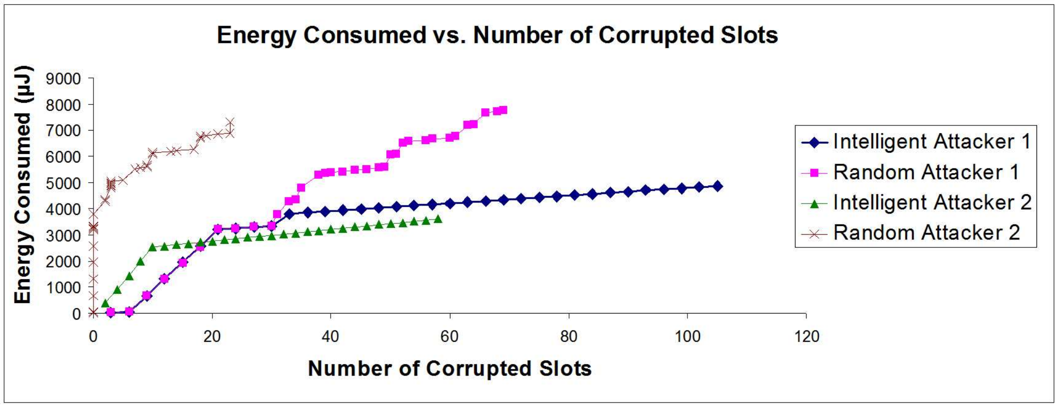


Fig. 13. Energy consumed vs number of corrupted slots.

However, if the synchronization between the attacker and the PAN coordinator is not fine-grained but still allowing to communicate with a small drift in the attacker's clock, the adversary can emit regular packets in the GTS interval to corrupt the communication, but is not able to synchronize precisely with the CFP slots. This allows the coordinator to detect the attack and extract his *ID* by from the source field of the received packets. In other cases, in which the adversary emits jamming signals instead of regular packets or emits regular packets with precise synchronization, GTS attack is considered very hard to detect.

## V. Conclusions

This paper investigates WSN attacks including a brief survey of physical layer, MAC layer, routing layer, transport layer, and application layer attacks. Furthermore, a new IEEE 802.15.4 MAC layer attack, the GTS attack [1], is defined and evaluated with respect to intelligent and random attacker behavior scenarios.

Based on the definition of the GTS attack, a sample communication sequence of this attack, exploring the IEEE 802.15.4 specification, is designed. It has been shown that a GTS attack is quite possible to realize. The implementation of the suggested approach with different scenarios is built using ns-2.31. To study their effects on the communication process during the CFP periods, the number of total corrupted slots and the number of total collisions are analyzed in various attacker cases, and the bandwidth utilization and energy consumption evaluations of the results are presented.

In order to numerically evaluate the effects of the different attack scenarios two new variables, the corruption strength and the transmission strength are introduced. It is observed that the intelligent attacker can achieve a corruption strength of up to 78.37 % which actually means that only one quarter of the available bandwidth is actually used for the communication during the CFP period.

Another aspect that has been evaluated is the energy consumption from the point of view of the attacker. An intelligent GTS attacker uses the energy much more efficiently than a random GTS attacker. On the whole, the intelligent attack method causes more damage to the sensor network communication requiring less energy from the attacker node as compared to the random attack method.

Future work directions will focus on tunning different parameters in the GTS attack scenarios. The detection probability will be investigated when there is a lack of fine-grained time synchronization between the PAN coordinator and the GTS attacker. Additionally, a GTS-based application will be simulated and analyzed under GTS attack conditions.

## References

[1] R. Sokullu, O. Dagdeviren, and I. Korkmaz, ”On the IEEE 802.15.4 MAC Layer Attacks: GTS Attack”, *in Proc. of SENSORCOMM08*, 2008, pp. 673-678.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, ”A Survey on Sensor Networks”, *IEEE Communications Magazine*, vol.40, no.8, 2002, pp. 102-114.

[3] M. Kuorilehto, M. Hannikainen, and T. D. Hamalainen, ”A Survey of Application Distribution in Wireless Sensor Networks”, *EURASIP Journal on Wireless Communications and Networking*, vol.5, no.5, 2005, pp. 774-788.

[4] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, ”Wireless Sensor Networks for Habitat Monitoring”, *in Proc. of ACM International Workshop on Wireless Sensor Networks and Applications*, 2002, pp. 88-97.

[5] E. Biagioni and K. Bridges, ”The Application of Remote Sensor Technology to Assist the Recovery of Rare and Endangered Species”, *in Special issue on Distributed Sensor Networks for the International Journal of High Performance Computing Applications*, vol.16, no.3, 2002, pp. 315-324.

[6] L. Schwiebert, S. K. S. Gupta, and J. Weinmann, ”Research Challenges in Wireless Networks of Biomedical Sensors”, *in Proc. of Mobile Computing and Networking*, 2001, pp. 151-165.

[7] M. B. Srivastava, R. R. Muntz, and M. Potkonjak, ”Smart Kindergarten: Sensorbased Wireless Networks for Smart Developmental Problem-Solving Enviroments”, *in Proc. of Mobile Computing and Networking*, 2001, pp. 132-138.

[8] A. D. Wood and J. A. Stankovic, ”A Taxonomy for Denial-of-Service Attacks in Wireless Sensor Networks”, *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, CRC Press, 2004.

[9] Committee on National Security Systems (CNSS), *National Information Assurance Glossary*, NSTISSI no.4009, 2003.

[10] IEEE Std 802.15.4TM-2003, IEEE Standard for Information technology-Telecommuncations and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).

[11] R. Sokullu, I. Korkmaz, O. Dagdeviren, A. Mitseva, and N.R. Prasad, ”An Investigation on IEEE 802.15.4 MAC Layer Attacks”, *in Proc. of WPMC*, 2007.

[12] S. Radosavac, J.S. Baras, and I. Koutsopoulos, ”A Framework for MAC Layer Misbehavior Detection in Wireless Networks”, *in Proc. of the 4thACM Workshop on Wireless security*, 2005, pp. 33-42.

[13] H. Chan and A. Perrig, ”Security and Privacy in Sensor Networks”, *IEEE Computer*, IEEE, vol.36, no.10, 2003, pp. 103-105.

[14] D. R. Raymond and S. F. Midkiff, ”Denial-of-Service in Wireless Sensor Networks: Attacks and Defenses”, *IEEE Pervasive Computing*, vol.7, no.1, 2008, pp. 74-81.

[15] A. Wood and J. A. Stankovic, ”Denial of Service in Sensor Networks”, *IEEE Computer*, vol.35, no.10, 2002, pp. 54-62.

[16] V.B. Misic, J. Fung, and J. Misic, ”MAC Layer Attacks in 802.15.4 Sensor Networks”, *Security in Sensor Networks*, Auerbach Publications, Taylor & Francis Group, 2007, pp. 27-44.

[17] S. Radosavac, A.A. Crdenas, J.S. Baras, and G.V. Moustakides, ”Detecting IEEE 802.11 MAC Layer Misbehavior in Ad Hoc Networks: Robust Strategies against Individual and Colluding Attackers”, *Journal of Computer Security, special Issue on Security of Ad Hoc and Sensor Networks*, vol.15, no.1, 2007, pp. 103-128.

[18] W. Xu, K. Ma, W. Trappe, and Y. Zhang, ”Jamming Sensor Networks: Attack and Defense Strategies”, *IEEE Network*, vol.20, no.3, 2006, pp. 41-47.

[19] Y.W. Law, P. Hartel, J. den Hartog, and P. Havinga, ”Link-Layer Jamming Attacks on S-MAC”, *in Proc. of IEEE WSN*, 2005, pp. 217-225.

[20] Y. Xiao, S. Sethi, H.H. Chen, and B. Sun, ”Security Services and Enhancements in the IEEE 802.15.4 Wireless Sensor Networks”, *in Proc. of IEEE GLOBECOM*, vol.3, 2005.

[21] N. Sastry and D. Wagner, ”Security Considerations for IEEE 802.15.4 Networks”, *in Proc. of the ACM Workshop on Wireless Security*, 2004, pp. 32-42.

[22] Y.C. Wang and Y.C. Tseng, ”Attacks and Defenses of Routing Mechanisms in Ad Hoc and Sensor Networks”, *Security in Sensor Networks*, Auerbach Publications, Taylor & Francis Group, 2007, pp. 3-25.

[23] C. Karlof and D. Wagner, ”Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures”, *in Proc. of IEEE SNPA*, vol.1, 2003, pp. 113-127.

[24] Y.C. Hu, A. Perrig, and D.B. Johnson, ”Packet Leashes: A Defense

against Wormhole Attacks in Wireless Networks", *in Proc. of IEEE INFOCOM*, vol.1, 2003, pp. 1976-1986.

[25] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil Attack in Sensor Networks: Analysis & Defenses", *in Proc. of IPSN*, vol.1, 2004, pp. 259-268.

[26] H. Deng, W. Li, and D.P.Agrawal, "Routing Security in Wireless Ad Hoc Networks", *IEEE Communications Magazine*, vol.40, no.10, 2002, pp. 70-75.

[27] C.L. Schuba, I.V. Krsul, M.G. Kuhn, E.H. Spafford, A. Sundaram, and D. Zamboni, "Analysis of a Denial of Service Attack on TCP", *in Proc. of IEEE Symp. Security and Privacy*, 1997, pp. 208-223.

[28] J. Deng, R. Han, and S. Mishra, "Defending against Path-Based DoS Attacks in Wireless Sensor Networks", *in Proc. of 3rd ACM Workshop Security of Ad Hoc and Sensor Networks*, 2005, pp. 89-96.

[29] J. Misic, F. Amini, and M. Khan, "On Security Attacks in Healthcare WSNs Implemented on 802.15.4 Beacon Enabled Clusters", *in Proc. of IEEE Consumer Communications and Networking Conference*, 2007, pp. 741-745.

[30] Y. Ji, W. Park, S. Kim, and S. An, "Efficient GTS Allocation Algorithm for IEEE 802.15.4", *in Proc. of ICCS*, 2007, pp. 869-872.

[31] A. Koubaa, M. Alves, and E. Tovar, "i-GAME: An Implicit GTS Allocation Mechanism in IEEE 802.15.4 for Time-Sensitive Wireless Sensor Networks", *in Proc. of ECRTS*, 2006, pp. 183-192.

[32] A. Koubaa, M. Alves, and E. Tovar, "Time Sensitive IEEE 802.15.4 Protocol", *Sensor Networks and Configuration*, Springer, 2007, pp. 19-49.

[33] K. Fall and K. Varadhan, "The ns manual", `http://www.isi.edu/nsnam/ns/doc`, 2007.

[34] J. Zheng and M.J. Lee, "A Comprehensive Performance Study of IEEE 802.15.4", *Sensor Network Operations*, IEEE Press, Wiley Interscience, 2006, pp. 218-237.

[35] I. Ramachandran, A.K. Das, and S. Roy, "Analysis of the Contention Access Period of IEEE 802.15.4 MAC", *ACM Transactions on Sensor Networks*, vol.3, no.1, 2007.

# SFN Gain Simulations in Non-Interfered and Interfered SFN Network

Jyrki T.J. Penttinen
*Member, IEEE*
*jyrki.penttinen@nsn.com*

## Abstract

*The DVB-H (Digital Video Broadcasting, Handheld) coverage area depends mainly on the area type, i.e. on the radio path attenuation, as well as on the transmitter power level, antenna height and radio parameters. The latter set has effect also on the audio / video capacity. In the detailed network planning, not only the coverage itself is important but the quality of service level should be dimensioned accordingly.*

*This paper describes the SFN gain related items as a part of the detailed radio DVB-H network planning. The emphasis is put to the effect of DVB-H parameter settings on the error levels caused by the over-sized Single Frequency Network (SFN) area. In this case, part of the transmitting sites converts to interfering sources if the safety distance margin of the radio path is exceeded. A respective method is presented for the estimation of the SFN interference levels. The functionality of the method was tested by programming a simulator and analyzing the variations of carrier per interference distribution. The results show that the theoretical SFN limits can be exceeded e.g. by selecting the antenna height in optimal way and accepting certain increase of the error level that is called SFN error rate (SER) in this paper. Furthermore, by selecting the relevant parameters in correct way, the balance between SFN gain and SER can be planned in controlled way.*

***Index Terms****—Mobile broadcast, single frequency network, radio planning, performance evaluation.*

## 1. Introduction

The DVB-H is an extended version of the terrestrial television system, DVB-T. Both are defined in the ETSI standards along with the satellite and cable versions of the DVB.

The mobile version of DVB suits especially for the moving environment as it has been optimized for the fast variations of the field strength and different terminal speeds. Furthermore, DVB-H is suitable for the delivery of various audio / video channels in a single bandwidth, and the small terminal screen shows adequately the lower resolution streams compared to the full scale DVB-T.

As DVB-H is meant for the mobile environment, the respective terminals are often used on a street level for the reception. This creates a significant difference in the received power level compared to DVB-T which uses fixed and directional rooftop antenna types. Furthermore, the DVB-H terminal has normally only small, in-built panel antenna, which is challenging for the reception of the radio signals.

The DVB-H service can be designed using either Single Frequency Network (SFN) or Multi Frequency Network (MFN) mode. In the former case, the transmitters can be added within the SFN area without co-channel interferences even if the cells of the same frequency overlap. In fact, the multi-propagated SFN signals increases the performance of the network by producing SFN gain.

Especially in the Single Frequency Network, the coverage planning is straightforward as long as the maximum distance of the sites does not exceed the allowed value defined by the guard interval (GI). The guard interval takes care of the safe reception of the multi-path propagated signals originated from various sites or due to the reflected radio waves. If the GI and FFT dependent geographical SFN boundary is exceeded, part of the sites starts to act as interferers instead of providing useful carrier.

The maximum size of the non-interfered Single Frequency Network of DVB-H depends on the guard interval and FFT mode. The distance limitation between the extreme transmitter sites is thus possible to calculate in ideal conditions. Nevertheless, there might be need to extend the theoretical SFN areas e.g. due to the lack of frequencies.

Sites that are located within the SFN area minimises the effect of the inter-symbol-interferences as the guard interval protects the OFDM signals of DVB-H, although in some cases, sufficiently strong multipath signals reflected from distant objects might cause interferences in tightly dimensioned network. On the other side, if certain degradation in the quality level of the received signal is accepted, it could be justified to even extend the SFN limits.

This paper is an extension to [1] and presents first DVB-H radio network dimensioning and SFN principles. A simulation method that was developed for estimating the balance of the SFN interference level and the SFN gain, is presented next. Case studies were carried out by utilizing a set of DVB-H radio parameters. The results shows the variations in the carrier per noise and interference levels, $C / (N + I)$, in function of related radio parameters in over-sized SFN.

## 2. DVB-H Dimensioning

The main link budget items affecting on the dimensioning of the DVB-H network are capacity and coverage related parameters. They also have inter-dependencies so the final dimensioning requires iterative approach.

### 2.1. Capacity Planning

In the initial phase of the DVB-H network planning, the offered capacity of the system is dimensioned. The total capacity in certain DVB-H band – defined as 5, 6, 7, or 8 MHz – does have effect also on the size of the coverage area. The dimensioning process is thus iterative, with the aim to find a balance between the capacity, coverage and the cost of the network.

The capacity can be varied by tuning the modulation, guard interval, code rate and channel bandwidth. As an example, the parameter set of QPSK, GI ¼, code rate ½ and channel bandwidth 8 MHz provides a total capacity of 4.98 Mb/s, which can be divided between one or more electronic service guides (ESG) and various audio / video sub-channels with typically around 200-500 kb/s bit stream dedicated for each. The capacity does not depend on the number of carriers (FFT mode) but the selected FFT affects though on the Doppler shift tolerance. As a comparison, the parameter set of 16-QAM, GI 1/32, code rate 7/8 and channel bandwidth of 8 MHz provides a total capacity of 21.1 Mb/s. It should be noted, though, that the latter parameter set is not practical due to the clearly increased *C*/*N* requirement. The relation between the radio parameter values, Doppler shift tolerance and capacity can be investigated more thoroughly in [2].

### 2.2. Coverage Planning

When the coverage criteria are known, the cell radius can be estimated by applying the link budget calculation. The generic principle of the DVB-H link budget can be seen in Table 1. The calculation shows an example of the transmitter output power level of 2,400 W with the quality value of 90% for the area location, though assuming that the SFN gain does not exist. According to the link budget, the outdoor reception of this specific case yields a successful reception when the radio path loss is equal or less than 140.3 dB.

Table 1. An example of DVB-H link budget.

| ***Parameters*** | ***Symbol*** | ***Value*** |
|---|---|---|
| **General parameters** | | |
| Frequency | $f$ | 680.0 MHz |
| Noise floor for 6 MHz BW | $P_n$ | -106.4 dBm |
| RX noise figure | $F$ | 5.2 dB |
| **Transmitter (TX)** | | |
| Transmitter output power | $P_{TX}$ | 2,400.0 W |
| Transmitter output power | $P_{TX}$ | 63.8 dBm |
| Cable and connector loss | $L_{cc}$ | 3.0 dB |
| Power splitter loss | $L_{ps}$ | 3.0 dB |
| Antenna gain | $G_{TX}$ | 13.1 dBi |
| Antenna gain | $G_{TX}$ | 11.0 dBd |
| Eff. Isotropic Radiating Power | $EIRP$ | 70.9 dBm |
| Eff. Isotropic Radiating Power | $EIRP$ | 12,308.7 W |
| Eff. Radiating Power | $ERP$ | 68.8 dBm |
| Eff. Radiating Power | $ERP$ | 7,502.6 W |
| **Receiver (RX)** | | |
| Min. C/N for the used mode | $(C/N)_{min}$ | 17.5 dB |
| Sensitivity | $P_{RXmin}$ | -83.7 dBm |
| Antenna gain, isotropic ref. | $G_{RX}$ | -7.3 dBi |
| Antenna gain, ½ wave dipole | $G_{RX}$ | -5.2 dBd |
| Isotropic power | $P_i$ | -76.4 dBm |
| Loc. variation. | $L_{lv}$ | 7.0 dB |
| Building loss | $L_b$ | 14.0 dB |
| GSM filter loss | $L_{GSM}$ | 0.0 dB |
| Min. req. received power outd. | $P_{min(out)}$ | -69.4 dBm |
| Min. req. received power ind. | $P_{min(in)}$ | -55.4 dBm |
| Min. req. field strength outd. | $E_{min(out)}$ | 64.5 dBµV/m |
| Min. req. field strength ind. | $E_{min(in)}$ | 78.5 dBµV/m |
| Maximum path loss, outdoors | $L_{pl(out)}$ | 140.3 dB |
| Maximum path loss, indoors | $L_{pl(in)}$ | 126.3 dB |

The Okumura-Hata model [3] can be applied in order to obtain the estimation for the cell radius (unit in kilometres) e.g. in large city type:

$$\begin{aligned} L(dB) &= 69.55 + 26.16\lg(f) - 13.82\lg(h_{BS}) \\ &- a(h_{MS})_{type} + \left[44.9 - 6.55\lg(h_{BS})\right]\lg(d) \end{aligned} \quad (1)$$

For $f$ ≥400 MHz, the area type factor is:

$$a(h_{MS})_{LC} = 3.2\left[\lg\left(11.75 h_{MS}\right)\right]^2 - 4.97 \tag{2}$$

$$d = 10^{\left(\frac{L(dB)-\left[69.55+26.16\lg(f)-13.82\lg(h_{BS})-a(h_{MS})_i\right]}{44.9-6.55\lg(h_{BS})}\right)} \tag{3}$$

Figure 1 presents the estimated cell range of the example that is calculated with the large city model and by varying the transmitter antenna height and power level. As can be noted, the antenna height has major impact on the cell radius compared to the transmitter power level.

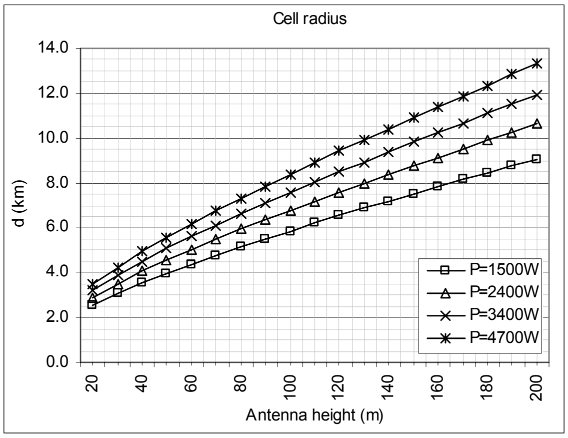


Figure 1. The cell range calculated with the Okumura-Hata model for the large city, varying the transmitter power levels.

In the SFN related reference material, different values for the SFN gain is proposed to be added to the DVB-H link budget, typically from 0 to 3 dB. As an example, [4] mentions that due to the large standard variation of the combined signal reception in environment with multi-path propagation, no SFN gain is recommended for the planning criteria. In the same document, a SFN gain of around 1.5 dB was obtained for the 2 transmitter case by carrying out field measurements. A 3-transmitter field measurement case can be found in [5], which shows that about 2 dB SFN gain was achieved. For the large amount of sites, no practical field results can be found due to the complexity of the test setup.

Nevertheless, the possible SFN gain of, e.g., 2 dB would have around same effect on the coverage area growth as changing the 1500 W transmitter to 2400 W power level category, i.e. there can be an important cost effect in selected areas of the DVB-H network depending on the functionality of the SFN gain.

## 3. Theory of the SFN Limits

The DVB-H radio transmission is based on the OFDM (Orthogonal Frequency Division Multiplexing). The idea of the technique is to create various separate data streams that are delivered in sub-bands. The error correction is thus efficient as the sufficiently high-quality sub-bands are used for the processing of the received data, depending on the level of the correction schemes used in the transmission.

In order to work, the system needs to minimize the interference levels between the sub-bands. The sub-band signals should thus be orthogonal. In order to comply with this requirement, the carrier frequencies are selected in such way that the spacing between the adjacent channels is the inverse of symbol duration.

According to [2], the GI and the FFT mode determinate the maximum delay that the mobile can handle for receiving correctly the multi-path components of the signals. The Table 2 summarises the maximum allowed delays and respective distances. The maximum allowed distance per parameter setting has been calculated assuming the radio signal propagates with the speed of light.

Table 2. The guard interval lengths and respective safety distances.

| GI | *FFT = 2K* | *FFT = 4K* | *FFT = 8K* |
|---|---|---|---|
| 1/4 | 56 μs / 16.8 km | 112 μs / 33.6 km | 224 μs / 67 km |
| 1/8 | 28 μs / 8.4 km | 56 μs / 16.8 km | 112 μs / 33.6 km |
| 1/16 | 14 μs / 4.2 km | 28 μs / 8.4 km | 56 μs / 16.8 km |
| 1/32 | 7 μs / 2.1 km | 14 μs / 4.2 km | 28 μs / 8.4 km |

As long as the distance between the extreme transmitter sites is less than the safety margin dictates, the difference of the delays between the signals originated from different sites never exceeds the allowed value unless there is a strong multipath propagated signal present.

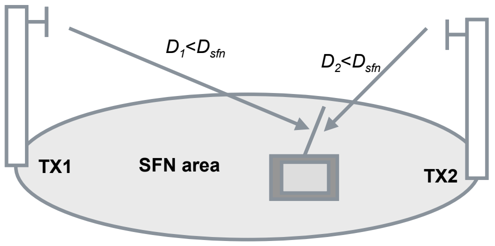


Figure 2. If all the sites of certain frequency band are located inside the SFN area with the distance between the extreme sites less than $D_{sfn}$, no inter-symbol interferences are produced.

On the other hand, when the terminal drifts outside of the original SFN area and receives sufficiently strong signals from the original SFN, no problems arises either in this case as it can be shown that the difference of the signal delays from the respective SFN sites are always within the safety limits.

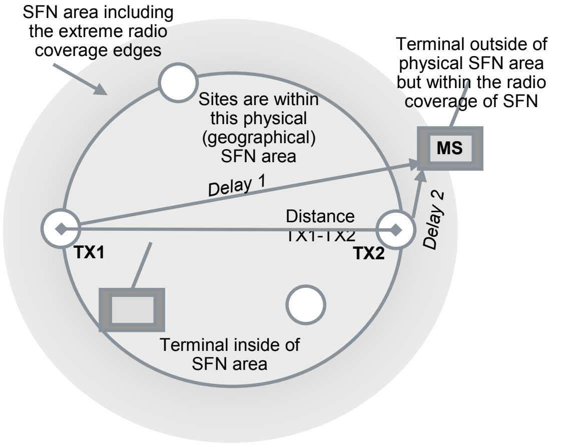

SFN area including the extreme radio coverage edges
Terminal outside of physical SFN area but within the radio coverage of SFN
Sites are within this physical (geographical) SFN area
MS
Delay 1
Delay 2
Distance TX1-TX2
TX1
TX2
Terminal inside of SFN area

Figure 3. The GI applies also outside the physical SFN cell area where the signal level originated form the SFN is sufficiently high.

The situation changes if the inter-site distance exceeds the allowed theoretical value. As an example, GI of 1/4 and 8K mode provide 224 μs margin for the safe propagation delay. Assuming the signal propagates with the speed of light, the SFN size limit is 300,000 km/s · 224 μs yielding about 67 km of maximum distance between the sites. If any geographical combination of the site locations using the same frequency exceeds this maximum allowed distance, they start producing interference in those spots where the difference of the arriving signals is higher than 224 μs.

If the level of interference is greater than the noise floor, and the minimum $C/N$ value that the respective mode required in non-interfered situation is not any more obtained, the signal in that specific spot is interfered and the reception suffers from the frame errors that disturb the fluent following of the contents. In order to achieve correct reception, the additional interference increases the required received power level of the carrier to $C/N \rightarrow C / (N + I)$.

Figure 4 shows that if the $D_{eff}$, i.e. the difference between the signals arriving from the sites, is more than the allowed safety distance in over-sized SFN, the site acts as an interferer. Whilst the carrier per interference and noise level from the $TX_2$ complies with the minimum requirement for the $C/N$, the transmission is still useful.

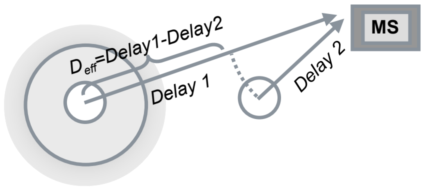


Figure 4. When the mobile station is receiving signals from the over-sized SFN-network, the sum of the interfering signals might destroy the reception if their level is sufficiently high compared to the sum of the useful carrier levels.

Even if the $C$ / ($N$ + $I$) level gets lower when the terminal moves from one site to another, the situation is not necessarily critical as the effective distance $D_{eff}$ of the signals might be within the SFN limits e.g. in the middle of two sites, although their distance from each others would be greater than the maximum allowed. In other words, the otherwise interfering site might not be considered as interference in the respective spot but it might give SFN gain by producing additional carrier $C_2$. This phenomenon can be observed in practice as the SFN interferences tends to accumulate primarily in the outer boundaries of the network.

Figure 5 shows the principle of the relative interference which increases especially when the terminal moves away from the centre of the SFN network.

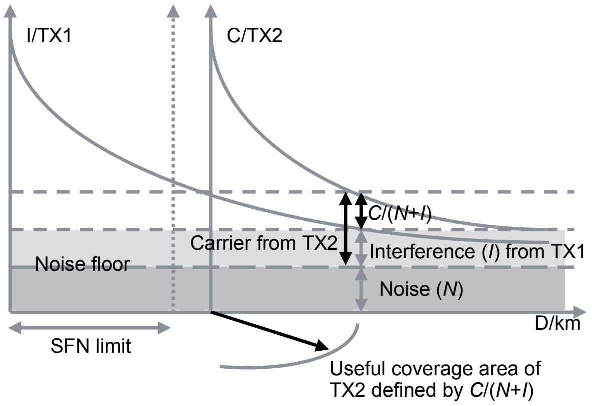

I/TX1
C/TX2
C/(N+I)
Carrier from TX2
Interference (I) from TX1
Noise floor
Noise (N)
D/km
SFN limit
Useful coverage area of TX2 defined by C/(N+I)

Figure 5. The principle of interference when the location of transmitter $TX_1$ is out of the SFN limit.

When moving outside of the network, the relative difference between the carrier and interfering signal gets smaller and it is thus inevitable that the $C$ / ($N$ + $I$) will not be sufficient any more at some point for the correct reception of the carrier, although the $C/N$ level without the presence of interfering signal would still be sufficiently high. The essential question is thus, where the critical points are found with lower $C$ / ($N$ + $I$)

value than the original requirement for $C/N$ is, and where the interference thus converts active, i.e. when $D_{eff}$ is longer than the safety margin.

As an example, the distance of two sites could be 70 km, which is more than $D_{sfn}$ with any of the radio parameter combination of DVB-H. For the parameter set of FFT 8k and GI 1/8, the safety distance for $D_{sfn}$ is about 34 km, which is clearly less than the distance of these sites. Let's define the radiating power (EIRP) for each site to +60 dBm. We can now observe the received power level of the sites in the theoretical open area by applying the free space loss, $f$ representing the frequency (MHz) and $d$ the distance (km):

$$L = 20\log f + 20\log d + 32.44 \qquad (4)$$

Figure 6 shows the carrier (or interference) from $TX_1$ located in 0 km and carrier (or interference) from $TX_2$ located in 70 km, when the parameter set allows $D_{sfn}$ of 34 km. The interference is included in those spots where the $D_{eff}$ is higher than $D_{sfn}$. If the $D_{eff}$ is shorter than $D_{sfn}$, the respective received useful power level is shown taking into account the SFN gain of these two sites by summing the absolute values of the power levels:

$$C_{tot} = \sqrt{C_1^2 + C_2^2} \qquad (5)$$

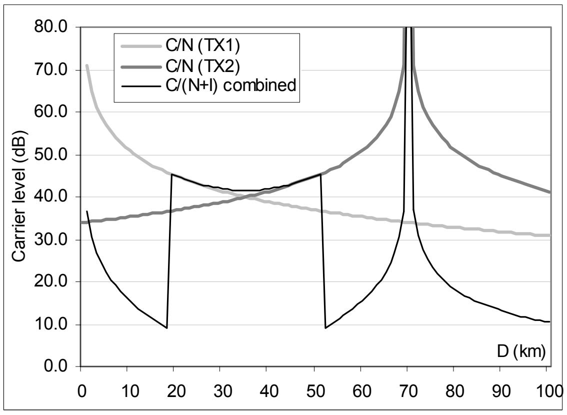


Figure 6. The combined $C/(N+I)$ along the route from 0 to 100 km, taking into account the SFN-gain when the interference is not present.

As Figure 6 shows, the $TX_1$ is acting as a carrier and $TX_2$ as interferer from 0 km ($TX_1$ location) to 18 km, because the $D_{eff} > D_{sfn}$. Nevertheless, the carrier of $TX_1$ is dominating within this area in order to provide sufficiently high $C / (N + I)$ for the successful reception for the QPSK, CR 1/2 and MPE-FEC 1/2 as it requires 8.5 dB. The segment of 18 km to 52 km is clear from the interferences as all the $D_{eff} < D_{sfn}$, and in addition, the receiver gets SFN gain from the combined carriers of $TX_1$ and $TX_2$. The $TX_1$ starts to act as an interferer from 52 km to 100 km (or, until the $C/N$ limit of the used mode). Nevertheless, the interference of $TX_1$ is already so attenuated such a far away from its origin that the $C / (N + I)$ is high enough for the successful reception from $TX_2$ of above mentioned QPSK still within the area of 80 – 100km. With any other parameter settings, the SFN interference level is high enough to affect on the successful reception in these breaking points where the $D_{eff}$ makes the signal act as interferer instead of carrier.

As can be seen from this example, the interference level takes place when the terminal moves towards the boundaries or boundary sites of the network. As a result, the boundary site's coverage area gets smaller, and depending on the parameter setting, there will be interferences between the sites.

The required $C/N$ for some of the most commonly used parameter setting can be seen in Table 3 [2]. The terminal antenna gain (loss) is taken into account in the presented values. The Table present the expected $C/N$ values in Mobile TU-channel (typical urban) for the "possible" reference receiver.

Table 3. The minimum $C/N$ (dB) for the selected parameter settings.

| Parameters | *C/N* |
|---|---|
| QPSK, CR 1/2, MPE-FEC 1/2 | 8.5 |
| QPSK, CR 1/2, MPE-FEC 2/3 | 11.5 |
| 16-QAM, CR 1/2, MPE-FEC 1/2 | 14.5 |
| 16-QAM, CR 1/2, MPE-FEC 2/3 | 17.5 |

The FFT size has impact on the maximum velocity of the terminal, and the GI affects on both the maximum velocity as well as on the capacity of the radio interface. In fact, in these simulations, if only the requirement for the level of carrier is considered without the need to take into account the maximum functional velocity of the terminal or the radio channel capacity, the following parameter combinations results the same $C/N$ and $C / (N + I)$ performance due to their same requirement for the safety distances:

- FFT 8K, GI 1/4: only one set
- FFT 8K, GI 1/8: same as FFT 4K, GI 1/4
- FFT 8K, GI 1/16: same as FFT 4K, GI 1/8 and FFT 2K, GI 1/4
- FFT 8K, GI 1/32: same as FFT 4K, GI 1/16 and FFT 2K, GI 1/8
- FFT 4K, GI 1/32: same as FFT 2K, GI 1/16
- FFT 2K, GI 1/32: only one set

## 4. Methodology for the SFN simulations: first variation (unlimited SFN network)

### 4.1. General

In order to estimate the error level of various sites that is caused by extending the theoretical geometrical limits of SFN network, a simulation can be carried out as presented in [6]. For the simulation, the investigated variables can be e.g. the antenna height and power level of the transmitter, in addition to the GI and FFT mode that defines the SFN limits.

The setup for the simulation consists of radio propagation type and geometrical area where the cells are located. The most logical way is to dimension the network according to the radio interface parameters, i.e. the cell radius should be dimensioned according to the minimum *C*/*N* requirement.

For this, a link budget calculator is included to the initial part of the simulator. It estimates the radius for both useful carriers as well as for the interfering signals, noise level being the reference.

Depending on the site definitions, there might be need to apply other propagation models as the basic Okumura-Hata [3] is valid for the maximum cell radius of 20 km and antenna heights up to 200 m. One of the suitable models for the large cells is ITU-R P.1546 [7] which is based on the interpolation of the pre-calculated curves.

When estimating the total carrier per interference levels, both total level of the carriers and interferences can be calculated separately by the following formulas, using the respective absolute power levels (W) for the *C* and *I* components:

$$C_{tot} = \sqrt{C_1^2 + C_2^2 + ... + C_n^2} \tag{6}$$

$$I_{tot} = \sqrt{I_1^2 + I_2^2 + ... + I_n^2} \tag{7}$$

In each simulation round, the site with the highest field strength is identified. In case of uniform network and equal site configurations, the site with lowest propagation loss corresponds to the nearest cell TX$_1$ which is selected as a reference. Once the nearest cell is identified, the task is to investigate the propagation delays of signals between the nearest and each one of the other sites, and calculate if the difference of arriving signals $D_{eff}$ is greater or lower than the SFN limit $D_{sfn}$. In general, if the difference of the signal arrival times of TX$_1$ and TX$_n$ is greater than GI defines, the TX$_n$ is producing interfering signal (if the signal is above the noise floor), and otherwise it is adding the level of total carrier energy (if the signal level is above the minimum requirement for carrier).

In order to obtain the level of *C* and *I* in certain area type, the path loss can be estimated with Okumura-Hata radio propagation model or ITU-R P.1546.

The total path loss can be calculated by applying the following formula:

$$L_{tot} = L_{pathloss} + L_{norm} + L_{other} \tag{8}$$

$L_{norm}$ represents the fading loss caused by the long-term variations, and other losses may include e.g. the fast fading as well as antenna losses.

For the long-term fading, a normal distribution is commonly used in order for modelling the variations of the signal level. The PDF of the long-term fading is the following [8]:

$$PDF(L_{norm}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[\frac{-\left(x-\bar{x}\right)^2}{2\sigma^2}\right] \tag{9}$$

The term $x$ represents the loss value, and $\bar{x}$ is the average loss (0 in this case). In the snap-shot based simulations, the $L_{norm}$ is calculated for each arriving signal individually as the different events does not have correlation. The respective PDF and CDF are obtained by creating a probability table for normal distributions. Figure 7 shows an example of the PDF and CDF of normal distributed loss variations when the mean value is 0 and standard deviation is 5.5 dB.

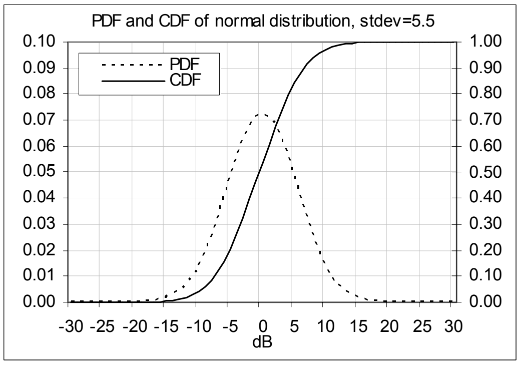


Figure 7. PDF and CDF of the normal distribution representing the variations of long-term loss when the standard deviation is set to 5.5 dB.

The fast (Rayleigh) fading is present in those environments where multi-path radio signals occurs, e.g., on the street level of cities. It can be presented with the following PDF:

$$L_{\log norm} = \frac{x}{\delta^2} e^{-(\frac{x^2}{2\delta^2})} \tag{10}$$

Figure 8 shows the PDF and CDF of the fast fading representing the variations of short-term loss when the standard deviation is set to 5.5 dB.

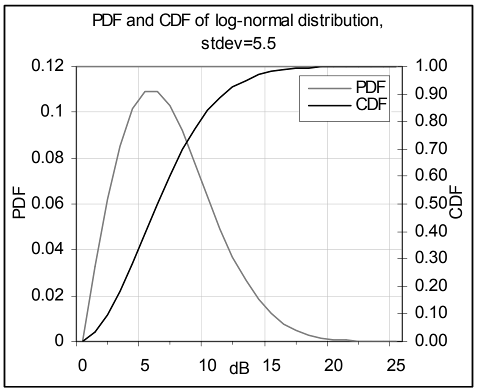


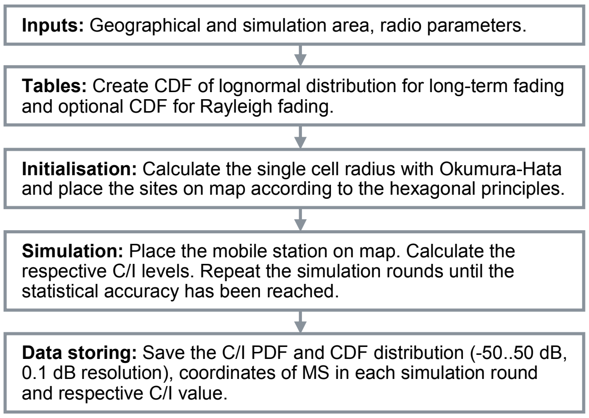
Figure 8. PDF and CDF of the log-normal distribution for fast fading.

### 4.2. Simulator

A block diagram of the presented SFN interference simulator is shown in Figure 9. The simulator was programmed with a standard Pascal code. It produces the results to text files, containing the *C*/*N*, *I*/*N* and *C* / (*N* + *I*) values showing the distribution in scale of -50...+50 dB and with 0.1 dB resolution, using integer type table indexes of -500 to +500 that represents the occurred cumulative values. Also the terminal coordinates and respective *C*/*N*, *I*/*N* and *C* / (*N* + *I*) for all the simulation rounds is produced. If the value occurs outside the scale, it is added to the extreme dB categories in order to form the CDF correctly.

A total of 60,000 simulation rounds per each case were carried out. It corresponds to an average of 60,000 / (50 dB · 10) = 120 samples per *C*/*I* resolution, which fulfils the accuracy of the binomial distribution. Each text file was post-processed and analysed with Microsoft Excel.

The terminal was placed randomly in 100 km × 100 km area according to the uniform distribution in function of the coordinates (*x*, *y*) during each simulation round. The raster of the area was set to 10 m. Small and medium city area type was selected for the simulations. The total *C*/*I* value is calculated per simulation round by observing the individual signals of the sites.



**Inputs:** Geographical and simulation area, radio parameters.

**Tables:** Create CDF of lognormal distribution for long-term fading and optional CDF for Rayleigh fading.

**Initialisation:** Calculate the single cell radius with Okumura-Hata and place the sites on map according to the hexagonal principles.

**Simulation:** Place the mobile station on map. Calculate the respective C/I levels. Repeat the simulation rounds until the statistical accuracy has been reached.

**Data storing:** Save the C/I PDF and CDF distribution (-50..50 dB, 0.1 dB resolution), coordinates of MS in each simulation round and respective C/I value.

Figure 9. The simulator's block diagram.

The nearest site is selected as a reference during the respective simulation round. If the arrival time delays difference $\Delta t_2$–$\Delta t_1$ is less than $D_{sfn}$ defines, the respective signal is marked as useful carrier *C*, or otherwise it is marked as interference *I*. In the generic format, the total *C* / (*N* + *I*) can be obtained from the simulation results in the following way:

$$\frac{C}{N+I} = \left(C_{tot}[dB] - noisefloor[dB]\right) - \left(I_{tot}[dB] - noisefloor[dB]\right) \tag{11}$$

The term *N* represents the reference which is the sum of noise floor and terminal noise figure. The noise figure depends on the terminal characteristics. In the simulations, it was estimated to 5 dB as defined in [2].

The simulator calculates the expected radius of single cell in non-interfering case and fills the area with uniform cells according to the hexagonal model. This provides partial overlapping of the cells. Each simulation round provides information if that specific connection is useless, e.g. if the criteria set of 1) effective distance $D_{eff} > D_{sfn}$ in any of the cells, and 2) *C* / (*N* + *I*) < minimum *C*/*N* threshold. If both criteria are valid, and if the *C*/*N* would have been sufficiently high without the interference in that specific round, the SFN interference level is calculated.

Figure 10 shows an example of the site locations. As can be seen, the simulator calculates the optimal cell radius according to the parameter setting and locates the transmitters on map according to the hexagonal model, leaving ideal overlapping areas in the cell border areas. The size and thus the number of the cells depends on the radio parameter settings without interferences, and in each case, a result is a uniform service level in the whole investigated area. The same network setup is used throughout the complete simulation, and changed if the radio parameters of the following simulation require so.

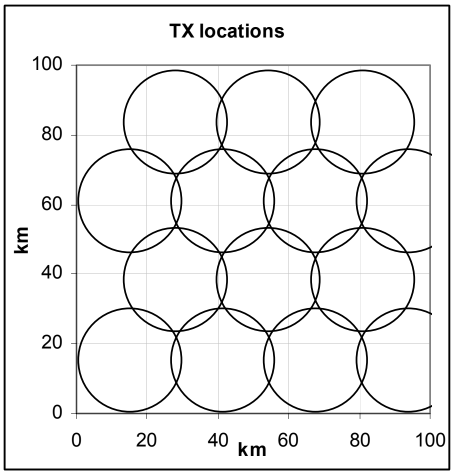


Figure 10. Example of the transmitter site locations the simulator has generated.

The behaviour of *C* and *I* can be investigated by observing the probability density functions, i.e. PDF of the results. Nevertheless, the specific values of the interference levels can be obtained by producing a CDF from the simulation results.

Figure 11 shows two examples of the simulation results in CDF format. In this specific case, the outage probability of 10% (i.e. area location probability of 90 %) yields the minimum $C$ / $(N + I)$ of 10 dB for 8K, which complies with the original $C/N$ requirement (8.5 dB) of this case. On the other hand, the 4K mode results about 7 dB with 10% outage, which means that the minimum quality targets can not quite be achieved any more with these settings.

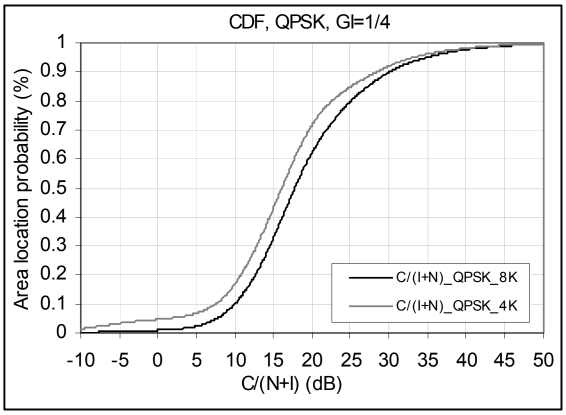


Figure 11. Example of the cumulative distribution of $C/(N{+}I)$ for QPSK 4K and 8K modes with antenna height of 200 m and Ptx +60 dBm.

### 4.3. Results

By applying the principles of the DVB-H simulator, the $C/I$ distribution was obtained according to the selected radio parameters. The variables were the modulation scheme (QPSK and 16-QAM), antenna height (20-200 m) and FFT mode (4K and 8K).

Figures 12-13 show the resulting networks that were used as a basis for the simulations. The simulator selects randomly the mobile terminal location on the map and calculates the $C/I$ that the network produces at that specific location and moment. This procedure is repeated during 60,000 simulation rounds. One of the results after the complete simulation is the estimation for the occurred errors due to the interfering signals from the sites exceeding the safety distance (i.e. if the arrival times of the signals exceed the maximum allowed delay difference). This event can be called “SFN error rate”, or SER.

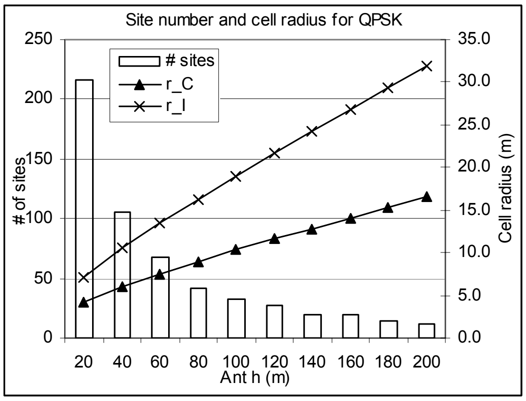


Figure 12. The network dimensions for the QPSK simulations.

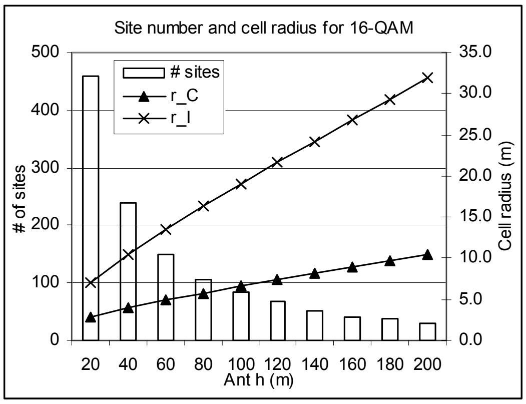


Figure 13. The network dimensions for the 16-QAM simulations.

In Figure 14, the plots indicates the locations where the results of $C / (N + I)$ corresponds 8.5 dB or less for QPSK. In this case, the interfering plots represent the relative SFN area error rate (SAER) of 0.83%, i.e. the erroneous (SAR) cases over the number of total simulation rounds as for the simulated plots.

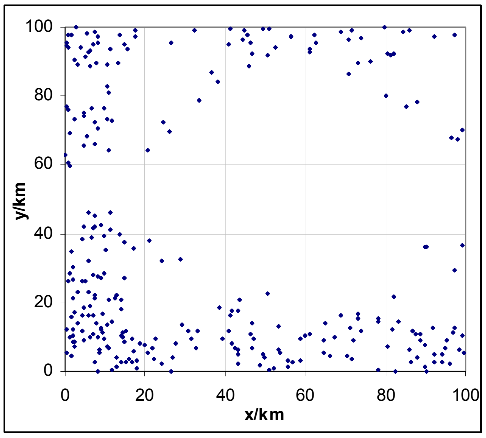


Figure 14. An example of the results in geographical format with $C/(N+I)$<8.5 dB.

It can be assumed that when the SER level is sufficiently low, the end-users will not experience remarkable reduction in the DVB-H reception due to the extended SFN limits. In this analysis, a SER level of 5% is assumed to still provide with sufficient performance as it is in align with the limits defined in [2] for frame error rate before the MPE-FEC (FER) and frame error rate after the MPE-FEC (MFER). The nature of the SER is slightly different, though, as the interferences tend to cumulate to certain locations as can be observed in Figure 14 obtained from the simulator.

According to simulations, the SFN interference level varies clearly when radio parameters are tuned. Figures 15-18 summarise the respective SFN area error analysis, the variable being the transmitter antenna height. Figures show that with the uniform radio parameters and varying the antenna height, modulation and FFT mode, the functional settings can be found regardless of the exceeding of the theoretical SFN limits.

If a 5% limit for SER is accepted, the analysis show that antenna height of about 80 m or lower produces SER of 5% or less for QPSK, 8K, with minimum $C / (N + I)$ requirement of 8.5 dB. If the mode is changed to 4K, the antenna height should be lowered to 35-40 m from ground level in order to comply with 5% SER criteria in this very case. 16-PSK produces higher capacity and smaller coverage.

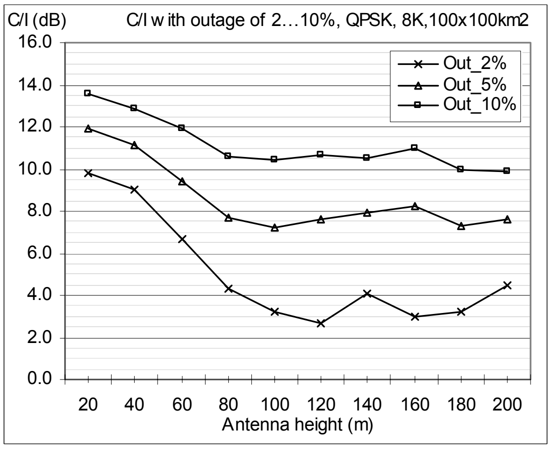


Figure 15. The summary of the case 1 (QPSK, 8K). The results show the $C/(N+I)$ with 2%, 5% and 10 % SER criteria.

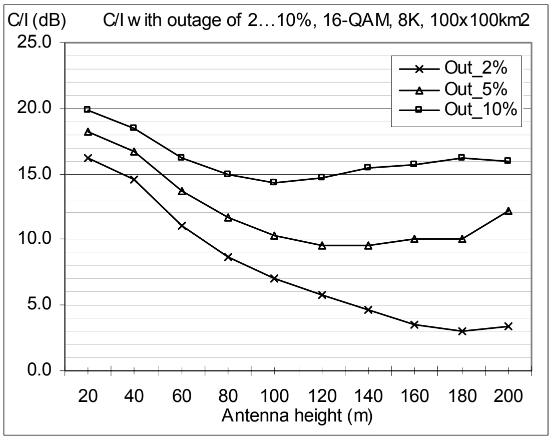


Figure 16. The summary of the case 2 (16-QAM, 8K).

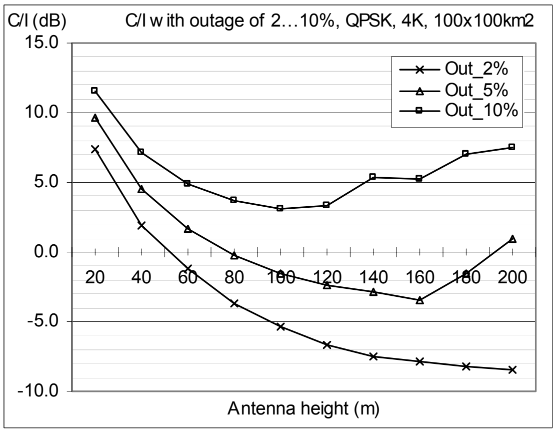


Figure 17. The summary of the case 3 (QPSK, 4K).

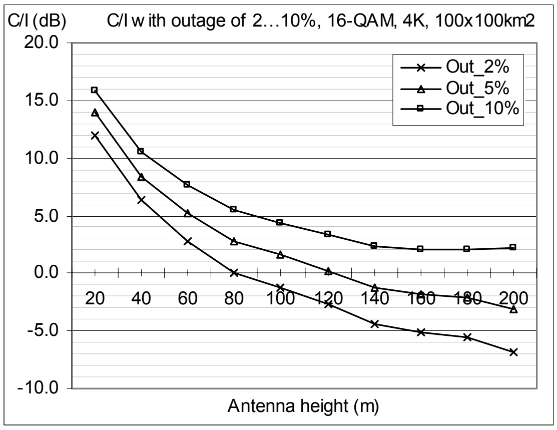


Figure 18. The summary of the case 4 (16-QAM, 4K).

The results show this clearly as the respective SER of 5% (16-QAM, 8K and minimum *C* / (*N* + *I*) requirement of 14.5 dB for this modulation) allows the use of antenna height of about 120 m. If the mode of this case is switched to 4K, the antenna should be lowered down to 50 m in order to still fulfil the SER 5% criteria.

The +60 dBm EIRP represents relatively low power. The higher power level raises the SER level accordingly. For the mid and high power sites the optimal setting depends thus even more on the combination of the power level and antenna height. According to these results, it is clear that the FFT mode 8K is the only reasonable option when the SER should be kept in acceptable level. Especially the QPSK modulation might not allow easily extension of SFN as the modulation provides largest coverage areas. On the other hand, when providing more capacity, 16-QAM is the most logical solution as it gives normally sufficient capacity with reasonable coverage areas. The stronger CR and MPE-FEC error correction rate decreases the coverage area but it is worth noting that the interference propagates equally also in those cases.

The general problem of the SER arises from the different loss behaviour of the useful carrier and interfering signal. Depending on the case, the interfering signal might propagate 2-3 times further away from the originating site compared to the useful carrier as can be seen from Figures 12 and 13.

In practice, the SER level can be further decreased by minimising the propagation of the interfering components. This can be done e.g. by adjusting the transmitter antenna down-tilting and using narrow vertical beam widths, producing thus the coverage area of the carrier and interference as close to each others as possible. Also the natural obstacles of the environment can be used efficiently for limiting the interferences far away outside the cell range.

Figure 19 shows the previously presented results presenting the outage percentage for the different modes having 8.5 dB C */ (N + I)* limit for QPSK and 14.5 % for 16-QAM cases in function of transmitter antenna height.

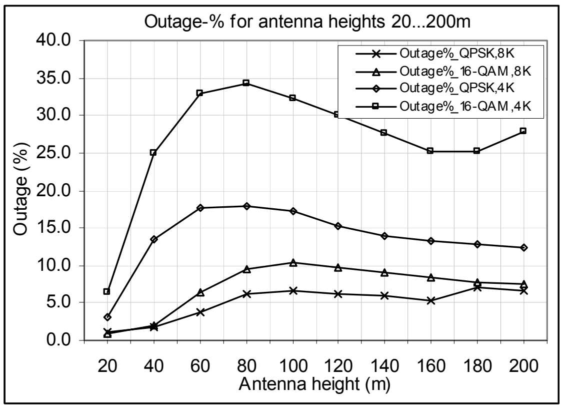


Figure 19. The summary of the cases 1-4 presenting the outage percentage in function of the transmitter antenna height.

This version of the simulator gives indication about the behaviour of the *C* / (*N* + *I*) in geographical area. In order to estimate the SFN gain, the individual cells could be switched on and off for the comparison of the differences in overall *C* / (*N* + *I*) distribution. Nevertheless, when the investigated area is filled with the cells, it normally leaves outages in the northern and eastern sides as the area cannot be filled completely as shown in Figure 10. It also produces partial cell areas, if the centre of the site fits into the area but the edge is outside. An enhanced version of the simulator was thus developed in order to investigate the SFN gain in more controlled way, i.e. instead of the fixed area size the method uses the variable reuse pattern sizes. The following Chapter 6 describes the method.

## 5. Methodology for the simulations: second variation (SFN network with fixed reuse patterns)

### 5.1. Simulator

The second version of the presented SFN performance simulator is based on the hexagonal cell layout [1]. Figure 20 presents the basic idea of the cell distribution.

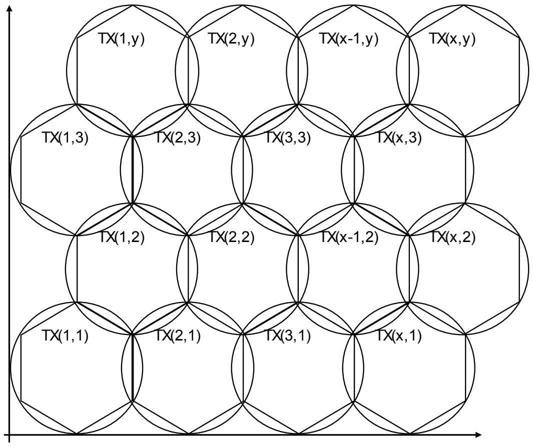


Figure 20. The active transmitter sites are selected from the 2-dimensional cell matrix with the individual numbers of the sites.

As can be seen from Figure 20, the cells are located in such way that they create ideal overlapping areas. The tightly located hexagonal cells fill completely the circle-shape cells. A uniform parameter set is used in each cell, including the transmitter power level and antenna height, yielding the same radius for each cell per simulation case.

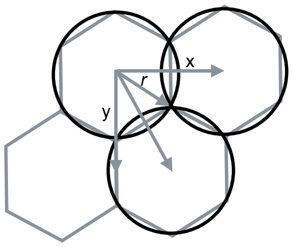


Figure 21. The *x* and *y* coordinates for the calculation of the site locations.

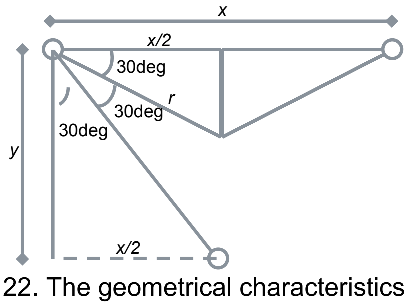


Figure 22. The geometrical characteristics of the hexagonal model used in the simulator.

As the relative location of the cells is fixed, the coordinates of each cell depends on the uniform cell size, i.e. on the radius. Taking into account the characteristics of the hexagonal model, the *x* coordinates can be obtained in the following way depending if the row for *y* coordinates is odd or even.

The distance between two sites in *x*-axis is:

$$x = 2r\cos(30°) = 2 \cdot 0.866r \tag{12}$$

The common inter-site distance in *y*-axis is:

$$y = \frac{r\cos(30°)}{\tan(30°)} = \frac{2r}{3} \tag{13}$$

For the odd rows the formula for the *x*-coordinate of the site *m* is thus the following:

$$x(m)_{odd} = r + (m-1) \cdot 1.732r \tag{14}$$

In the formula, *m* represents the number of the cell in *x*-axis. In the same manner, the formula for *x*-coordinates can be created in the following way:

$$x(m)_{even} = r + \frac{1}{2}1.732r + (m-1) \cdot 1.732r \tag{15}$$

For the *y* coordinates, the formula is the following:

$$y(n) = r + (n-1) \cdot \frac{2}{3}r \tag{16}$$

The simulations can be carried out for different cell layouts. Symmetrical reuse pattern concept was selected for the simulations presented in this paper. The most meaningful reuse pattern size *K* can be obtained with the following formula [9]:

$$K = (k-l)^2 - kl \tag{17}$$

The variables *k* and *l* are positive integers with minimum value of 0. In the simulations, the reuse pattern sizes of 1, 3, 4, 7, 9, 12, 16, 19 and 21 was used for the *C* / (*N* + *I*) distribution in order to obtain the carrier and interference distribution in both non-interfering and interfering networks (i.e. SER either exists or not depending on the size of the SFN area). In this way, the lower values of *K* provides with the non-interfering SFN network until a limit that depends on the GI and FFT size parameters.

The single cell (*K*=1) is considered as a reference in all of the cases. The fixed parameter set was the following:

- Transmitter power: 60 dBm
- Transmitter antenna height: 60 m

- Receiver antenna height: 1.5 m
- Long-term fading with normal distribution and standard deviation of 5.5 dB
- Area coverage probability in the cell edge: 70%
- Receiver noise figure: 5 dB
- Bandwidth: 8 MHz
- Frequency: 700 MHz

For the used bandwidth, the combined noise floor and noise figure yields -100.2 dBm as a reference for calculating the level of *C* and *I*. The path loss was calculated with Okumura-Hata prediction model for small and medium sized city. The 70 % area coverage probability corresponds with 10% outage probability in the single cell area.

These settings result a reference *C*/*N* of 8.5 dB for QPSK and 14.5 dB for 16-QAM. The value is the minimum acceptable *C*/*N*, or in case of interferences, *C* / (*N* + *I*) value that is needed for the successful reception of the signal.

Figures 23 and 24 present the symmetrical reuse patterns that were selected for the simulations. The grey hexagonal means that the coordinates has been taken into account calculating the order number of the sites according to the formulas 14-16, but the respective transmitter has been switched off in order to form the correct reuse pattern.

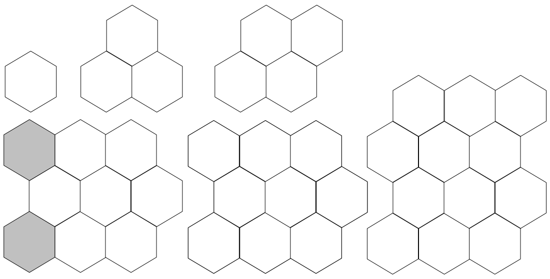


Figure 23. The reuse patterns with *K* of 1, 3, 4, 7, 9 and 12.

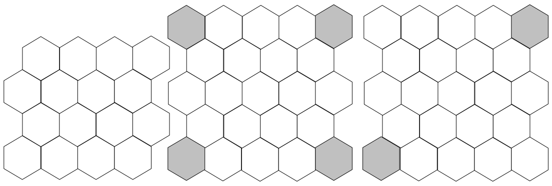


Figure 24. The reuse patterns with *K* of 16, 19 and 21.

Figure 25 shows the site locations for the QPSK and *K*=7, and Figure 26 shows an example of the *C*/*N* distribution with the parameter values of *K*=7, GI=1/4, and FFT=8K.

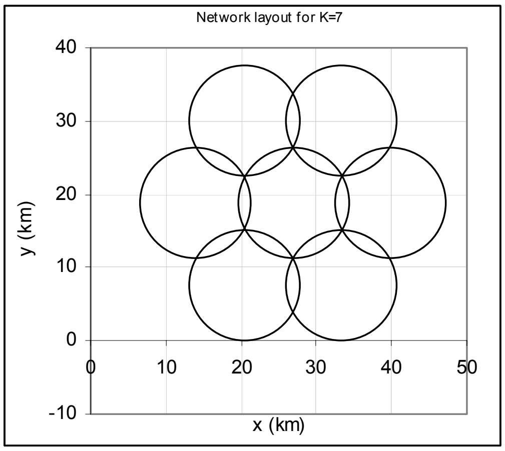


Figure 25. An example showing the layout of the QPSK network with *K*=7.

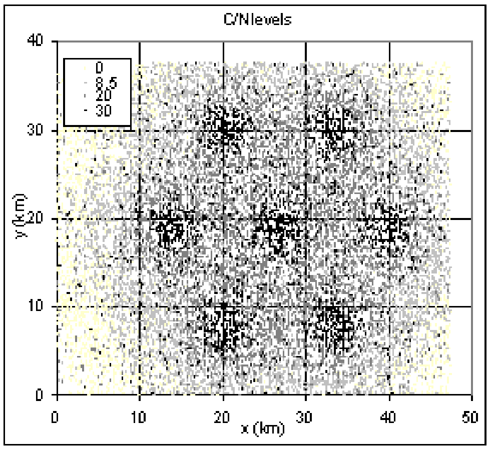


Figure 26. An example of the simulated case with QPSK and *K*=7.

According to the *C*/*I* link analysis, the case presented in Figure 26 is free of SFN interferences.

The actual simulation results for *C*/*N*, or in case of the interferences, for *C* / (*N* + *I*), is done in such way that only the terminal locations inside the calculated cell areas are taken into account. If the terminal is found outside of the network area (the circles) in some simulation round, the result is simply rejected.

Figure 27 shows the principle of the filtered simulation. As the terminal is always inside the coverage area of at least one cell, it gives the most accurate estimation of the SFN gain with different parameter values. Furthermore, the method provides a reliable means to locate the MS inside the network area according to the uniform distribution.

The network is dimensioned in such way that the area location probability is 70% in the cell edge. The dimensioning can be made according to the characteristics of long-term fading.

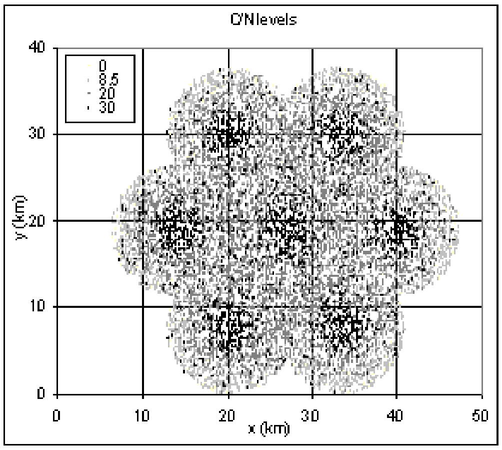


Figure 27. The filtered simulation area. This principle is used in the simulations in order to keep the network borders always constant. If the mobile station is inside the planned network area, it provides a reliable estimation of the SFN gain.

Figure 28 shows snap-shot type example of the *C/N* values with less than 8.5 dB, which is the limit for the respective parameter settings of QPSK cases.

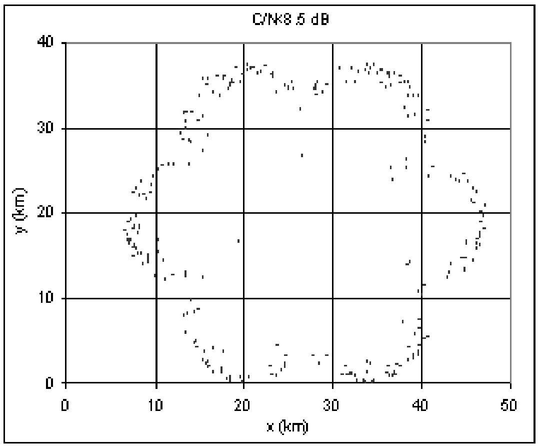


Figure 28. An example of the distribution of the simulation results that yields less than 8.5 dB for the *C/N*.

As a verification of the geographical interference class, a study that can be called a *C/I*-link analysis can be carried out. It is a method to revise all the combinations (hash) of the distances between each pair of sites ($TX_1$-$TX_2$, $TX_1$-$TX_3$, $TX_2$-$TX_1$, $TX_2$-$TX_3$ etc.) marking the link as useful (*C*) if the guard distance between the respective sites is less than the maximum allowed SFN diameter ($D_{sfn}$). If the link is longer, it is marked as a potential source of interference (*I*). The interference link proportion can be obtained for each case by calculating the interference links over the total links. It gives a rough idea about the "severity" of the exceeding of the SFN limit, with a value range of 0-100% (from non-interfering network up to interfered network where all the transmitters are a potential source of interference).

## 5.2. Results

Tables 4 and 5 summarises the *C/I* link analysis for the different reuse pattern sizes and for FFT and GI parameter values. The values presents the percentage of the over-sized legs of distances between the cell sites compared to the amount of all the legs.

Table 4. The *C/I* link analysis for QPSK cases.

| | Reuse pattern size (*K*) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ***FFT,GI*** | ***3*** | ***4*** | ***7*** | ***9*** | ***12*** | ***16*** | ***19*** | ***21*** |
| 8K, 1/4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.5 |
| 4K, 1/4 | 0 | 0 | 0 | 11.1 | 24.2 | 36.7 | 42.1 | 47.1 |
| 2K, 1/4 | 0 | 16.7 | 42.9 | 55.6 | 65.2 | 72.5 | 75.4 | 78.1 |
| 8K, 1/8 | 0 | 0 | 0 | 11.1 | 24.2 | 36.7 | 42.1 | 47.1 |
| 4K, 1/8 | 0 | 16.7 | 42.9 | 55.6 | 65.2 | 72.5 | 75.4 | 78.1 |
| 2K, 1/8 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 8K, 1/16 | 0 | 16.7 | 42.9 | 55.6 | 65.2 | 72.5 | 75.4 | 78.1 |
| 4K, 1/16 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 2K, 1/16 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 8K, 1/32 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 4K, 1/32 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 2K, 1/32 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

The *C/I* link analysis shows that in case of large network (21 cells in the SFN area), the only reasonable parameter set for the QPSK modulation seems to be FFT=8K and GI=1/4. This is due to the fact that QPSK provides with the largest cell sizes (with the investigated parameter set the *r* is 7.5 km). The cell size of the investigated 16-QAM case is smaller (*r*=5.0 km) which provides the use of the parameter set of (FFT = 8K, GI = 1/4), (FFT = 4K, GI = 1/4) and (FFT = 8K, GI = 1/8). The interference distance $r_{interference}$ = 13.5 km is the same in all the cases as the interference affects until it reaches the reference level (the sum of noise floor and terminal noise figure).

The *C/I* link investigation gives thus a rough idea about the most feasible parameter settings. In order to obtain the information about the complete performance of DVB-H, the combination of the SFN gain and SER level should be investigated as shown next.

Table 5. The C/I link analysis for 16-QAM cases.

| | Reuse pattern size (*K*) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ***FFT,GI*** | ***3*** | ***4*** | ***7*** | ***9*** | ***12*** | ***16*** | ***19*** | ***21*** |
| 8K, 1/4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4K, 1/4 | 0 | 0 | 0 | 0 | 0 | 0.8 | 1.8 | 5.7 |
| 2K, 1/4 | 0 | 0 | 14.3 | 30.6 | 42.4 | 49.1 | 57.9 | 61.9 |
| 8K, 1/8 | 0 | 0 | 0 | 0 | 0 | 0.8 | 1.8 | 5.7 |
| 4K, 1/8 | 0 | 0 | 14.3 | 30.6 | 42.4 | 49.1 | 57.9 | 61.9 |
| 2K, 1/8 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 8K, 1/16 | 0 | 0 | 14.3 | 30.6 | 42.4 | 49.1 | 57.9 | 61.9 |
| 4K, 1/16 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 2K, 1/16 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 8K, 1/32 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 4K, 1/32 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 2K, 1/32 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

Figures 29 and 30 show examples of two extreme cases of the simulations, i.e. the PDF of non-interfered and completely interfered situation.

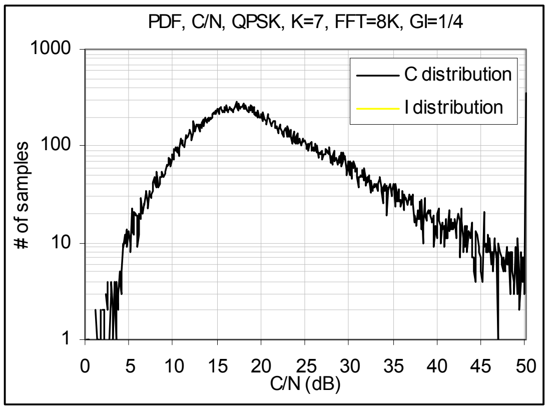


Figure 29. An example of the *C*/*N* distribution in non-interfered SFN network.

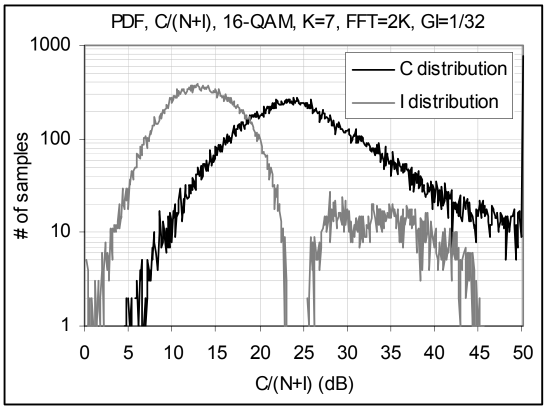


Figure 30. An example of *C*/*N* and *I*/*N* in interfered case. This parameter combination does not provide functional service in simulated area.

Figures 29 and 30 show two examples of the PDF, i.e. occurred amount of samples per *C*/*N* and *C*/*I* in scale of 0-50 dB, with 0.1 dB resolution.

The PDF gives a visual indication about the general quality of the network. Nevertheless, in order to obtain the exact values of the performance indicators, a cumulative presentation is needed. Figure 31 shows an example of the CDF in the non-interfering QPSK network with the reuse pattern size as a variable. The case shows the *C*/*N* for the parameter set of QPSK, GI 1/4 and FFT 8K. This mode is the most robust against the interferences as it provides with the longest guard distance.

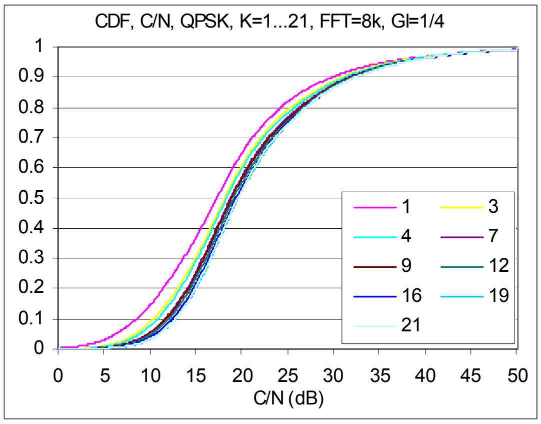


Figure 31. The CDF of C/N in non-interfered network for reuse pattern sizes of 1-21.

Figure 32 shows an amplified view to the critical point, i.e. to the 10% outage probability point.

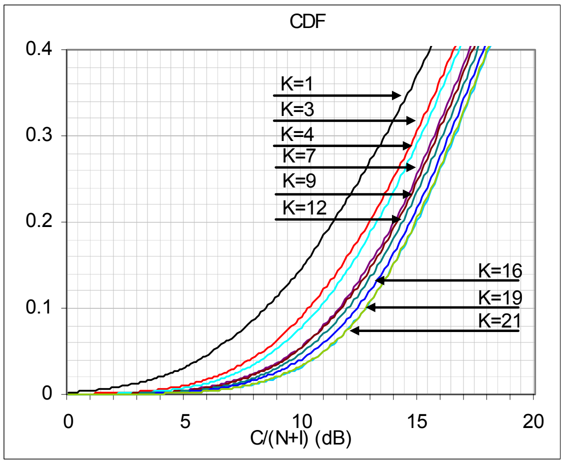


Figure 32. An amplified view of the example of the processed simulation results for QPSK.

As can be seen from Figure 32, the single cell (*K*=1) results a minimum of 8.5 dB for the 10 % outage probability, i.e. for the area location probability of 90% in the whole cell area which corresponds to the 70% area location probability in the cell edge. The cell is thus correctly dimensioned for the simulations.

In order to find the respective SFN gain level, the comparison with single cell and other reuse pattern sizes can be made in this 10% outage point. The following Figures 33 and 34 shows the respective simulation results for all the symmetrical reuse pattern sizes 1-21 for QPSK and 16-QAM.

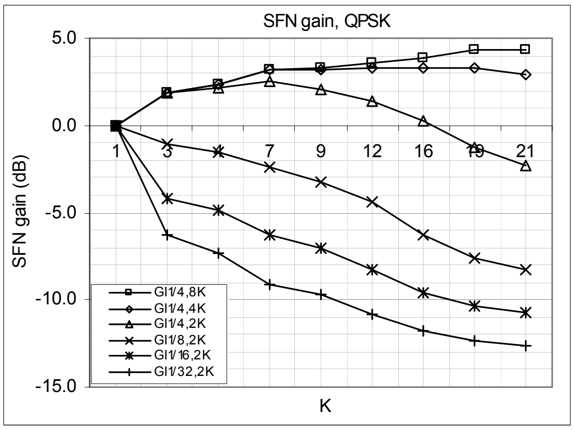


Figure 33. SFN gain levels for QPSK cases, with the reuse pattern sizes of 1-21.

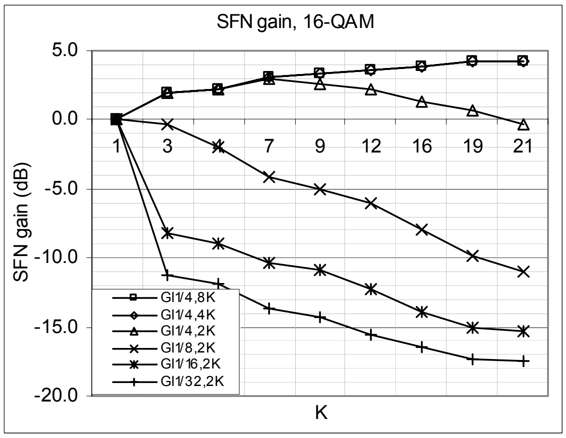


Figure 34. SFN gain levels for 16-QAM cases, with the reuse pattern sizes of 1-21.

The simulation results show the level of SFN gain. The reference case {FFT 8K and GI 1/4} results the maximum gain for the reuse pattern sizes of *K*=1-21 for both QPSK and 16-QAM and provides a non-interfering network. In addition, the parameter set of {FFT = 4K, GI = 1/4} and {FFT = 8K, GI = 1/8} results a network where SFN errors can be compensated with the SFN gain.

According to the results shown in Figure 33, the QPSK case could provide SFN gain of 3-4 dB in non-interfering network. It is interesting to note that in the interfering cases, also the parameter set of {GI = 1/4, FFT = 4K corresponding FFT = 8K, GI = 1/8} results positive SFN gain even with the interference present for all the reuse pattern cases up to 21. Also the parameter set of {GI = 1/4, FFT = 2K, corresponding FFT = 8K, GI = 1/16 and FFT = 4K, GI = 1/8} provides an adequate quality level until reuse pattern of 16 although the error level (SER) increases.

According to Figure 34, the 16-QAM gives equal SFN gain, resulting about 3-4 dB in non-interfering network. For the {FFT = 4K, GI = 1/4} and the corresponding parameter set of {FFT = 8K, GI = 1/8}, the SFN gain is higher than the SER even with higher reuse pattern sizes compared to QPSK, because the 16-QAM cell size is smaller.

As can be seen from Figures 33-34 and from the *C*/*I* link analysis of Tables 4-5, the rest of the cases are practically useless with the selected parameter set.

# 6. Methodology for the simulations: third variation (urban SFN network)

The dense and urban area of Mexico City was used as a basis for the next simulations as described in [11] by applying suitable propagation prediction models (Okumura-Hata and ITU-R P.1546-3).

## 6.1. Simulation environment

The city is located on relatively flat ground level with high mountains surrounding the centre area. The height of the planned DVB-H site antennas was 60, 190, 30, 20, 20, 30 and 60 meters from the tower base, respectively for the sites 1-7. The site number 7 represents the mountain installation with the tower base located 800 meters above the average ground level which results the effective antenna height of 860 meters compared to the city centre level. Site number 4 is also situated in relatively high level, but in this case, the surrounding area of the site limits its coverage area. The rest of the sites are located in the base level of Mexico City centre.

As the cell radius of the investigated sites is clearly smaller than 20 km, the Okumura-Hata [3] is suitable for the path loss prediction for all the other sites except for the mountain site number 7. Figure 38 shows the principle of the geographical profile of this site, varying the horizontal angle from the site to the centre by 10 degree steps. As the profile shows, there are smaller mountains found in front of the site.

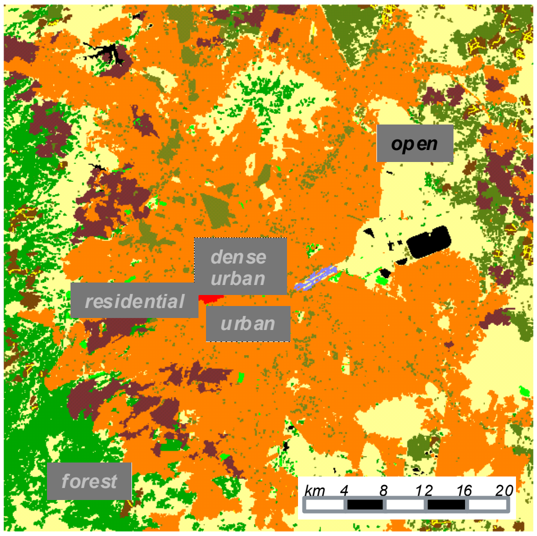


Figure 35. The clutter type of the investigated area.

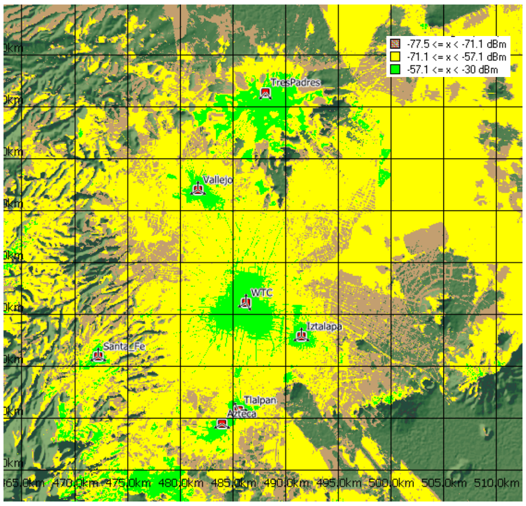


Figure 36. The predicted coverage area of the investigated network as analyzed with a separate radio network planning tool.

Figure 37 presents the location of the selected sites, and the Table 6 shows the site parameters.

For the site number 7, ITU-R P.1546 (version 3) [7] model was applied by using the antenna height of 860 meters and frequency of 680 MHz.

The calculation of the path loss for the site number 7 was done in practice by interpolating the correct ITU-R P.1546 curve for 860 meter antenna height (via 600 and 1200 meter heights) and for 680 MHz frequency (via 600 and 2000 MHz). Figure 39 shows the resulting curve after the iterations.

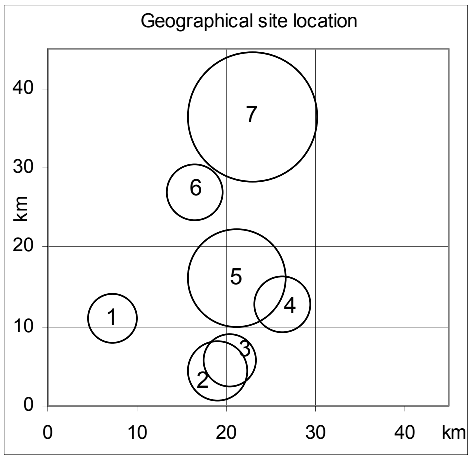


Figure 37. The site locations and informative relative site sizes of the simulator.

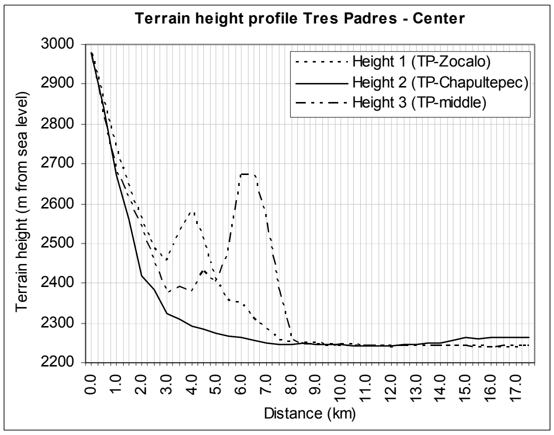


Figure 38. The profile of the mountain site 7.

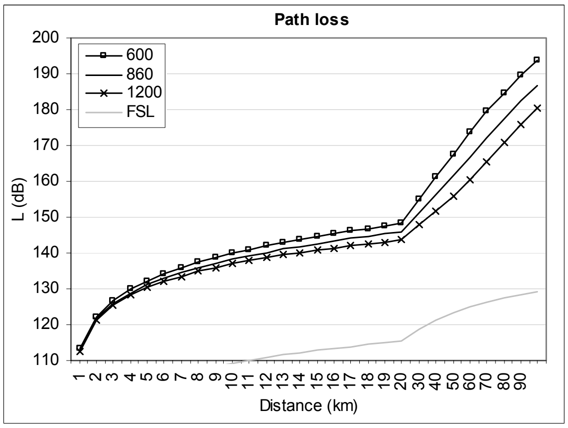


Figure 39. The estimated path loss *L* for the mountain site. FSL is free space loss reference.

Next, a trend line was created in order to present the tabulated values with a closed formula and to ease the simulations. For this specific case, there was one formula created for the path loss in distances of 1-20 km ($L_{20}$) and another one for the distances of 20-100 km ($L_{100}$), $d$ being the distance (km):

$$L_{20} = 10.659 \ln(d) + 113.84 \qquad (18)$$

$$L_{100} = 0.5124d + 135.55 \qquad (19)$$

In order to estimate the error between the trend lines and the original ITU-model, Figure 40 was produced. In the functional area of the mountain site, the maximum error is < 0.5 dB.

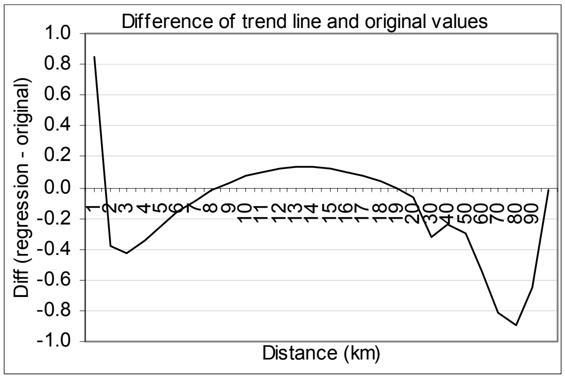


Figure 40. The estimated error margin for the trend lines used for the mountain site path loss calculation.

The link budget of the simulator takes into account separately the radiating power levels and antenna heights of each site as seen in Table 6.

Table 6. The site parameters.

| Site | *Coord., km* | | *EIRP* | | *Radius, km* | |
|---|---|---|---|---|---|---|
| nr | x | y | dBm | W | QPSK | 16-QAM |
| 1 | 7.2 | 11.0 | 70.5 | 11258 | 4.1 | 2.8 |
| 2 | 19.0 | 4.4 | 69.3 | 8481 | 5.6 | 3.8 |
| 3 | 20.5 | 5.8 | 69.3 | 8481 | 4.6 | 3.2 |
| 4 | 26.4 | 12.7 | 69.5 | 8860 | 5.0 | 3.4 |
| 5 | 21.2 | 16.1 | 71.3 | 13411 | 15.5 | 9.8 |
| 6 | 16.6 | 27.0 | 69.3 | 8481 | 5.0 | 3.4 |
| 7 | 23.0 | 36.4 | 71.1 | 12837 | 26.3 | 15.8 |

During the simulations, the receiver was placed randomly in the investigated area (45km × 45km = 2025 km$^2$) according to the snap-shot principle and uniform geographical distribution. In each simulation round, the separate sum of the carrier per noise and the interference per noise was calculated by converting the received power levels into absolute powers. The result gives thus information about the balance of SFN gain and SFN interference levels. Tables of geographical coordinates with the respective sum of carriers and interferences were created by repeating the simulations 60,000 times. Also carrier and interference level distribution tables were created with a scale of -50 … +50 dB.

The long-term as well as Rayleigh fading was taken into account in the simulations by using respective distribution tables independently for each simulation round. A value of 5.5 dB was used for the standard deviation. The area location probability in the cell edge of 90% was selected for the quality criteria, producing about 7 dB shadowing margin for the long-term fading. Terminal antenna gain of -7.3 dBi was used in the calculations according to the principles indicated in [3]. Terminal noise figure of 5 dB was taken into account. Both Code Rate and MPE-FEC Rate were set to 1/2.

### 6.2. Simulation results

The usable coverage area was investigated by post-processing the simulation results. The simulations were carried out by using QPSK and 16-QAM modulations, CR ½, MPE-FEC ½ and all the possible variations of FFT and GI.

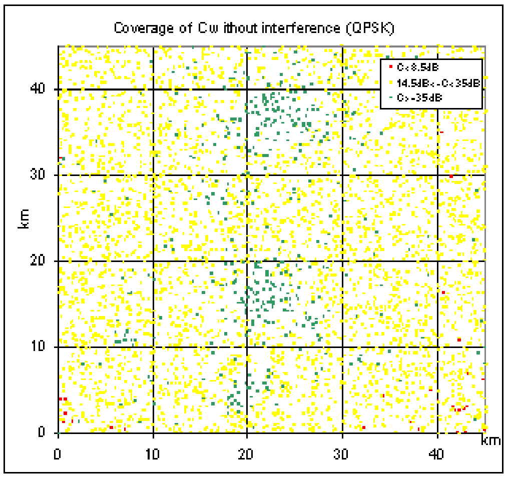


Figure 41. Example of the non-interfered network with the parameter setting of QPSK, GI=1/4 and FFT = 8k.

As expected, the parameter set of {QPSK, FFT 8k, GI 1/4} produces the largest coverage area practically without interferences (Figure 41). The results of this case can be considered thus as a reference for the interference point of view.

When the GI and FFT values are altered, the level of interference varies respectively. The results show that in addition to the parameter set of {FFT 8k, GI 1/4}, also {FFT 8k, GI 1/8}, {FFT 4k, GI 1/4} produces useful coverage areas, i.e. the balance of the SFN gain and SFN interferences seem to be in acceptable levels, whilst the other parameter settings produces highly interfered network.

The 16-QAM produces smaller coverage areas compared to the QPSK as the basic requirement for the $C / (N + I)$ of 16-QAM is 14.5 dB instead of the 8.5 of QPSK.

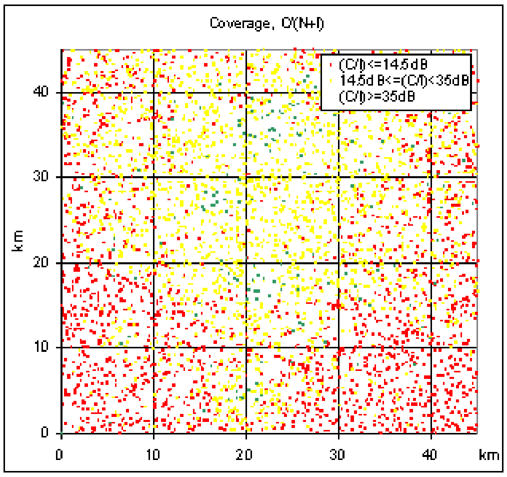


Figure 42. Comparative example of the simulation results for the parameter set of 16-QAM, FFT 8k and GI 1/16.

As can be observed from the previous analysis and Figure 42, the SFN interferences tend to cumulate to the outer boundaries of the planned coverage area.

For the QPSK with FFT 8k and GI ¼, the area is practically free of interferences, i.e. the $C / (N + I)$ is > 8.5 dB in every simulated location. The dark colour in the middle shows the site locations with the $C / (N + I)$ greater than 35 dB.

By observing the 90% probability in the cell edge (about 95% in the cell area), i.e., 5 % outage probability of Figures 43-44, the mode {FFT 8k, GI 1/4} provides a minimum of about 7 dB and the set of {FFT 8k, GI 1/8} and {FFT 4k, GI 1/4} provides the same performance in the whole investigated area. This similarity is due to the $D_{sfn}$ limit, which is complied totally in both of the cases as the sites are grouped inside about 30 km diameter. It is worth noting that the values are calculated over the whole map of 45km × 45km.

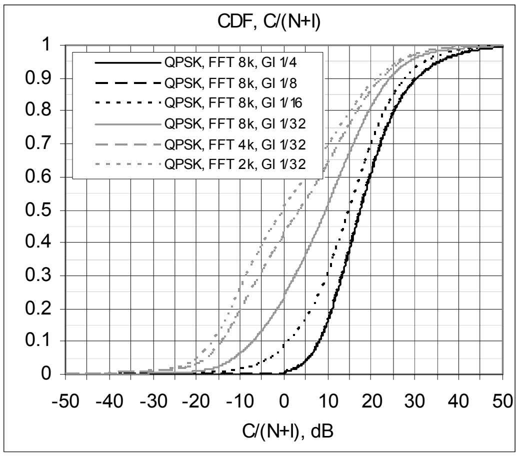


Figure 43. The cumulative $C/(N+I)$ distribution of different modes for QPSK.

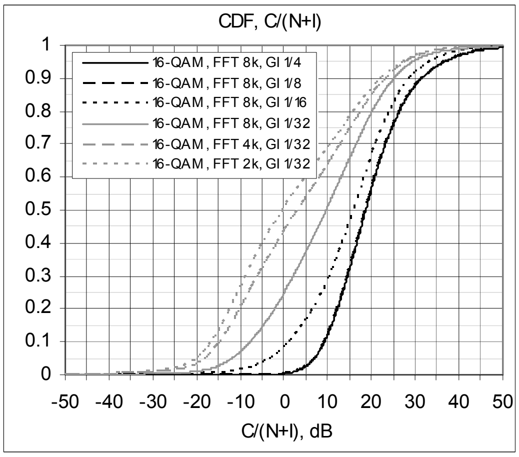


Figure 44. The cumulative $C/(N+I)$ distribution of different modes for 16-QAM.

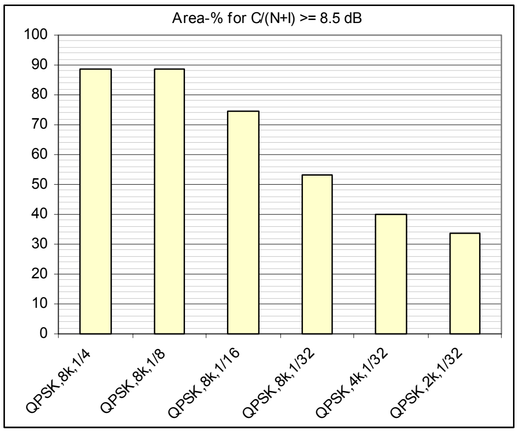


Figure 45. The functional area percentage for QPSK modes compared to the total area.

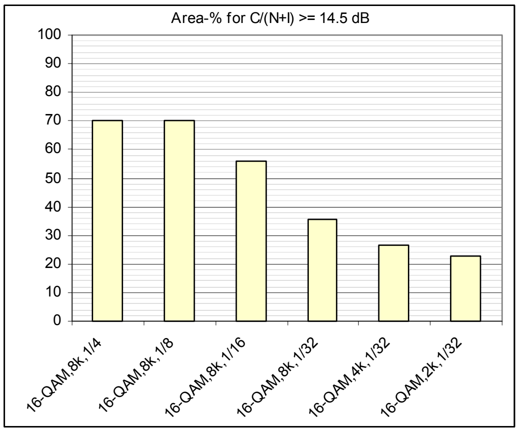


Figure 46. The functional area percentage for 16-QAM modes compared to the total simulated area.

It seems that the QPSK mode would provide a good performance in the investigated area when using the non-interfering {FFT 8k, GI 1/4} parameters, whilst 16-QAM gives smaller yet non-interfered coverage area. The advantage of the latter case is the double radio channel capacity compared to the QPSK with greater coverage area. The parameter set of {FFT 8k, GI 1/8} and {FFT 4k, GI 1/4} looks also useful, providing the possibility to either rise the maximum velocity of the terminal (FFT 4k), or give more capacity (GI 1/8). As for the rest of the parameter settings, the optimal balance can not be achieved due to the raised interference levels.

The SFN gain of the investigated network could be observed more specifically by switching on and off the individual sites, by carrying out the $C$ / ($N$ + $I$) simulations and by noting the differences in the cumulative density function. This is not, though, accurate method unless the simulations are limited inside the maximum calculated cell radius of each site. The network layout used in this case is highly irregular and does not contain too much overlapping areas compared to the total area of 45 km × 45 km, so the separate SFN gain investigation was not carried out. On the other hand, the presented results already include the total sum of the SFN gain and interference.

It can be estimated though that especially with the QPSK modes that provides with the largest coverage areas, the mountain site does have an effect within the overlapping areas of the nearest cells. According to the simulations presented in Figure 33, this case could provide an SFN gain of about 1 dB in such areas. Similarly, if there is spot with three overlapping cells in the middle of the area (i.e. in the area without interferences), the SFN gain could be around 2 dB according to the simulations presented in the chapter showing the balance of the SFN gain and SFN interferences. The results presented in [9] support this observation.

## 7. Conclusion

The presented simulation method provides both geographical and cumulative distribution of the SFN gain and interference levels. The method takes into account the balancing of the coverage and capacity as well as the optimal level of SFN gain and the interference level in case the over-sized SFN is used. It can be applied for the theoretical, e.g. hexagonal cell layouts, as well as for the practical environments, taking into account the radio propagation modelling for different sites.

The method can thus be used in the detailed optimization of the DVB-H networks. The principle of the simulator is relatively straightforward and the method can be applied by using various different programming languages. In these investigations, a standard Pascal was used for programming the core simulator.

The SFN gain results are in align with the practical results of e.g. [4] and [5] for the low number site. For the high number of the sites, no reference results were found due to the practical challenges in setting up the test cases. Nevertheless, estimating the theoretical limits by applying the formula 5, the results are in logical range. The SFN interference level results behave also logically and are in align with e.g. [10].

The results show that the radio parameter selection is essential in the detailed planning of the DVB-H network. As the graphical presentation of the results indicate, the effect of the parameter value selection on the interference level and thus on the quality of service can be drastic, which should be taken into account in the detailed planning of DVB-H SFN.

Especially the controlled extension of the SFN limit might be interesting option for the DVB-H operators. The simulation method and related results shows logical behaviour of the SFN error rate when varying the essential radio parameters. The results also show that the optimal setting can be obtained using the respective simulation method by balancing the SFN gain and SFN errors. As expected, the 8K mode is the most robust when extending the SFN whilst 4K limits the maximum site antenna height. 16-QAM provides suitable performance for the extension, but according to the results, even QPSK which provides larger coverage areas is not useless in SFN extension when selecting the parameters correctly.

## 8. References

[1] Jyrki T.J. Penttinen. The SFN gain in non-interfered and interfered DVB-H networks. The Fourth International Conference on Wireless and Mobile Communications 2008, IARIA, Published by the IEEE CS Press. 6p.

[2] DVB-H Implementation Guidelines. Draft TR 102 377 V1.2.2 (2006-03). European Broadcasting Union. 108 p.

[3] Masaharu Hata. Empirical Formula for Propagation Loss in Land Mobile Radio Services. IEEE Transactions on Vehicular Technology, Vol. VT-29, No. 3, August 1980. 9 p.

[4] Maite Aparicio (Editor). Wing TV. Services to Wireless, Integrated, Nomadic, GPRS-UMTS&TV handheld terminals. D8 – Wing TV Country field trial report. Project report, November 2006. 258 p.

[5] David Plets. New Method to Determine the SFN Gain of a DVB-H Network with Multiple Transmitters. 58th Annual IEEE Broadcast Symposium, 15-17 October 2008, Alexandria, VA, USA. 6 p.

[6] Jyrki T.J. Penttinen. The Simulation of the Interference Levels in Extended DVB-H SFN Areas. The Fourth International Conference on Wireless and Mobile Communications 2008, IARIA, Published by the IEEE CS Press. 6 p.

[7] Recommendation ITU-R P.1546-3. Method for point-to-area predictions for terrestrial services in the frequency range 30 MHz to 3000 MHz. 2007. 57 p.

[8] Gerard Faria, Jukka A. Henriksson, Erik Stare, Pekka Talmola. DVB-H: Digital Broadcast Services to Handheld Devices. IEEE 2006. 16 p.

[9] William C.Y. Lee. Elements of Cellular Mobile Radio System. IEEE Transactions on Vehicular Technology, Vol. VT-35, No. 2, May 1986. pp. 48-56.

[10] Airi Silvennoinen. DVB-H –lähetysverkon optimointi Suomen olosuhteissa (DVB-H Network Optimization under Finnish Conditions). Master's Thesis, Helsinki University of Technology, 15.5.2006. 111 p.

[11] Jyrki T.J. Penttinen. DVB-H Performance Simulations in Dense Urban Area. The Third International Conference on Digital Society, ICDS 2009, IARIA, Published by the IEEE CS Press. 6 p.

[12] Minseok Jeong. Comparison Between Path-Loss Prediction Models for Wireless Telecommunication System Design. IEEE, 2001. 4 p.

## Biography

**Mr. Jyrki T.J. Penttinen** has worked in telecommunications area since 1994, for Telecom Finland and it's successors until 2004, and after that, for Nokia and Nokia Siemens Networks. He has carried out various international tasks, e.g. as a System Expert and Senior Network Architect in Finland, R&D Manager in Spain and Technical Manager in Mexico and USA. He currently holds a Senior Solutions Architect position in Madrid, Spain. His main activities have been related to mobile and DVB-H network design and optimization.

Mr. Penttinen obtained M.Sc. (E.E.) and Licentiate of Technology (E.E.) degrees from Helsinki University of Technology (TKK) in 1994 and 1999, respectively. He has organized actively telecom courses and lectures. In addition, he has published various technical books and articles since 1996.

# Routing with Metric-based Topology Investigation

Frank Bohdanowicz, Harald Dickel, and Christoph Steigner
Institute for Computer Science
University of Koblenz-Landau
{bohdan,dickel,steigner}@uni-koblenz.de

## ABSTRACT

As routing takes place in an entirely distributed system where local routers have no direct access to globally consistent network state information, a routing algorithm has to make uncertain forwarding decisions. As the network state may change, due to failures or new adoptions of networks, routing algorithms have to adapt themselves to the new situation. This network convergence phase should be carried out as quickly and precisely as possible. Besides the problem of generating the proper updates for the locally distributed routers, the problem of forwarding the routing updates is also manifest: routing updates travelling along routing loops may become obsolete or outdated. We developed a new distance vector algorithm which solves the problem of routing loops. This provides distance vector routing with crucially improved convergence, stability, and scalability abilities, thus making distance vector routing once again an attractive revitalized alternative to link state routing.

*Keywords– routing; distance vector routing; metric-based topology investigation; routing loops; counting to infinity problem; routing convergence*

## 1. INTRODUCTION

The major goal of all routing algorithms is to achieve a fast and correct convergence after a change in the network topology. In this phase, the forwarding of the actual routing updates is crucial in contrast to the forwarding of existent but obsolete update information. In distance vector algorithms, routing updates which have made their way along network loops contain in most of the cases obsolete information which should not be considered anywhere further on. Whenever network topologies in the Internet contain loops, alternative routes are available in case of a link failure. Unfortunately, topology loops complicate the correct detection of routes.

The major approaches to cope with this problem are distance vector, link state, and vector path algorithms.

Distance vector routing algorithms like the Routing Information Protocol (RIP) [10] cannot cope with routing loops efficiently. A routing loop is the trace of a routing update which occurs at a router again reporting seemingly new reachability information which is based on already known information. This event results in a temporary inconsistency during the convergence process. The well-known split horizon approach fails if the network topology contains loops. In this case, the routing convergence is impeded because invalid old routing updates may find their way along topology loops and cause routing loops which appear as the well-known *counting to infinity* (CTI) problem [10]. RIP, as a classic representative of the distance vector protocol family, can only inadequately cope with CTIs by limiting the metric to a small maximum. This does not solve the CTI problem since misguided data traffic may congest the trace of the routing loop and the maximum metric cannot be reached in a short time. Up to now there has been no proper solution for the CTI problem.

We show in this paper that the distance vector approach can be improved by a mechanism that can recognize all those outdated updates which were propagated along loops. Our simulation results and analysis show that distance vector routing can be significanty improved. Our new Routing with Metric-based Topology Investigation (RMTI) protocol can alleviate the CTI problem found in distance vector routing protocols like RIP. Our RMTI protocol is entirely compatible to RIP since it uses the same routing update message format. The convergence time of our RMTI protocol does not depend on an upper metric limit, so it is applicable in large-scale network environments.

Path vector routing like the interdomain Border Gateway Protocol (BGP) [17] was designed to solve the routing loop problem by including the entire path (AS-path) from source to destination in its updates in order to detect the occurrence of routing loops. It has, however, been shown that BGP suffers from forwarding loops during routing convergence after topology changes [12, 15].

Due to the drawbacks of the classical distance vector routing, in recent years the focus in further development of interior routing protocols was on link state routing. But link state routing like Open Shortest Path First (OSPF) [11] do not solve the routing loop problem entirely due to the fact that these routing algorithms also suffer from forwarding loops [6, 8, 22]. The brute force effort of the link state algorithms, the overhead prone reliable flooding technique, limits its deployment [9]. Besides this, the link state approach has some disadvantages such as the absence of local routing policy facilities which provide network administrators with comprehensive capabilities to influence traffic density. In distance vector and path vector routing, the routing update flow is directly related to the actuated traffic flow. Thus distance vector and path vector algorithms can naturally cope with routing policies. By our deployment of RMTI, we show that distance vector routing algorithms can be an attractive alternative to the link state routing suite.

This paper gives a detailed description of our RMTI protocol, implementation, and evaluation results based on [1, 2]. The paper is organized as follows: In Section 2 we give a short overview of other approaches to distance vector routing which solve the CTI problem. In Section 3 we discuss the routing problem as a loop problem and state our vocabulary of loop concepts. In Section 4 we present the principles of our new RMTI approach. In Section 5 we present our protocol together with some characteristic examples of routing loop detection and update rejection. In Section 6 we describe our implementation. We close with our conclusion in Section 7.

## 2. RELATED WORK

To avoid routing loops and the CTI problem, several enhanced distance vector protocols which increase the amount of information exchanged among nodes and new routing architectures have been proposed.

The Ad hoc On-Demand Distance Vector (AODV) protocol [14] by Perkins expands the distance vector information originally based on subnet N, next hop NH and distance D, to a 4-tuple (N,NH,D,SEQ), where SEQ denotes the sequence number. The result is that although this approach is provably loop free [13], it is not compatible with RIP due to the required protocol changes. The Enhanced Interior Gateway Routing Protocol (EIGRP) used by Cisco is based on the DUAL algorithm [7] proposed by Garcia-Luna-Aceves. DUAL provides loop-free paths at every instance, which was proved in [7]. However, it is a Cisco proprietary routing protocol and not compatible with RIP due to a different protocol design. A solution called Source Tracing was proposed by Cheng et al [3] and Faimann [5]. In this approach, updates and routing tables provide additional information by adding a first-hop indication (the head of the path). Loops can be recognized recursively.

These protocols avoid the CTI problem because they provide loop freedom, but they are likewise not compatible with the RIPv2 standard or are proprietary approaches.

In contrast to these approaches, we aim to provide a solution that has a complete and solid backward compatibility with every existing implementation of RIPv2 [10]. The enhanced knowledge is based on the information already provided by the RIP protocol. So even deploying a new RMTI router only at selected nodes is possible.

## 3. THE ROUTING PROBLEM

In a computer network, a router's task is to connect several subnets to build an internetwork. Routers have to ensure that, within this internetwork, communication between arbitrary locations in different subnets becomes possible. In the following we use the term *network* as a synonym for the term *internetwork*. The Internet Protocol (IP) is the key communication protocol in such networks. It uses packet forwarding to deliver a data packet addressed to a destination in a certain subnet. A data packet is forwarded from router to router until it reaches a router which is directly connected to the data packet's destination subnet. Finally this router delivers the data packet to its destination. A router has to know which router is the *next hop router* in the forwarding process in order to reach a subnet. Therefore it maintains a forwarding table. Basically a forwarding table is a list of entries containing the next hop router for every subnet. The task of building up and maintaining a correct forwarding table is called *routing*. Usually, routing is achieved by a routing protocol which is running distributed among the routers. The fact that a routing protocol is a distributed system offers some favorable and welcome properties such as enhanced reliability and scalability, when compared to a centralized approach.

A real network is far from being a static entity. New subnets are connected, sometimes old ones are removed, new links between subnets are established via routers, or existing links and routers may fail. A routing protocol has to deal with all these events. Therefore routers have to communicate among each other all changes in the network relevant to the routing task. They are exchanging *update messages* to announce their actual view of the network state. Unfortunately it always needs some time to distribute the update messages all over the network. It is impossible to guarantee that all routers share a common and consistent view at any one time. Furthermore it is possible that *update messages* on their way through the network become outdated. The information enclosed may be no longer valid and may cause misleading assumptions by a router which receives such outdated information. It is still an ongoing challenge to design a routing protocol and make it work as a real distributed system that solves all routing problems.

All these problems get worse with the presence of loops in the network topology. Further on we will look at the problems caused by loops in detail.

Loops appear in different forms and on several occasions throughout this paper. We use five variations of the term *loop*: topology loop, forwarding loop, routing loop, Simple Loop, and Source Loop. To avoid confusion, we have to make a proper distinction between these concepts and the usage of the term loop in this paper.

A *topology loop* is a loop within the network based on the physical network topology. In Figure 1a we have the topology loop $(r_1, s_1, r_2, s_2, r_3, s_3, r_1)$ and in Figure 1c an additional topology loop $(r_1, s_4, r_4, s_5, r_5, s_6, r_1)$. Topology loops add redundancy by offering multiple routes to certain subnets and enhance the reliability of a network.

Data packet delivery in an IP network is done by hop by hop packet forwarding as stated above. A data packet addressed to a destination in a certain subnet $s$ is sent to the appropriate next hop router listed in the forwarding table. If a data packet forwarded by a router $r$ returns to router $r$ again after some intermediate hops, the entries in the forwarding tables of the routers have built a *forwarding loop*. Once a data packet gets into such a forwarding loop, it sticks in this loop and circles around, never reaching its destination.

The functionality of the data forwarding principle is based on the assumption that the next hop router has a shorter distance to the subnet $s$ than the actual router. Unfortunately, different routers may have a different and inconsistent view of the distances to a destination subnet caused by the distributed nature of the routing protocol. This inconsistent view may lead to the configuration of a forwarding loop. Forwarding loops can consume a large amount of network bandwidth and can impact the end-to-end performance of the network. Therefore it is necessary to recognize and to prevent forwarding loops. In practice all common routing protocols suffer from forwarding loops during the convergence process after a topology change in the network [8].

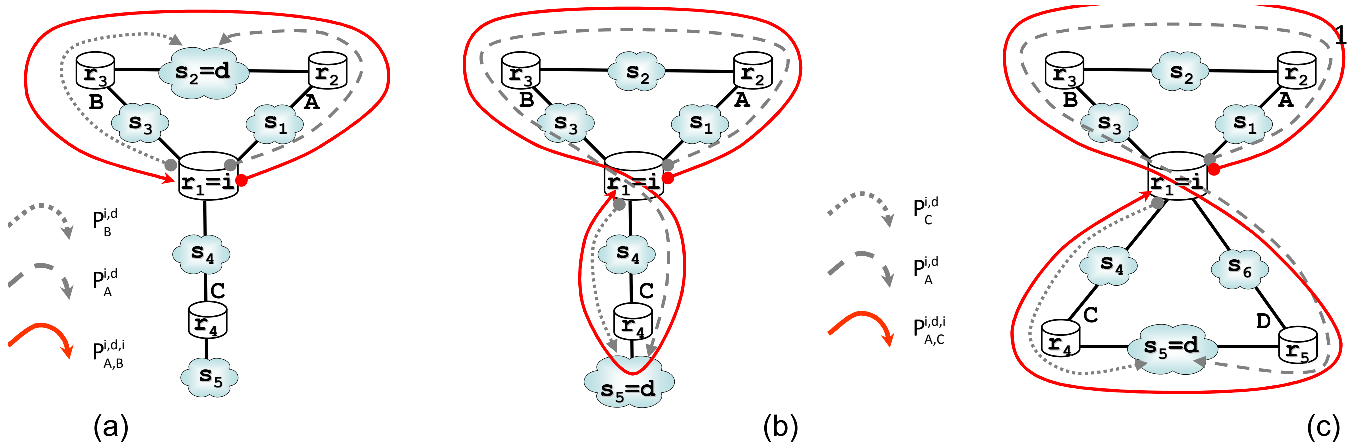


**Figure 1: Three examples of loops in a network. (a) shows a Simple Loop $P^{i,d,i}_{A,B}$ between the neighbor interfaces $A$ and $B$. (b) shows a Source Loop $P^{i,d,i}_{A,C}$ between the neighbor interfaces $A$ and $C$. The path $P^{i,d}_B$ is part of the path $P^{i,d}_A$. (c) shows a different topology with another Source Loop $P^{i,d,i}_{A,C}$ between the neighbor interfaces $A$ and $C$. Here the path $P^{i,d}_B$ is not enclosed in the path $P^{i,d}_A$.**

Forwarding loops arise in link state routing by computing inconsistent shortest path trees by distinct routers on the basis of different link state databases [22], or in distance vector routing due to the occurrence of a routing loop.

While the term *forwarding loop* denotes a loop in the forwarding process of a data packet, we use the term *routing loop* exclusively to denote a loop in the trace of a routing update. Like a forwarding loop reflects a loop in the forwarding process, a routing loop reflects a loop in the routing process. The routing process manages the forwarding table of a router and has direct influence on the forwarding process. Every routing loop is in close relation to a forwarding loop. If router $r_1$ in Figure 1a sends a routing update designating the reachability of subnet $s_4$ to router $r_2$, then $r_2$ may insert an updated entry to subnet $s_4$ in its forwarding table. Now $r_2$ forwards data packets to $s_4$ via $r_1$. The traffic flow takes the opposite direction of the update flow. So in fact every routing loop causes a corresponding forwarding loop in the opposite direction.

In distance vector routing, routing loops appear as the counting to infinity problem. Our RMTI approach avoids CTIs by evaluating the metrics of the routing updates more carefully than other distance vector algorithms. RMTI detects and distinguishes two different shapes of loops composed of traces of routing updates. We call these loops *Simple Loop* and *Source Loop*. A Simple Loop (Figure 1a) is a path which leaves a distinct router at one interface and comes back to the same router on another interface, *without having passed* through the same router in between. A Source Loop (Figure 1b+c) is a path which leaves a distinct router at one interface and comes back to the same router on another interface, *having passed* the same router in between.

In contrast to RIP, we can detect these loops by not deleting old routing information as soon as new and better information arrives at a router; rather, we maintain some information in order to detect Simple Loops and Source Loops.

We show that the detection of Source Loops enables us to avoid all kinds of CTI situations in all loop topologies by rejecting malicious routing updates. We implemented a fully functional version of RMTI and did comprehensive tests with different network topologies in order to analyze and evaluate the RMTI behavior.

## 4. DESIGN RATIONALE

Now we present a formal network model and a notation which is sufficiently comprehensive to sketch out and prove the concept of our approach.

### 4.1 The Network Model

A computer network is a collection of devices like hosts, routers, switches, or subnets which are connected via network links. In formal modeling, a computer network is typically represented by a graph. The devices are the nodes of the graph and the edges correspond to the network links. The routing problem is to find a path between any two nodes. For the purpose of discussing the routing problem in computer networks, it is sufficient to consider two types of nodes (subnets and routers) and one type of edges (interfaces). Figure 2 shows a network graph with 7 subnets $(s_1, \ldots, s_7)$, 5 routers $(r_1, \ldots, r_5)$ and 13 interfaces $(A, \ldots, M)$. The number of subnets $s_1, \ldots, s_n$ are subsumed to the set $\mathcal{S} = \{s_1, \ldots, s_n\}$. The subnets are connected via routers and the routers are attached to subnets via interfaces. The interfaces represent the network links and correspond to the edges of the graph.

In general we use upper-case characters to denote interfaces and $\mathcal{I}$ is the set of interfaces $\mathcal{I} = \{A, B, C, D, \ldots\}$. An interface is assigned to exactly one router and in our model a router is fully specified by its interfaces. So, we use the interfaces which are assigned to a router $r$ to define $r = \{U_1, U_2, \ldots, U_n\}, U_1 \ldots U_n \in \mathcal{I}$. In Figure 2 by example we have $r_1 = \{A, B, C\}, r_2 = \{D, E\}, \ldots,$ etc.

Let $\mathcal{R}$ denote the set of routers $\mathcal{R} = \{r_1, \ldots, r_k\}$. Since an interface is an element of one unique router only, we have $\forall U \in \mathcal{I}\ (U \in r_i \Rightarrow U \notin r_j), \quad r_i, r_j \in \mathcal{R}, i \neq j$.

Every interface is attached to exactly one subnet. An interface is the link of a router to one certain subnet.

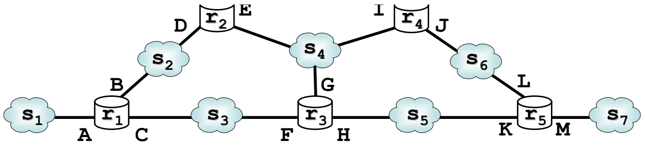


**Figure 2: Formal representation of a network.**

The topology of the network graph is given by the relation $CON \subseteq \mathcal{I} \times \mathcal{S}$ and defines the mapping between interfaces and subnets: $(U_i, s_j) \in CON, U_i \in \mathcal{I}, s_j \in \mathcal{S}$, if and only if interface $U_i$ is connected to subnet $s_j$.

Finally, a router $r \in \mathcal{R}$ is connected to subnet $s \in \mathcal{S}$, if and only if one of its interfaces $U \in r$ is connected to subnet $s$ that is $\exists U \in r$ with $(U, s) \in CON$.

Given the sets of subnets $\mathcal{S}$, interfaces $\mathcal{I}$, and routers $\mathcal{R}$ together with the connection relation $CON$, we have a complete formal specification of the topology of a network. It can be represented as a graph like the one in Figure 2. This graph is the representation of the following specification: $\mathcal{S} = \{s_1, \ldots, s_7\}, \mathcal{I} = \{A, \ldots, M\}, \mathcal{R} = \{r_1, \ldots, r_5\}$ with $r_1 = \{A, B, C\}, r_2 = \{D, E\}, r_3 = \{F, G, H\}, r_4 = \{I, J\}, r_5 = \{K, L, M\}$, and $CON = \{(A, s_1), (B, s_2), (C, s_3), (D, s_2), (E, s_4), (F, s_3), (G, s_4), (H, s_5), (I, s_4), (J, s_6), (K, s_5), (L, s_6), (M, s_7)\}$

Router nodes are drawn as cylinder, subnet nodes as clouds, and there is an edge between a router $r \in \mathcal{R}$ and a subnet $s \in \mathcal{S}$ if and only if $\exists U \in r$ with $(U, s) \in CON$.

To discuss the basic questions in the area of routing, namely "Is there a route from a certain node in the network to a certain destination?", we have to define some useful terms and properties about transitions from node to node in a network graph.

An elementary step from a router $i \in \mathcal{R}$ via outgoing interface $O \in i$ to an adjacent router $j \in \mathcal{R}$ via incoming interface $I \in j$ using subnet $s \in \mathcal{S}$ is called a *hop* (see Figure 3). It is defined by a 3-tuple $H^{i,j} = (O, s, I)$, whereas $(O, s), (I, s) \in CON$.

If the destination of a hop is simply given by a subnet and not by a designated interface of a router, we use the special symbol $*$ as the last element of the 3-tuple, i.e. the hop from router $i$ to subnet $d$ is notated as $H^{i,d} = (O, d, *)$.

Definition 1. *(Hop)*
*A* hop *$H$ from a router $i \in \mathcal{R}$ via outgoing interface $O \in i$ to an adjacent router $j \in \mathcal{R}$ via incoming interface $I \in j$ using subnet $s \in \mathcal{S}$ is the 3-tuple $H = (O, s, I)$ with $(O, s), (I, s) \in CON$ and $\mathcal{H}$ denotes the set of all hops.*

In abbreviated form we use the router identifiers as a superscript and the interface identifiers as a subscript to specify a hop. The notation $H^{i,j}_{O,I}$ indicates that this hop is leaving router $i$ via interface $O$ and leading to router $j$ via interface $I$ (Figure 3).

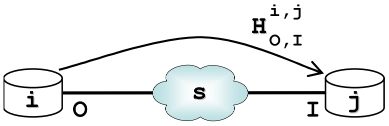


**Figure 3: The hop $H^{i,j}_{O,I} = (O, s, I)$ from router $i$ to router $j$ uses the outgoing interface $O$ and reaches the incoming interface $I$ of router $j$ via subnet $s$.**

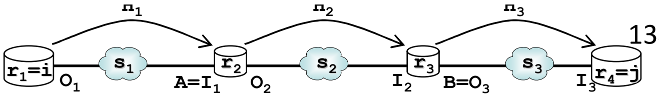


**Figure 4: A path $P^{i,j}_{A,B}$ from router $i$ to router $j$.**
$P^{i,j}_{A,B} = (H_1, H_2, H_3) = ((O_1, s_1, A), (O_2, s_2, I_2), (B, s_3, I_3))$

If two routers $i$ and $j$ are connected to the same subnet $s$, then there exists one hop $H^{i,j}$ from $i$ to $j$ as well as a hop $H^{j,i}$ from $j$ to $i$. In this case $i$ and $j$ are called *neighbors*.

Definition 2. *(Neighbor)*
*Let $i, j \in \mathcal{R}, i \neq j$ be two distinct routers. $j$ is a* neighbor *of $i$, iff $\exists O \in i, I \in j$, and $H^{i,j}_{O,I} \in \mathcal{H}$.*

In Figure 2 $r_1$ is a neighbor of $r_2$ and $r_2$ is a neighbor of $r_4$ but $r_1$ and $r_4$ are not neighbors.

If router $j$ is a neighbor of router $i$ there exists a hop $H^{i,j}_{O,I}$ from $i$ to $j$. According the definition of a hop, $O$ is an interface of router $i$ and $I$ is an interface of $i$'s neighbor $j$, where $O$ and $I$ are connected to the same subnet $s$. To point out this relationship between router $i$ and interface $I$ we call $I$ a *neighbor interface* of $i$. In Figure 2 $r_2$ has the three neighbor interfaces $B, G$, and $I$.

A path through the network is a sequence of hops (Figure 4).

Definition 3. *(Path)*
*A path $P^{i,j}_{A,B}$ beginning at router $i \in \mathcal{R}$ leading to router $j \in \mathcal{R}$ is a sequence of hops*

$$\begin{aligned} P^{i,j}_{A,B} &= (H_1, H_2, \ldots, H_l) \\ &= ((O_1, s_1, I_1), (O_2, s_2, I_2), \ldots, (O_l, s_l, I_l)) \\ &= ((O_1, s_1, A), (O_2, s_2, I_2), \ldots, (B, s_l, I_l)) \end{aligned}$$

*with $O_1 \in i, I_l \in j$, $I_1 = A, O_l = B$ and $\exists\ r \in R$ with $I_j, O_{j+1} \in r$ for $1 \leq j < l$. The metric of $P^{i,j}_{A,B}$ is $m^{i,j}_{A,B} = l$ which is simply the number of hops and $\mathcal{P}$ denotes the set of all paths.*

Again, we use an abbreviated form with superscripts to indicate the start and the destination router of a path and subscripts to specify the first and the last hop of a path more precisely. By using the notation $P^{i,j}_{A,B}$ of a path $P \in \mathcal{P}$, we specify that interface $A$ is a neighbor interface of router $i$, and interface $B$ is a neighbor interface of router $j$. With this notation we can distinguish in Figure 2 a path from $r_2$ to $r_5$ via $r_3$ and a path via $r_4$. The former is noted $P^{r2,r5}_{G,H}$ and the latter $P^{r2,r5}_{I,J}$.

If a path from router $i$ leads to a subnet $d$, then the last hop $H_l$ in the sequence of hops takes the form $H_l = (O_l, d, *)$ and we simply write $P^{i,d}$ using $d \in \mathcal{S}$ as a superscript instead of a destination router. And finally, we write $P^{i,d,j}$ to denote a path $P \in \mathcal{P}$ which traverses subnet $d$. In this case there exists a hop $H_k, 1 \leq k \leq l$ in $P$ with $H_k = (O_k, d, I_k)$.

## 4.2 Simple Loops and Source Loops

A closed path $P^{i,i}_{A,B}$ begins and ends at router $i$ through the network where the first hop $H_1 = (O_1, s_1, A)$ leaving router $i$ and the last hop $H_l = (B, s_l, I_l)$ returning to router $i$ use the different neighbor interfaces $A$ and $B$ (Figure 1).

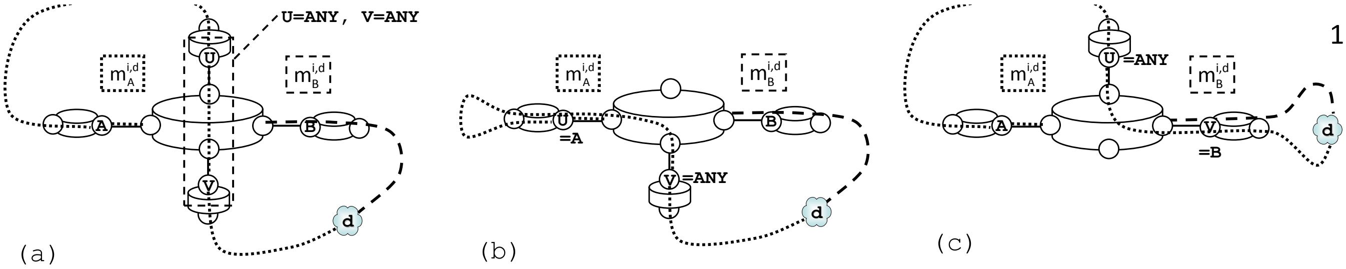


**Figure 5: Source Loop types. If a path $P^{i,d,i}_{A,B}$ is a Source Loop, the router $i$ is passed in between via two neighbor interfaces. These may be interface $A$, $B$, or $ANY$ (any other) neighbor interface of router $i$. Therefore we have $3^2 = 9$ possible Source Loop types (Table 1). Let us call these intermediate interfaces $U$ and $V$. (a) shows the Source Loop type with $U = ANY$ and $V = ANY$, (b) results with $U = A$ and $V = ANY$, and in (c) we have $U = ANY$ and $V = B$.**

It is tempting to assume that such a closed path offers two distinct useful paths to a subnet $d$ that is part of this path $P^{i,d,i}_{A,B}$ as shown in Figure 1a. However, this has to be considered very carefully. Figures 1b and 1c show examples that this assumption does not hold in general.

Figure 1a shows the network graph of an example network topology. If we examine paths from router $r_1 = i$ to a destination subnet $d = s_2$, there is a path $P^{i,d}_A$ via interface $A$ from neighbor router $r_2$ and a path $P^{i,d}_B$ via interface $B$ from neighbor router $r_3$ to subnet $d$. If we combine these paths, we obtain a closed path $P^{i,d,i}_{A,B}$, starting at router $i$ via neighbor $r_2$ and ending at router $i$ coming in from neighbor $r_3$.

The network topologies of Figures 1b and 1c show a different situation. Again there are two different paths from router $r_1 = i$ to a destination subnet $d$. There is a path $P^{i,d}_C$ from router $i$ via neighbor interface $C$ to subnet $d = s_5$ and a path $P^{i,d}_A$ from router $i$ via neighbor interface $A$ to subnet $d$. If we combine these paths, we obtain a closed path $P^{i,d,i}_{A,C}$ once again with both paths building a loop. But there is an important difference between the path $P^{i,d,i}_{A,B}$ in Figure 1a and the two paths $P^{i,d,i}_{A,C}$ in Figures 1b and 1c.

In Figure 1a there is a topology loop between the neighbor interfaces $A$ and $B$ of router $i$. If the path $P^{i,d}_B$ is no longer usable (due to a link or router failure on that path), there is an alternative path $P^{i,d}_A$ to subnet $d$ which router $i$ might use. But in the topologies of Figures 1b and 1c the paths $P^{i,d,i}_{A,C}$ traverse router $i$ in between. There is no topology loop via the neighbor interfaces $A$ and $C$ at router $i$. If the path $P^{i,d}_C$ is no longer usable, there is no alternative path $P^{i,d}_A$ to subnet $d$ which router $i$ might use since $P^{i,d}_C$ is part of the path $P^{i,d}_A$. Therefore we have to distinguish between these two different types of closed paths. A path from router $i$, traversing a subnet $d$, ending at router $i$ but never passing router $i$ in between is called a Simple Loop.

Definition 4. *(Simple Loop)*
*A Simple Loop is a path $P^{i,d,i}_{A,B}$ where $O_1, I_l \in i$, $I_1 = A$, $O_l = B$, $\exists n\ 1 \le n \le l\ s_n = d$, and $\forall I_j\ 1 \le j < l\ I_j \notin i$.*

We denote the set of all Simple Loops within a network with $SIL$ and the metric of a Simple Loop $P^{i,d,i}_{A,B} \in SIL$ is $\text{silm}^{i,d,i}_{A,B} = m^{i,d,i}_{A,B} = l$.

A path from router $i$, traversing a subnet $d$ and ending at router $i$ but this time passing router $i$ in between is called a Source Loop.

Definition 5. *(Source Loop)*
*A Source Loop is a path $P^{i,d,i}_{A,B}$ where $O_1, I_l \in i$, $I_1 = A$, $O_l = B$, $\exists n\ 1 \le n \le l\ s_n = d$, and $\exists I_j\ 1 \le j < l\ I_j \in i$*

Figure 5 explains the possible Source Loop types depending on what interfaces are used by the intermediate passing of router $i$. Only a Simple Loop provides an alternative path to a destination subnet (Figure 1a). On the other hand, the attempt to use the first hop of a Source Loop $P^{i,d,i}_{A,B}$ in order to reach subnet $d$ results in the configuration of a routing loop (Figure 1b and 1c). Discovering a Simple Loop at a router requires the exclusion of the possibility that a closed path $P^{i,d,i}_{A,B}$ is a Source Loop. In order to detect Source Loops, we need to identify the Simple Loop with the lowest metric out of a set of recognized Simple Loops of different metric sizes. This can be done by simply inspecting all Simple Loop metrics.

The minimal Simple Loop metric (msilm) between two neighbor interfaces $A$ and $B$ on router $i$ is:

$$\text{msilm}^{i}_{A,B} = \min\{\text{silm}^{i,d,i}_{A,B} \text{ for all subnets } d\}$$

Furthermore, we define the minimal return path metric (mrpm) which can be derived from the minimal Simple Loop metric (msilm). The minimal return path metric (mrpm) via an interface A on router $i$ is:

$$\text{mrpm}^{i}_{A} = \min\{\text{msilm}^{i}_{A,B} \text{ for all neighbor interfaces } B \neq A \text{ of router } i\}$$

Based on the minimal return path metrics, we are able to give a criterion which is sufficient to rule out that a path is a Source Loop.

Theorem 1. *(Simple Loop Test)*
*Let $P^{i,d}_B$ be the path to subnet d with the lowest metric $m^{i,d}_B$ and $P^{i,d}_A$ another path to subnet d with metric $m^{i,d}_A$, and $B \neq A$. Then the path $P^{i,d,i}_{A,B}$ is a Simple Loop if the following inequality holds:*

$$m^{i,d}_A < mrpm^{i}_A + m^{i,d}_B \qquad (1)$$

*Proof.* The path $P^{i,d,i}_{A,B}$ consists of the constituents $P^{i,d}_A$ and $P^{i,d}_B$ with $m^{i,d,i}_{A,B} = m^{i,d}_A + m^{i,d}_B - 1$. If $P^{i,d,i}_{A,B}$ is a Source Loop, the constituent $P^{i,d}_A$ passes router $i$ and consists of $P^{i,i}_{A,C}$ and $P^{i,d}_D$ for some interfaces $A, C, D \in IF$ with $m^{i,d}_A = m^{i,i}_{A,C} + m^{i,d}_D$. $\text{mrpm}^i_A$ is the minimal return path metric via $A$, so that $m^{i,i}_{A,C} \geq \text{mrpm}^i_A$. $m^{i,d}_D \geq m^{i,d}_B$, because $m^{i,d}_B$ is the lowest metric to subnet $d$ by precondition. This results in $m_A^{i,d} = m_{A,C}^{i,i} + m_D^{i,d} \geq \mathrm{mrpm}_A^i + m_B^{i,d}$. So, if $m_A^{i,d} < \mathrm{mrpm}_A^i + m_B^{i,d}$, the path $P_{A,B}^{i,d,i}$ is not a Source Loop and therefore must be a Simple Loop. □

In order to form a Source Loop, the lower limit of $m_A^{i,d}$ is the sum of the minimal return path metric $\mathrm{mrpm}_A^i$ and the metric $m_B^{i,d}$ in order to reach destination subnet $d$ via neighbor interface $B$. If the inequality holds, the path $P_{A,B}^{i,d,i}$ is simply *too short* to be a Source Loop and therefore must be a Simple Loop. We call the verification that this inequality holds the *Simple Loop Test*. The metric of this path $P_{A,B}^{i,d,i}$ is:

$$m_{A,B}^{i,d,i} = m_A^{i,d} + m_B^{i,d} - 1 \tag{2}$$

If the Simple Loop Test holds for a path $P_{A,B}^{i,d,i}$, we have detected a Simple Loop of metric $m_{A,B}^{i,d,i} = m_A^{i,d} + m_B^{i,d} - 1$.

Figure 1a shows a Simple Loop. There is a path $P_B^{i,d}$ with metric $m_B^{i,d} = 2$ and a path $P_A^{i,d}$ with metric $m_A^{i,d} = 2$, too. The minimal return path metric $\mathrm{mrpm}_A^i$ is 3 which yields:

$$m_A^{i,d} = 2 < 3 + 2 = \mathrm{mrpm}_A^i + m_B^{i,d}$$

The inequality is satisfied, the Simple Loop Test is successful and therefore it is proved that $P_{A,B}^{i,d,i}$ is a Simple Loop.

Figure 1b shows a concrete Source Loop of Figure 5c's type. The topology of the network is the same as in Figure 1a, so again $\mathrm{mrpm}_A^i$ is 3, but we choose a different subnet $d = s_5$. The path $P_C^{i,d}$ has the metric $m_C^{i,d} = 2$ and the metric of path $P_A^{i,d}$ is 5. In this case the inequality is not fulfilled and the Simple Loop Test fails:

$$m_A^{i,d} = 5 \not< 3 + 2 = \mathrm{mrpm}_A^i + m_B^{i,d}$$

Figure 1c shows a Source Loop in a different network topology. It is of Figure 5a's type. The path $P_A^{i,d}$ consists of the Simple Loop $P_{A,B}^{i,i}$ and the path $P_D^{i,d}$. The Simple Loop Test is always performed using the path with the lowest metric to the destination subnet, which is in this case $P_B^{i,d}$. Given that the metric $m_B^{i,d} \leq m_D^{i,d}$ it is evident that no Source Loop can pass the Simple Loop Test:

$$m_A^{i,d} = 5 \not< 3 + 2 = \mathrm{mrpm}_A^i + m_B^{i,d} \quad (\leq \mathrm{mrpm}_A^i + m_D^{i,d})$$

Again the Simple Loop Test fails.

So far we have analyzed the properties of paths in a static network. We developed the Simple Loop Test in order to identify Simple Loops out of the metric of a closed path in a network. To apply these considerations to distance vector routing, we have to take the distance vector routing process into account. The metrics to destination subnets are sent via update messages between adjacent routers throughout the network. So we have to look at the potential succession patterns of update messages and the chronological sequence in which these messages arrive at a router. However, the Simple Loop Test together with the split horizon rule is sufficient to prevent the acceptance of any update message that produces a Source Loop. To prove this, we examined all possible Source Loop types produced by a simple combinatorial scheme.

| | $V = B$ | $V = A$ | $V = ANY$ |
|---|---|---|---|
| $U = A$ | ESH | ISH+ESH | ESH (fig.3b) |
| $U = B$ | ISH | SLT | SLT |
| $U = ANY$ | SLT (fig.3c) | SLT | SLT (fig.3a) |

**Table 1: Four of the nine Source Loop types (see Figure 5) are prohibited by the application of the split horizon rule at the local router (internal split horizon, ISH) or the split horizon rule at the neighbor router (external split horizon, ESH). The remaining five Source Loop types are detected by the application of the Simple Loop Test (SLT).**

Assume an update message arrives at router $i$ via neighbor interface $A$ containing the metric $m_A^{i,d}$ to subnet $d$. If router $i$ has already an entry in its routing table to subnet $d$ via a neighbor interface $B \neq A$ and metric $m_B^{i,d}$, we can build a closed path $P_{A,B}^{i,d,i}$ with metric $m_{A,B}^{i,d,i} = m_A^{i,d} + m_B^{i,d} - 1$. If this path $P_{A,B}^{i,d,i}$ is a Simple Loop, there is a loop in the network between the neighbor interfaces $A$ and $B$ of router $i$. However, it is also possible that $P_{A,B}^{i,d,i}$ is a Source Loop.

If the path $P_{A,B}^{i,d,i} = (H_1, H_2, \ldots, H_l)$ is a Source Loop, the router $i$ is passed in between via two neighbor interfaces $U$ and $V$ of router $i$. Therefore, $H_n$ and $H_{n+1}, (1 < n < l,\ I_n, O_{n+1} \in i)$ exists in the sequence of hops with $H_n = (\mathbf{U}, s_n, I_n)$ and $H_{n+1} = (O_{n+1}, s_{n+1}, \mathbf{V})$ (Figure 5).

In this case router $i$ must have received an update message from some neighbor interface $V$ sometime in the past and sent out an update message to some neighbor interface $U$ containing metric $m_V^{i,d}$. This information must have traveled through the network in a loop, finally returning to router $i$ from neighbor interface $A$. Since $U$ or $V$ may be neighbor interfaces $A$, $B$, or any other neighbor interface of router $i$, there are $3^2 = 9$ possible Source Loop types. Thus, the occurrence of a Source Loop is either prohibited by the split horizon rule or can be detected by the Simple Loop Test (Table 1).

If $U = V$, the split horizon rule of the local router $i$ (internal split horizon) eliminates the possibility that such a Source Loop occurs. If $U = A$, the split horizon rule of the neighbor router with interface $A$ (external split horizon) eliminates the possibility that such a Source Loop occurs (Figure 5b). If $U = ANY$ and $V = B$ (Figure 5c), the Simple Loop Test detects a Source Loop like in Figure 1b. And finally, if $U \neq A$ and $V \neq B$ (Figure 5a), the Simple Loop Test detects a Source Loop like in Figure 1c.

## 4.3 RMTI

The basic functionality of RMTI is the evaluation of redundant routing information which would usually be rejected immediately by the router. The RMTI processes are independent of the underlaying distance vector protocol. There is no need for an additional or altered communication between routers enhanced with RMTI. RMTI improves the underlaying routing protocol and allows it to maintain or enlarge the existing network infrastructure.

Up to now, we have implemented RMTI on the basis of the Routing Information Protocol (RIP). Thus, our RMTI-protocol is compatible to RIP. RIP works as follows: Assume that a RIP router i receives a routing update from an adjacent RIP router j to subnet $d$ with metric $m^{j,d}$ via interface A of router j. This indicates the existence of a corresponding path $P^{j,d}$ from RIP router j to subnet $d$. Router i does not know the complete path $P^{j,d}$, but knows the number of hops this path consists of. Then, from the view of router i, there will be a path $P_A^{i,d}$ with metric $m_A^{i,d} = m^{j,d} + 1$ by prepending a hop from i to j to the path $P^{j,d}$. Therefore,

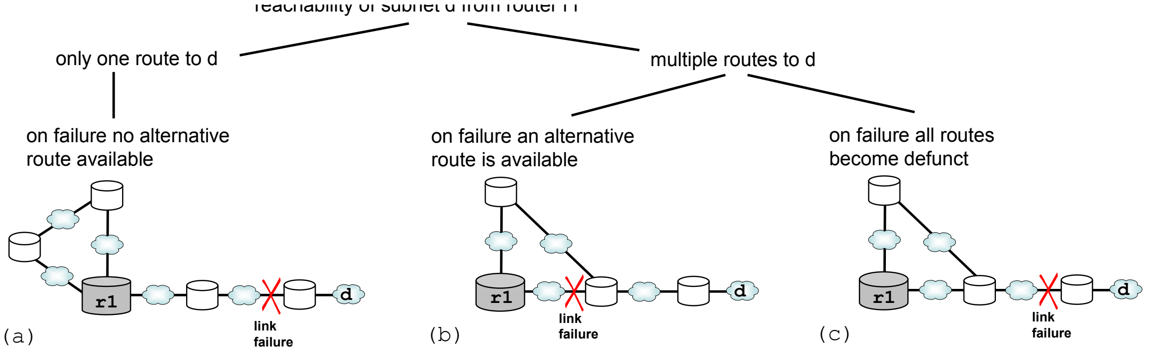

reachability of subnet d from router r1
only one route to d
multiple routes to d
on failure no alternative route available
on failure an alternative route is available
on failure all routes become defunct
r1
d
link failure
(a)
r1
link failure
d
(b)
r1
d
link failure
(c)

**Figure 6: The basic topologies containing a network loop. The position of the loop and the location of a link failure determines the availability of alternative routes**

router i knows the metric $m_A^{i,d}$ and the first hop toward d. If a RIP router has a valid path to subnet $d$, it will reject all equivalent or inferior paths to the same subnet.

RIP has three timers: an update timer (default 30 sec) for sending out routing updates periodically, a timeout timer (default 180 sec) to recognize invalid entries in the routing table and to mark them as unreachable when the timer expires, and a garbage collection timer (default 120 sec) to completely delete an entry from the routing table.

The drawback of distance vector routing in general, and RIP in particular, is the vulnerability to routing loops and, in correlation to that, the CTI problem. Due to this CTI problem, the RIP specification limits the distance of a route to a subnet to the maximum metric of 15 hops. Metric 16 marks the subnet as unreachable and is defined by the RIP term *infinity*. This restriction does not avoid the CTI problem but reduces its duration and, therefore, its impact on the network. Besides that, it also limits the maximum size and complexity of RIP networks. The split horizon technique only avoids routing loops between two directly connected routers (two-hop loops), but fails within topology loops.

In addition to the usual operation of distance vector routing, RMTI builds up two tables containing information about topology loops in the local network environment. The msilm-table contains upper bounds for the minimal Simple Loop metrics between every two neighbor interfaces and the mrpm-table contains the minimal return path metrics $\text{mrpm}_A^i$ for all neighbor interfaces of the local router $i$.

As shown in Figure 1a, assume that router $i$ has an entry to subnet $d$ via neighbor interface $B$ with metric $m_B^{i,d}$. This indicates a path $P_B^{i,d}$. Now $i$ receives an update to subnet $d$ via neighbor interface $A \neq B$. This indicates an alternative path $P_A^{i,d}$ to subnet $d$ with metric $m_A^{i,d}$. A combination of these paths results in a closed path $P_{A,B}^{i,d,i}$ with metric $m_{A,B}^{i,d,i} = m_A^{i,d} + m_B^{i,d} - 1$. We perform the Simple Loop Test to verify that $P_{A,B}^{i,d,i}$ is a Simple Loop. If this test is passed, we have detected a Simple Loop with metric $m_{A,B}^{i,d,i}$. If this Simple Loop is the first one between the interfaces $A$ and $B$ that we have detected so far, or it has a lower metric than all other detected Simple Loops between $A$ and $B$, we update the entry in the msilm table and calculate the new minimal return path metrics $\text{mrpm}_A^i$ and $\text{mrpm}_B^i$.

Due to the fact that the mrpm entry corresponds to the metric of a minimal Simple Loop and the mrpm entries are needed in the Simple Loop Test, the question of initialization has to be considered. At the beginning, the initial upper bound values for the minimal Simple Loop metrics (msilm) and the minimal return path metrics (mrpm) are set to $2 * \text{infinity} - 1$ (with the usual value infinity$= 16$ this is 31), indicating that no actual upper bounds have been calculated out of updates so far. In addition, we have to perform the Simple Loop Test by replacing the initial value for $\text{mrpm}_A^i$ with 2 in order to perform the strictest possible application of this test (2 is the metric of the smallest possible Simple Loop generally). Thus the Simple Loop Test is reduced to test $2 > |m_A^{i,d} - m_B^{i,d}|$ in this case. Every Simple Loop contains at least one subnet which is located opposite to the considered router in the topology loop. Both neighbor routers, which span the Simple Loop over the underlaying topology loop, advertise a route to this subnet and the difference between the metrics of the two routes will be smaller than 2. So there is always a subnet $d$, that allows the detection of a Simple Loop in the initial case. Once the initial Simple Loop Test is passed successfully, the metric of the first Simple Loop can be calculated on the metrics of the two corresponding routes. By the time the routing process is converged, every router has detected all local Simple Loops. This loop knowledge now enables RMTI to effectively meet all challenges during the convergence phase of a network after link failures and topology changes, as the next section will show.

## 5. PERFORMANCE AND PARADIGMS

Topology loops provide connection redundancy in order to ensure reliability and stability of the network. Figure 6 shows basic topologies containing a topology loop. Three basic cases are illustrated there. If a network topology contains loops, certain subnets are reachable from a router by more than one link. In case of a link failure on the preferred route, there might be an alternative route to the subnet.

The answer to the question, whether or not there is an alternative route available from router $r_1$ to subnet $d$ after a link failure, depends on the position of the loop and the location of the link failure in the topology.

In Figure 6a, there exists only one route to subnet $d$ from router $r_1$. In case of a link failure to that route, there is no alternative route available to subnet $d$ despite the existence of the loop. In the topology of Figure 6b, the loop provides two possible routes from router $r_1$ to subnet $d$. There is an alternative route available if a link failure occurs within this loop. However, if the link failure occurs beyond this loop, all routes to subnet $d$ from router $r_1$ become defunct and no alternative route exists (Figure 6c).

If router $r_1$ is a conventional RIP router, the topology of Figure 6a is prone to routing loops and the CTI problem. Router $r_1$ accepts malicious routing updates to subnet $d$ from its neighbors. Section 5.3 presents the test results with RMTI instead of RIP in operation and demonstrates the main advantage of RMTI over RIP. Complex network topologies with multiple loops are composed of the basic topologies in Figure 6. Section 5.4 demonstrates the ability of RMTI to handle these complex network topologies. The test results show that RMTI is far superior in solving routing decisions as compared to standard RIP.

We implemented RMTI on top of an existing RIP router in order to perform the following performance tests. Thus, we are able to demonstrate the applicability of the RMTI approach in real deployment environments and substantiate its benefits, especially compared to standard RIP.

We use the well-known and widely used Quagga routing software suite to enable and explore the routing process. The Quagga routing suite consists of several advanced routing software daemons for Linux and Unix-like systems, where the *ripd* daemon implements the RIPv2 protocol used as the underlaying routing protocol. We additionally implemented a communication link in a network test environment which allow us to influence the succession of routing updates and evaluate network situations frequently.

## 5.1 The Test Environment

Figure 7 shows the operation schema of our test environment we developed in order to evaluate the performance of our RMTI daemon. We extended our routing daemon by a client-server communication to a supervisor which allows us to influence the succession of routing updates and the logging of all local information which appear on the routing daemon. Thus, we get a comprehensive representation of the available network information. The supervisor allows us to manipulate the routing behavior of every routing daemon by triggering updates in a predefined sequence in order to test critical network situations like the CTI problem. Our evaluation focused on topologies containing a loop. In topologies without a loop RMTI works just as well as standard RIP. We examined both routing daemons on a huge number of various network scenarios to verify the benefit of RMTI when it was exposed to the CTI problem and to verify the equality in the absence of the CTI problem. We performed a frequent modification of the network topology by using the open source virtualization tool *Virtual Network User Mode Linux* (VNUML) [19] which is based on *User Mode Linux* (UML) [4]. Using User Mode Linux is an efficient way to run various Linux systems inside one physical machine and connect them via virtual interfaces to build up arbitrary network topologies. VNUML is a comfortable tool to quickly design and boot up virtual network scenarios. A VNUML network can be administered and used just like a network with real Linux machines. Virtualization techniques are especially good for setting up arbitrary network topologies. They provide an opportunity to utilize real operating systems and routing software without the need for all the hardware like routers, switches and the cabling. Apart from this, a modification of the virtual network topology is much easier and quicker to perform than with real networks. Virtualization techniques are particularly suited to support the study of routing performance in complex network topologies. As we didn't want our RMTI daemon to become heavily optimized for virtual networks, we additionally connected real router hardware to our VNUML virtual networks. We used Linksys WRT54G routers equipped with the open source router operating system *openwrt* [20], which is a Linux-based routing system. Thus, we also tested the functionality of our RMTI daemon on real router hardware. VNUML additionally offers a complex but flexible network topology.

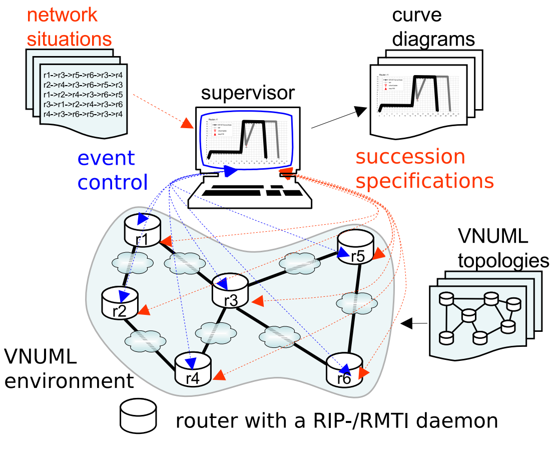


**Figure 7: The operation scheme of our test environment**

Our supervisor consists of a routing update generator which can coordinate all updates, so that all crucial succession patterns (latency and timing issues) in a specific topology will be exhausted. First, a topology of routing daemons has to be chosen and launched by VNUML. Second, all paths which contain a loop have then to be identified and represented. Third, this knowledge is used by the generator to control the updates of every routing daemon, to provoke a situation where the CTI problem will arise. Additionally all events are logged. After these initial conditions have been established, all routers are switched back to automatic mode, where they are no longer controlled and act autonomously. The provocation of a CTI causes the one update with old information to be injected into the topology loop, and the routing protocol has to react to this event. As shown in this section, RIP propagates faulty information and the course of CTI events begins within the topology loop. Metric changes are displayed directly in the test environment so CTIs can be observed or detected automatically. The curve diagrams depicted in this section describe the metric progression of a considered route on a distinct router in different network scenarios. All network scenarios described in this section were performed in our test environment and analyzed with our supervisor.

## 5.2 The Counting to Infinity Problem

In the following we demonstrate the occurrence of the CTI problem within the network scenario exemplified by Figure 8, which is the basic topology of Figure 6a. Starting from router $r_1$, which holds a route to subnet d via neighbor interface B with metric $m_B = 3$, we assume that the link to

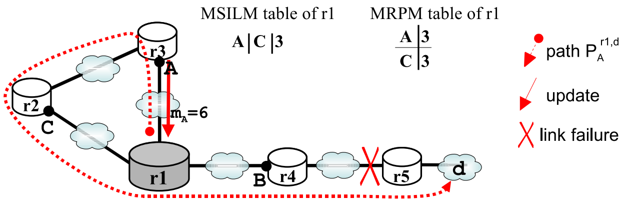


**Figure 8: The CTI problem arises.**

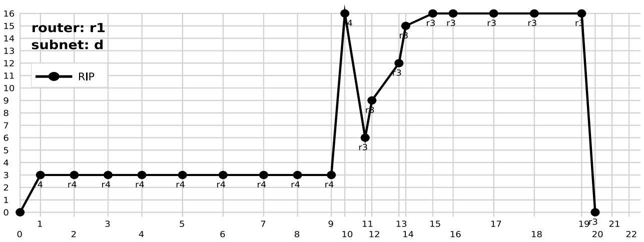


**Figure 9: The curve follows the metric of the route to subnet $d$ on router $r_1$. It depicts the course of CTI events.**

subnet $d$ via router $r_4$ becomes corrupted. This is announced by an update which router $r_1$ receives from $r_4$ and, therefore, the corresponding route in $r_1$'s routing table is marked as unreachable. Next $r_1$ receives a routing update via neighbor interface A from router $r_3$ advertising an alternative route to subnet $d$ (via $r_2$) with metric $\mathrm{m}_A = 6$. However, the routing information of this update was propagated along a Source Loop.

If router $r_1$ is a conventional RIP router, it will accept this faulty routing update from $r_3$, adapt its routing table, and propagate the new routing information. Then a routing loop is established and the course of CTI events will begin. Moreover, as long as the CTI events have not come to an end, a forwarding loop exists and a large amount of network bandwidth is consumed, due to data packets having been sent to subnet $d$ and circulating in the forwarding loop. The CTI events also suppress the propagation of the correct routing update announcing the subnet as unreachable. Existing RIP extensions like triggered updates would speed up the CTI problem but in the worst case routing updates would get lost and the elimination of the routing loop would require more time. The RIP extension Split Horizon also does not avoid the CTI problem in this situation.

The curve chart in Figure 9 refers to the network situation described in Figure 8. It shows the metric values of the route to subnet $d$, captured on the router $r_1$ while the events of the CTI sequence are progressing. The y-axis represents the metric of the route to subnet $d$ in the range from 0, as the destination subnet is unknown to the router $r_1$, to 16, as the destination subnet is known with metric 16 (infinity). The x-axis represents the number of check and update events in relation to the corresponding route entry in the routing table of the regarded router, e.g., incoming updates. Metric changes are directly displayed in a time-synchronous manner. The curve chart is generated automatically by our supervisor.

The curve chart in figure 9 shows the metric of the route to subnet $d$ on router $r_1$ and describes the CTI progression. As depicted in the chart, router $r_1$ holds a route to subnet $d$ via router $r_4$ with metric 3 (x-axis 1-9). Just before the CTI arises, the route to subnet d becomes corrupted (x-axis 10) and router $r_1$ marks it with infinity (metric 16). Next, router $r_1$ gets a supposed alternative route to subnet $d$ from router $r_3$ with metric 6 (x-axis 11). However, this route is outdated and invalid and belongs to a path which already includes router $r_1$. If router $r_1$ is equipped with standard RIP, it accepts the route to subnet $d$ via router $r_3$ and a routing loop occurs. The sequential incrementation of the metric up to infinity (x-axis 12-15) is the characteristic of the CTI problem.

## 5.3 Basic Topologies

The CTI problem in Figure 8 is prevented by router $r_1$, if it is enhanced with our RMTI approach. Regarding the situation in Figure 8, the routing update from $r_3$ will be rejected by $r_1$ and subnet $d$ is kept unreachable in $r_1$'s routing table because there is no Simple Loop spanned over router $r_3$ and $r_4$.

The test results captured on router $r_1$ are shown in Figure 10. The separate, unfilled, downward pointing triangle indicates that RMTI on $r_1$ receives and rejects the *critical update* that would have induced the CTI problem (gray curve in the background of Figure 10). The symbols and curve lines used are explained in the corresponding legend of each curve chart.

In the network situation of Figure 8, the RMTI router $r_1$ decides whether the update from router $r_3$ should be accepted or not after a link failure. As there is no Simple Loop between $r_3$ and $r_4$, the same information cannot be advertised from both routers.

The following case study (see Figure 11) refers to a pitfall of Simple Loop detection. It is not sufficient to perform the Simple Loop Test (Equation 1, Section 4.2) just with the last valid entry for a subnet $d$ in the forwarding table of a router $r_i$. This may lead to detecting a Source Loop as a Simple Loop by mistake. Instead, we always have to use the smallest metric for subnet $d$ recently known by $r_i$.

Starting from router $r_1$, which holds a route to subnet $d$ via $r_2$, we assume that subnet $d$ becomes unreachable due to a link failure behind $r_2$. Router $r_2$ sends a routing update with metric 16 to router $r_1$ and $r_4$. Router $r_1$ adapts its routing table and advertises the new information about subnet $d$ to its neighbors. Before router $r_4$ receives this routing update from $r_2$, it sends a routing update with old invalid reachability information about subnet $d$ to $r_1$. Router $r_1$ accepts the routing update from $r_4$ as seemingly new information about subnet $d$.

Additionally, we assume the situation is similar to the CTI

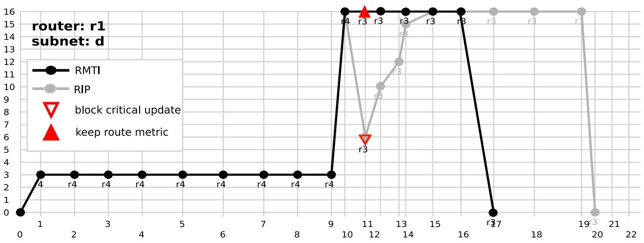


**Figure 10: The CTI is prevented by RMTI router $r_1$ discarding the faulty route information from $r_3$.**

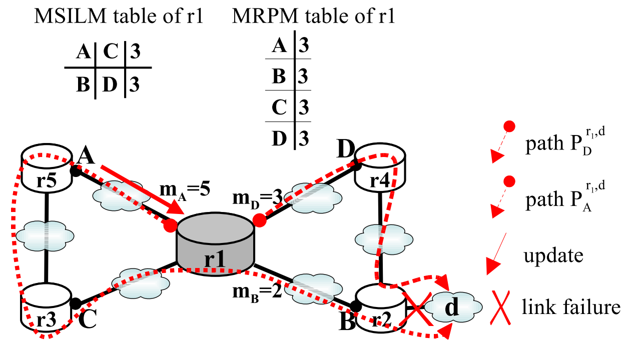


**Figure 11: The CTI problem arises. In this situation using the right values to perform the Simple Loop Test is essential to detect all Simple Loops correctly.**

problem in Figure 8. Router $r_5$ receives the routing update with metric 16 which was sent from $r_1$ before. Router $r_5$ adapts its routing table and marks the route to subnet $d$ as unreachable. However, router $r_3$ does not receive the routing update from $r_1$ and keeps its route to subnet $d$ as reachable. Next, router $r_5$ receives a routing update from $r_3$ with seemingly alternative reachability information about subnet $d$. Router $r_5$ accepts this route information and advertises it to $r_1$. Although router $r_1$ does not accept the routing update, it detects a Simple Loop between its neighbor interfaces $A$ and $D$. This is because $r_1$ receives information about subnet $d$ from $r_4$ with metric $\mathrm{m}_D = 3$ and from $r_5$ with metric $\mathrm{m}_A = 5$ and the Simple Loop Test of $r_1$ is passed with $m_A < mrpm_A + m_D \Rightarrow 5 < 3 + 3$. Hence, a CTI could occur, due to a falsely detected and stored Simple Loop.

We handle such situations by not immediately adjusting the internal data of RMTI with every accepted routing update that changes the routing table. The RMTI has to be performed with the smallest metric of the route to subnet $d$ seen in a short time interval before the metric is changed to a higher value or is set to infinity. Therefore, we have to store the smallest metric of the recent past. The time interval should depend on the RIP update timer interval. In this case $r_1$ receives redundant information about subnet $d$ from $r_4$ and $r_5$ but RMTI still uses the information about subnet $d$ from $r_2$ with metric $\mathrm{m}_B = 2$. Thus, the Simple Loop Test fails with $m_A < mrpm_A + m_B \Rightarrow 5 < 3 + 2$ and the Source Loop is detected.

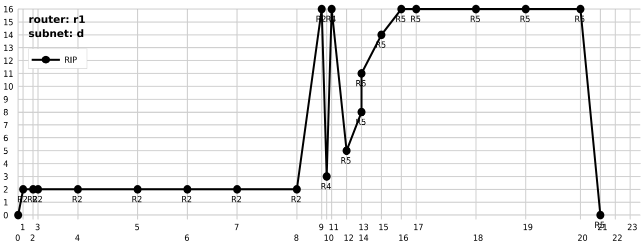


**Figure 12: The curve diagram of the CTI problem in relation to Figure 11.**

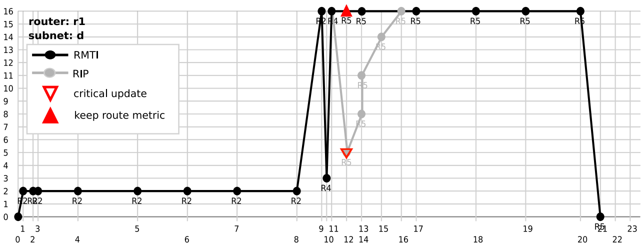


**Figure 13: The CTI problem is prevented by RMTI using always the shortest route to the destination subnet of the recent past.**

If router $r_1$ marks its route to subnet $d$ as unreachable and gets a routing update from $r_5$ with seemingly new information about subnet $d$, $r_1$ recognizes the malicious update by the missing Simple Loop between its neighbor interfaces $A$ and $B$. The curve chart in Figure 12 shows the succession of metric changes while the CTI problem occurs. However, as shown in Figure 13, although the routing information which returns to the router is no longer available in the routing table, RMTI can prevent the CTI problem.

## 5.4 Complex Topologies

In complex topologies the Simple Loop Test is needed (see Equation 1, Section 4.2) to prevent the CTI problem. In Figure 14, the network topology that was shown in Figure 8 is extended by a link between router $r_3$ and $r_4$. In this topology, a routing update sent from router $r_3$ to $r_1$ might contain valid reachability information about subnet $d$ because of the existing Simple Loop. If $r_1$ loses the route to subnet $d$ via $r_4$, subnet $d$ could be either still reachable or completely unreachable depending on the unknown location of the failed link. If the link between $r_1$ and $r_4$ fails, subnet $d$ will still be reachable for $r_1$ via $r_3$. Whereas if the link between $r_4$ and $r_5$ fails, subnet $d$ will become unreachable for $r_1$. However, if the CTI problem occurs, RMTI can still cope with these situations.

If the link between router $r_4$ and $r_5$ is corrupted and subnet $d$ is not reachable anymore, then router $r_4$ will propagate a routing update to its neighbors $r_1$ and $r_3$. Both accept the new route information from $r_4$ and mark their route to subnet $d$ as unreachable. But router $r_2$, having learned the route from $r_1$, sends a routing update with old route information to $r_3$. Router $r_3$ accepts this update from $r_2$ as a new valid route and replaces the old route entry in its routing table. If the routing update is then announced back to $r_1$, a Source Loop will appear. The RMTI router $r_1$ detects this Source Loop and prevents the CTI problem by rejecting the routing update from router $r_3$. The Simple Loop Test of $r_1$ fails with $m_A < mrpm_A + m_B \Rightarrow 6 < 3 + 3$. Therefore, the Source Loop is detected due to the inappropriate metrics.

However, due to an invalid old routing update from $r_3$ very shortly after the link failure was perceived by $r_1$ but not yet by $r_3$, router $r_1$ accepts the old invalid information about subnet $d$ from $r_3$ as a seemingly valid alternative route. This cannot be prevented because the routing update from $r_3$ could be invalid old route information as well as a valid alternative route information, e.g., if either the link between $r_4$ and $r_5$ or the link between $r_1$ and $r_4$ became corrupted.

Again, the major question in this situation concerns the data used in the Simple Loop Test. Assuming router $r_1$ and $r_3$ lost their route to subnet $d$. Then shortly after $r_1$ marks the route as unreachable, it accepts a faulty old routing update from $r_3$ with invalid route information. Router

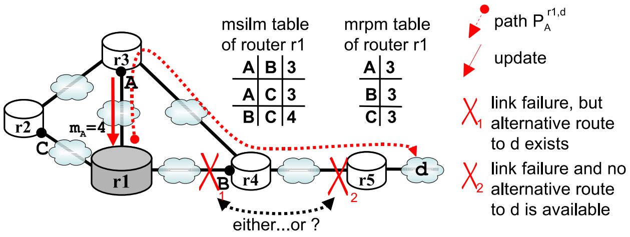


**Figure 14: Such a faulty routing update is generally difficult to detect for routers because the location of the link failure creates an ambiguous situation. However, the acceptance of the update does not induce a CTI in this case and does not confuse RMTI router $r_1$ which can still prevent the occurrence of the CTI problem.**

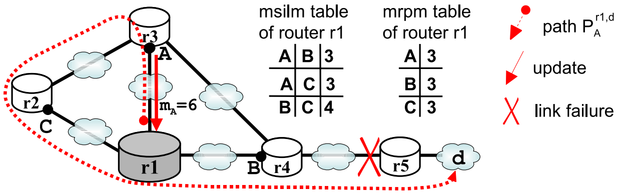


**Figure 15: RMTI router $r_1$ is able to detect this routing update via the Simple Loop Test.**

$r_3$ immediately corrects this by sending another routing update announcing subnet $d$ as unreachable. Router $r_1$'s last valid route to subnet $d$ describes a path via $r_3$ whereas $r_2$ still has the old route information from $r_1$. If this route information is announced back to $r_1$ in a Source Loop via $r_3$, as described in Figure 15, the Simple Loop Test in $r_1$ performs $m_A < mrpm_A + m_B \Rightarrow 6 < 4 + 3$ and is passed with $6 < 7$. Hence, if RMTI router $r_1$ would perform the Simple Loop Test in this way, simply with the latest valid metric of the route from the routing table, it would accept the routing update and the CTI problem would occur in this situation.

With the topology explained in Figure 11, we have shown how to handle such situations by not immediately adapting new route information from the routing table into RMTI. The behavior of the Simple Loop Test is stricter with lower metrics than with higher metrics. In this situation RMTI ignores the routing update from $r_3$ with metric 4 long enough and prevents the CTI problem. The metric progression of this network situation is shown in Figure 16.

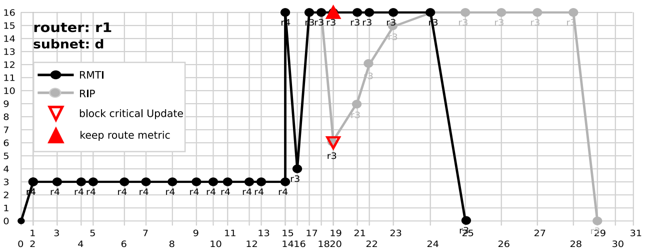


**Figure 16: The RMTI ignores routing updates which are advertised directly after a route failure because they could cause confusion. The routing update from $r_3$ (x-axis 16) is ignored by RMTI, otherwise the CTI problem could not be prevented.**

## 5.5 Indistinguishable Routes

A special case is shown in Figure 17 where the decision to accept or to reject a route not only depends on the length of the routes' metrics. It may be possible that the metric of an alternative route is equal to or even larger than the limit given by the Simple Loop Test. As shown in Figure 17, the routing updates may contain a valid alternative route to subnet $d$ (dashed line) or an invalid Source Loop (dotted line). Both variants of routing updates advertise the same metric 6 to $r_1$. The Simple Loop Test of $r_1$ performs $m_A < mrpm_A + m_B \Rightarrow 6 < 3 + 3$ and fails with $6 < 6$. Therefore, the routing update is rejected by the RMTI router $r_1$.

Figure 17 shows a Source Loop (dotted line) and an alternative path (dashed line). Router $r_1$ cannot distinguish the difference between these paths due to the same metric. In order to solve this decision problem, RMTI uses a timer to limit the execution time of the Simple Loop Test. RMTI supposes that Source Loops can be deleted by sending a routing update with metric infinity. Then a timer is started for a given time interval and further incoming routing updates with a valid metric will be blocked. If within this time interval a routing update with metric 16 is transmitted by $r_3$, a Source Loop has just been deleted and a CTI is prevented. If the timer expires and the route is still offered with a valid metric, it has to be a valid route. As the critical period for CTI situations is rather short, the timeout interval

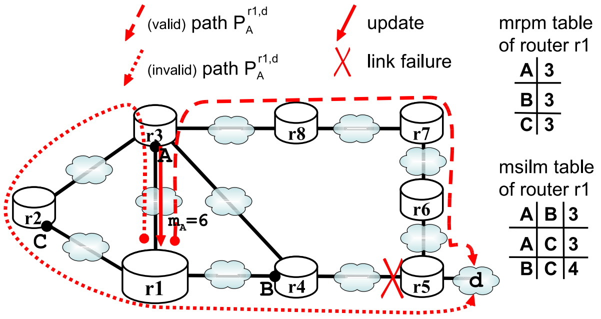


**Figure 17: RMTI has to distinguish between a valid and an invalid routing update with the same metric announced via the same neighbor interface.**

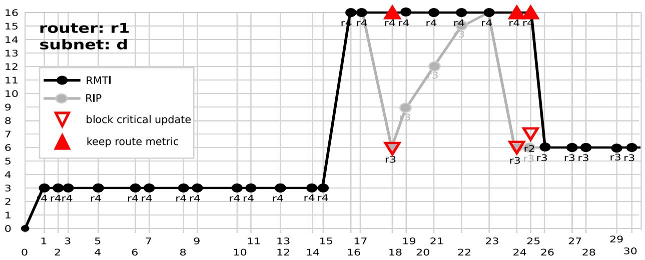


**Figure 18: RMTI mainly decides on metrics. However, if a route is refused by RMTI, it will be blocked just for a short time interval. During this time interval, an existing Source Loop should be deleted by a transmitted routing update.**

| Time | Route Status | RIP | RMTI |
|---|---|---|---|
| Start | Insert the route into the routing table and use it for forwarding. | | |
| Phase 1 | Default timeout timer is reinitialized with every incoming update. By default the timeout timer expires after 6 missing updates. (route metric < infinity) | Shorter routes overwrite longer routes. | Shorter routes overwrite longer routes. The minimal Simple Loops are identified and stored |
| The route becomes invalid (e.g., timeout timer expires) | Set the metric to infinity and do not use it for forwarding. Stop the timeout timer and start the garbage collection timer. | | |
| Phase 2 | Default garbage collection timer will be stopped by an incoming update with a metric smaller than infinity. (route metric = infinity) | The first routing update advertised with a metric smaller than infinity is accepted as alternative route. | Only routes in a Simple Loop with the route's adjacent subnet are accepted as alternative routes. |
| Garbage Collection Timer expired. | Delete the route from the routing table. | | |

**Table 2: Comparison between RIP and RMTI operation phases**

of the timer can be short, too. The timer could depend on the routing update interval or on the metric of the Simple Loops involved.

The curve diagram in Figure 18 shows the test results of the RMTI router within the network situation described in Figure 17. First, the CTI problem is avoided and no alternative route to the destination subnet (x-axis 18-23) exists. Second, a valid alternative route is advertised by router $r_3$; again $r_1$ rejects the update for the first time (also an incoming update from router $r_2$) but then the timer expires, a request is sent, and the alternative route is accepted next (x-axis 24-26).

## 6. IMPLEMENTATION OF RMTI

Up to now we have implemented RMTI as a slight extension to standard RIP. RMTI simply enhances RIP on detection of topology loops and prevention of routing loops. There is no need to change the RIP interaction behavior or any of the RIP message formats. Therefore, RMTI is compatible with RIP and both can be deployed on routers in the same network. In this section we introduce the implementation of RMTI in more detail and describe the changes which must be made in an existing RIP implementation. The template of a RMTI implementation which is described here is based on our existing RMTI daemon implementation. RMTI is implemented as an extension to the RIP update procedure of the routing table in which the present route information is compared with incoming new route information about the same destination subnet.

The control flow of the entire RMTI process is described in a flow chart in Figure 19. As shown in Figure 19a, the RIP implementation has to be extended at a particular point where the RMTI enhancements are then implemented. The control flow diagram is divided into a data collection phase (Figure 19b) and a decision phase (Figure 19c) in relation to the two distinct phases RMTI mainly operates in. Simple Loops will be detected in the data collection phase whereas Source Loops will be prevented in the decision phase. If the present route in the routing table is valid and an alternative route to the same subnet is advertised to the RMTI router, the metric of the corresponding path composed of the two routes will be calculated by equation 2 (see Section 4.2). The obtained path is a Simple Loop if equation 1 holds. Only the Simple Loop with the smallest metric is stored as the minimal Simple Loop of the corresponding two neighbor interfaces for further use. This happens as long as the route to the destination subnet is still available with a valid metric. When the route becomes invalid and is marked as unreachable in the routing table, the information about the existing Simple Loops is used to decide, whether or not to accept an advertised alternative route to the subnet via another neighbor router. Table 2 summarizes all operation phases and compares the behavior of RMTI with standard RIP.

| constant name | meaning |
|---|---|
| RMTIinfinity | RIP infinity *2 |
| RMTIalive | RIP timeout timer +<br>RIP garbage collection timer |

**Table 3: RMTI constants**

| variable name | meaning |
|---|---|
| destination | the IP-address of the destination subnet of the route. |
| metric ($m_B$) | the distance to the destination subnet. |
| nexthop | the next hop router (or gateway) along the route to the destination, this is the IP-address of the neighbor interface. |
| timer | the amount of time since the entry was last updated. |
| rmti_oldmetric | the last valid metric, used to calculate the metric of the path combination when the route is timed out and the metric is set to infinity. |
| rmti_oldnexthop | the corresponding next hop to the rmti_oldmetric variable. |
| rmti_validity | a boolean variable which indicates the phases the RMTI protocol operates in (RMTI data collection phase or RMTI decision phase). |
| rmti_synctime | a time stamp variable which specifies a time interval to control the point when the old metric and the old next hop should be adjusted with the information in the routing table. |
| rmti_checktime | a time stamp variable which marks the route in the routing table to recognize route advertisements which have already been rejected before. |

**Table 4: The variables in the route structure**

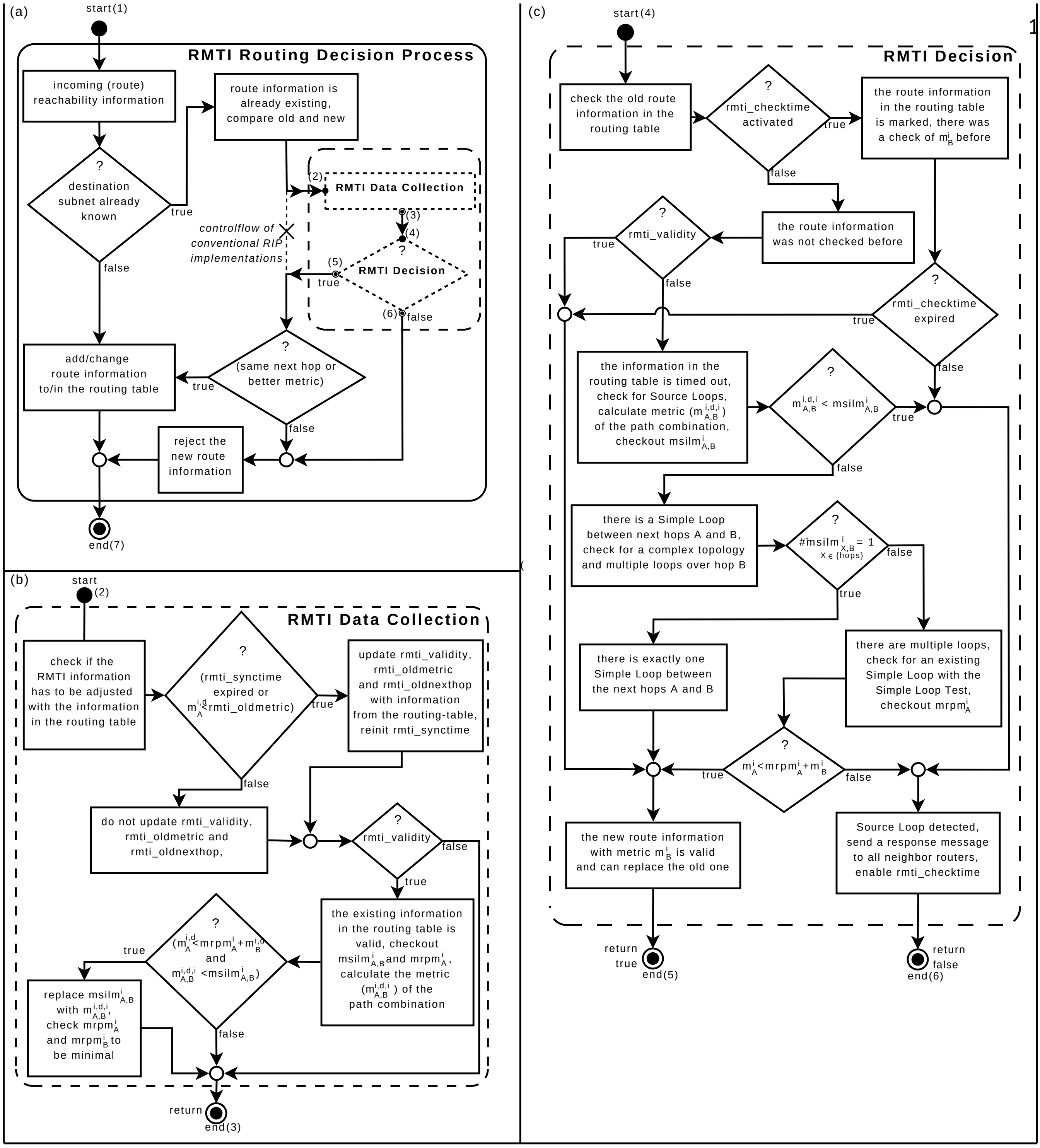

(a)
start(1)
RMTI Routing Decision Process
incoming (route) reachability information
? destination subnet already known
true
false
route information is already existing, compare old and new
(2)
RMTI Data Collection
(3)
(4)
? RMTI Decision
(5)
true
(6)
false
controlflow of conventional RIP implementations
? (same next hop or better metric)
true
false
add/change route information to/in the routing table
reject the new route information
end(7)
(b)
start (2)
RMTI Data Collection
check if the RMTI information has to be adjusted with the information in the routing table
? (rmti_synctime expired or $m^{i,d}_{A}$≠rmti_oldmetric)
true
false
update rmti_validity, rmti_oldmetric and rmti_oldnexthop with information from the routing-table, reinit rmti_synctime
do not update rmti_validity, rmti_oldmetric and rmti_oldnexthop,
? rmti_validity
false
true
the existing information in the routing table is valid, checkout $msilm^{i}_{A,B}$ and $mrpm^{i}_{A}$, calculate the metric ($m^{i,d,i}_{A,B}$) of the path combination
? ($m^{i,d}_{A}$<$mrpm^{i}_{A}$+$m^{i,d}_{B}$ and $m^{i,d,i}_{A,B}$ <$msilm^{i}_{A,B}$)
true
false
replace $msilm^{i}_{A,B}$ with $m^{i,d,i}_{A,B}$, check $mrpm^{i}_{A}$ and $mrpm^{i}_{B}$ to be minimal
return
end(3)
(c)
start(4)
RMTI Decision
check the old route information in the routing table
? rmti_checktime activated
true
false
the route information in the routing table is marked, there was a check of $m^{i}_{B}$ before
the route information was not checked before
? rmti_validity
true
false
? rmti_checktime expired
true
false
the information in the routing table is timed out, check for Source Loops, calculate metric ($m^{i,d,i}_{A,B}$) of the path combination, checkout $msilm^{i}_{A,B}$
? $m^{i,d,i}_{A,B}$ < $msilm^{i}_{A,B}$
true
false
there is a Simple Loop between next hops A and B, check for a complex topology and multiple loops over hop B
? #$msilm^{i}_{X,B}$ = 1 $X \in \{hops\}$
false
true
there is exactly one Simple Loop between the next hops A and B
there are multiple loops, check for an existing Simple Loop with the Simple Loop Test, checkout $mrpm^{i}_{A}$
? $m^{i}_{A}$<$mrpm^{i}_{A}$+$m^{i}_{B}$
true
false
the new route information with metric $m^{i}_{B}$ is valid and can replace the old one
Source Loop detected, send a response message to all neighbor routers, enable rmti_checktime
return true
end(5)
return false
end(6)

**Figure 19: Flow charts of the RMTI implementation**
(a) The flow chart of the entire RMTI process embedded in standard RIP. The RMTI algorithm can be implemented as a separate procedure independent of the existing implementation.
(b) The procedure of collecting information data by RMTI. If the route to the destination subnet in the routing table is available, the RMTI detects Simple Loops and stores the minimal one.
(c) This is the core procedure of RMTI which is entered only if the route in the routing table is marked with metric 16 (infinity) and might be replaced by the new alternative route information. Depending on the relation between the metric of the combined path of the two routes and the corresponding *msilm* value the return value of RMTI is true if the new route information meets all requirements of RMTI or false if not (see Equation 1, Section 4.2). Nevertheless, if the new alternative route information is accepted by RMTI, it also has to pass the remaining RIP decision procedures to be eventually inserted in the routing table. However, it will be completely discarded if the new alternative route is refused by the RMTI.

## 6.1 Constants and Data Structures

Based on a RIP implementation as underlaying router platform, there are two new constants necessary for an implementation of RMTI. They are constructed on well-known RIP constants. Table 3 describes the RMTI constants and their relation to the existing RIP constants. RMTIinfinity correlates to the infinity metric of RIP and marks a detected Simple Loop as no longer valid. RMTIinfinity corresponds to the longest possible Simple Loop metric calculated by equation 2 (see Section 4.2), when both routes to the destination subnet are marked as unreachable with metric infinity. As the metric of infinity is 16 in standard RIP, the metric of RMTIinfinity is 31.

The timeout constant RMTIalive corresponds to the number of seconds before a detected Simple Loop is marked as invalid with metric RMTIinfinity. RMTIalive is defined by the sum of the timeout timer and the garbage collection timer of the underlaying standard RIP implementation.

In comparison with RIP, RMTI needs a slightly modified routing table with a few additional constants. The routing table of standard RIP contains routes consisting at least of the destination subnet, the next hop router, and the metric of the route. Furthermore, a timer variable is needed to handle invalid routes. Table 4 describes all variables of a route entry in the routing table, which have to be implemented within RMTI based on RIP. Furthermore, RMTI uses two global tables, called *msilm* and *mrpm* table. Both tables store the metrics of the detected minimal Simple Loops to be used for the decision whether to accept or reject an alternative route advertisement.

The msilm table (see Table 5) stores the *minimal simple loop metric*. An entry in the msilm table consists of the two related neighbor interfaces which are identified by the source IP addresses of the incoming routing updates and the calculated metric of the corresponding Simple Loop. Additionally, a timer variable is used to detect if a Simple Loop is corrupted and the corresponding msilm entry is not valid anymore. Although the complexity of the msilm table increases quadratically with each stored metric, the msilm table remains rather small because the maximum number of Simple Loops depends on the number of neighbors. The msilm table is symmetric, i.e., the table entry msilm$_{A,B}$ is equal to entry msilm$_{B,A}$. There is no need to store both values, so the table can be linearized to save storage space and speed up access rates.

The mrpm table (see Table 6) contains the *minimal return path metric* addressed with the one corresponding neighbor interface. The mrpm entry corresponds to the smallest metric of any Simple Loop which starts at the specific neighbor interface and returns at the router via another arbitrary neighbor interface. Thus the mrpm entry is the smallest msilm value of a distinct neighbor interface and could be implemented as a reference to the corresponding msilm entry.

If the topology changes or a neighbor router becomes corrupted, the corresponding mrpm entry must be deleted. Therefore, the validity of the mrpm entry is proved with a timer variable and marked with RMTIinfinity when the timer expires.

## 6.2 The RMTI Data Collection Phase

As long as the route in the routing table is valid with a metric lower than infinity, RMTI gathers information about Simple Loops in relation to the subnet under consideration and updates the msilm and mrpm tables (Figure 19b). When the RMTI data collection phase is provided with further routing updates, the *rmti_synctime* variable is being evaluated first. The rmti_synctime variable is needed to cope with the problem of malicious updates (see Section 5.3). The information in the msilm and mrpm tables are not adjusted as soon as new routing information is entered in the routing table, but also the rmti_synctime variable has to be expired. If the given time interval is exceeded, it will be reset and the present information in RMTI will be adjusted with the information in the routing table.

Due to malicious routing updates which would affect RMTI negatively, we do not use the very last route information from the routing table but instead the last route information with the smallest metric in the given time interval. If the rmti_validity variable is false (resp. $metric = infinity$), the process will be aborted, because the considered route in the routing table is marked as unreachable and could not be part of a Simple Loop. If the rmti_validity variable is true ($metric < infinity$), RMTI will detect and update Simple Loops.

The Simple Loop Test is used to prove if the new incoming alternative route and the present route in the routing table can be combined to a valid Simple Loop. If the Simple Loop Test passes successfully, the metric of the detected Simple Loop will be calculated.

The Simple Loop with the smallest metric is stored as the minimal Simple Loop metric according to the two corresponding neighbor interfaces in the msilm table. If a msilm entry is changed, the mrpm entries of each neighbor interface must be recalculated. Then the RMTI data collection phase has come to an end and the decision phase has started.

## 6.3 The RMTI Decision Phase

Figure 19c describes the control flow of the RMTI decision phase. In this phase an alternative route would be completely rejected or passed along to the standard RIP decision process. At the beginning of the RMTI decision phase the rmti_checktime variable is checked in order to handle indistinguishable routes we have explained in Section 5.5. When an identical routing update was rejected by RMTI in the recent past, the rmti_checktime variable contains the time since the first rejection.

| variable name | meaning |
|---|---|
| interfaceA | the IP address of the neighbor interface A |
| interfaceB | the IP address of the neighbor interface B |
| msilm$_{A,B}$ | the minimal Simple Loop metric between the neighbor interfaces A and B |
| timer | the amount of time since the msilm entry was last acknowledged by routing updates |

**Table 5: The structure of the msilm entry**

| variable name | meaning |
|---|---|
| interfaceA | the IP address of the neighbor interface A |
| mrpm$_A$ | the minimal return path metric of neighbor interface A |
| timer | the amount of time since the mrpm entry was last acknowledged |

**Table 6: The structure of the mrpm entry**

If the value of the rmti_checktime variable exceeds a predefined time span an incoming alternative route will be passed along to the conventional RIP decision process without any further checks by RMTI and the rmti_checktime variable will be set to zero. If the rmti_checktime variable is zero an incoming alternative route will always be checked by RMTI.

If rmti_validity is true (resp. $metric < infinity$), a valid route exists in the routing table and the RMTI decision phase can be aborted without further consideration. If it is not true ($metric = infinity$), the new route information has to be processed to decide if it can replace the existing route information in the routing table. RMTI calculates the metric of the path combination and compares it to the msilm entry $\text{msilm}^i_{A,B}$. If $\text{msilm}^i_{A,B}$ is higher, then no Simple Loop has been detected between the two neighbor interfaces involved and the new route information can be completely rejected. If any Simple Loop exists between two neighbor interfaces, the metric of the smallest Simple Loop is stored in the msilm table. Otherwise, if no Simple Loop exists, the msilm table contains the initial value RMTIinfinity for this pair of neighbor interfaces. If a Simple Loop is stored in the msilm table, the presence of multiple loops must be excluded before the alternative route information can be approved. Dealing with multiple loops is more complex. We have explained the characteristics of complex topologies in Section 5.4.

Complex topologies can be discovered by counting the entries of a neighbor interface in the msilm table. If the neighbor interface of the old existing information has exact one entry listed in the msilm table, it has only one Simple Loop with another neighbor interface. Then the new alternative route information is passed along. If a complex topology is discovered, the Simple Loop Test (Equation 1, Section 4.2) has to be performed. If the Simple Loop Test fails, the new route information must be rejected and the present route information with metric infinity will be kept. If the Simple Loop Test passes, the new alternative route information will be accepted and passed along to the routing decision process of RIP.

As described in Section 5.5, due to indistinguishable routes with equal metrics, a new alternative route information is blocked by RMTI within a short period of time. When the first rejection of a route occurs, the rmti_checktime variable for the existing route in the routing table is activated and a corresponding routing update with metric infinity is sent out to all neighbors. This routing update will delete an existing Source Loop by purging the malicious route entries in all routers involved. Then any correlated CTI problems and routing loops are prevented.

If the present route in the routing table is marked as unreachable and there is a valid alternative route to the same destination subnet with an indistinguishable metric, then the first incoming corresponding routing update would be rejected by RMTI. But the convergence time would not be appreciably impaired due to the rmti_checktime variable.

## 7. CONCLUSION

It has been shown that our new RMTI protocol can recognize loops and therefore offers better convergence properties than other distance vector routing protocols. On the other hand, it is fully compatible with RIP by evaluating the common RIP updates more carefully. Every router can thus recognize all loops starting and ending at the router. This loop knowledge is used together with the states of the routing table before and after a network change in order to decide on the acceptance or rejection of an incoming new routing update. It has been shown that our RMTI protocol can handle simple and complex topologies as well. Problems like counting to infinity can no longer appear because we can detect every routing update from a router via a loop back to the same router. Therefore RMTI does not need to calculate with a given hop-count-limit like standard RIP does. As RMTI is a solution to the routing loop problem, overhead prone techniques like flooding of routing updates are not necessary. Investigations in our test environment showed that RMTI can converge faster than other distance vector routing protocols. In contrast to link-state protocols, RMTI allows the administration of local routing policies, which is a powerful method in order to impact the traffic density in the network. As RMTI does only a few additional loop tests in comparison to RIP, the runtime complexity grows still linearly with the number of subnets in a network domain.

Further investigations will be done to find rules for the optimization of the timer adjustments in order to maximize the benefit of the new RMTI technique. We will offer our new protocol as a Quagga daemon for Linux in order to invite readers to try out this new routing protocol. As RMTI can prevent all problems triggered by routing loops, it is a crucial improvement on the distance vector approach.

## 8. REFERENCES

[1] Ch. Steigner, H. Dickel, and T. Keupen, *RIP-MTI: A New Way to Cope with Routing Loops*, in Proceedings of the Seventh International Conference on Networking (ICN 2008), Cancun, Mexico , 2008.

[2] F. Bohdanowicz, H. Dickel, and Ch. Steigner, *Metric-based Topology Investigation*, in Proceedings of the Eighth International Conference on Networking (ICN 2009), Gosier, Guadeloupe/France, 2009.

[3] C. Cheng, R. Riley, S.P.R. Kumar, and J. J. Garcia-Luna-Aceves, *A loop-free extended bellman-ford routing protocol without bouncing effect*, ACM Sigc. Symp. Commun. Arch. and Prot., pp. 224-236, 1989.

[4] J. Dike, *User Mode Linux*, Prentice Hall, 2006.

[5] B. Rajagopalan and M. Faiman, *A new responsive distributed shortest-path routing algorithm*, ACM Sigcomm Symposium Commun. Arch. and Protocols, pp. 237-246, 1989.

[6] Francois, P. Bonaventure, O., *Avoiding Transient Loops During the Convergence of Link-State Routing Protocols*, Transactions on Networking, IEEE/ACM Volume: 15, Issue: 6, Dec. 2007

[7] J.J. Garcia-Luna-Aceves, *Loop Free Routing Using Diffusing Computations*, IEEE Transactions on Networking, 1993.

[8] Hengartner, U., Moon, S., Mortier R., and Diot, C., *Detection and Analysis of Routing Loops in Packet Traces*, Proc. of 2nd Internet Measurement Workshop (IMW 2002), Marseille, France, November 2002

[9] Kirill Levchenko, Geoffrey M. Voelker, Ramamohan Paturi, and Stefan Savage , *XL: An Efficient Network Routing Algorithm*, Proc. Sigcomm 2008, August, 2008.

[10] G. Malkin, *RIP Version 2*, RFC 2453, 1998, URL: http://tools.ietf.org/html/rfc2453, 05.08.2009.

[11] J. Moy, *OSPF Version 2*, RFC 2328, 1998, URL: http://tools.ietf.org/html/rfc2328, 05.08.2009.

[12] D. Pei, X. Zhao, D. Massey, and L. Zhang, *A Study of BGP Path Vector Route Looping Behavior*, IEEE International Conference on Distributed Computing Systems (ICDCS), March, 2004.

[13] C. E. Perkins, *Ad hoc networking*, Addison-Wesley, Amsterdam 2001.

[14] C. E. Perkins, E. Belding-Royer, S. Das, *Ad hoc On-Demand Distance Vector (AODV) Routing*, RFC 3561, 2003, URL: http://tools.ietf.org/html/rfc3561, 05.08.2009.

[15] Jian Qiu, Feng Wang and Lixin Gao, *BGP Rerouting Solutions for Transient Routing Failures and Loops*, in Proceedings of MILCOM, October, 2006.

[16] Quagga home page, http://www.quagga.net/, 05.08.2009.

[17] Y. Rekhter, T. Li, S. Hares, *A Border Gateway Protocol 4*, RFC 4271, 2006, URL: http://tools.ietf.org/html/rfc4271, 05.08.2009.

[18] A. Schmid and Ch. Steigner, *Avoiding Counting to Infinity in Distance Vector Routing*, Telecommunication Systems 19 (3-4): 497-514, March - April, 2002, Kluwer Academic Publishers.

[19] VNUML Project home page, http://www.dit.upm.es/vnumlwiki, 05.08.2009, Technical University of Madrid (UPM).

[20] OpenWRT Project home page, http://www.openwrt.org, 05.08.2009.

[21] Andrew S. Tanenbaum, *Computer Networks*, 3rd ed., Prentice Hall PTR, 1996, pp. 358-359

[22] Zifei Zhong, Ram Keralapura, Srihari Nelakuditi, Yinzhe Yu, Junling Wang, Chen-Nee Chuah and Sanghwan Lee, *Avoiding Transient Loops Through Interface-Specific Forwarding*, Transactions on Networking, IEEE/ACM, Volume: 15, Issue: 6, Dec. 2007 Transactions on Networking, Dec. 2007

# Analyzing and Improving Reliability in Multi-hop Body Sensor Networks

Bart Braem, Benoît Latré, Chris Blondia, Ingrid Moerman, Piet Demeester

***Abstract*— Body Sensor Networks are an interesting emerging application of wireless sensor networks to improve healthcare and the Quality of Life. Current research has mainly focused on single-hop networks, although some works clearly show advantages of multi-hop architectures. In this paper, we model probabilistic connectivity in such multi-hop body sensor networks. Instead of using a circular coverage area, a more accurate model is defined based on the path loss along the human body. Further, we propose improvements to CICADA, a cross-layer multi-hop protocol that handles both medium access and the routing of data in BSNs. CICADA is slot-based and uses schemes to allocate these slots. Results for two reliability improvements are given: randomization of the schemes and repeating the schemes received from a parent node. We show that these improvements positively affect the throughput of the network and lead to fewer retransmissions while the energy consumption of the nodes is hardly influenced.**

***Index Terms*— health care, routing protocols, network reliability, wireless sensor networks, BSN, CICADA, cross-layer multi-hop protocol, medium access protocol, path loss model**

## I. Introduction

The recent development of intelligent (bio-) medical sensors and the tendency of miniaturization has lead to devices that can be worn on or implanted in the human body. The sensors are equipped with a wireless interface, enabling an easier application. These sensors send their data to a personal device (e.g. a PDA or a smartphone) which acts as a sink or as a gateway to health care. This type of network is called a *Wireless Body Sensor Network* or BSN [1]. These systems reduce the enormous costs of patients in hospitals as monitoring can occur real-time and over a longer period of time, even at home.

Recent studies have spoken out for the use of multi-hop routing in wireless on-body networks, where intermediate sensors may be used as relay devices in order to reach the personal device [2], [3]. This is needed as the path loss around the body is very high [4], [5]. Multi-hop networking leads to an increased connectivity of the network and lowers the energy consumption even further. Current protocols mainly address the energy consumption or lifetime of the network and to a lesser extent the reliability.

In this paper we will discuss techniques to enhance the reliability in a BSN. Due to the lack of an existing reliability model for a BSN, a framework is proposed that determines the link probability based on a lognormal distribution instead of assuming a circular coverage area. Doing so, a more accurate model of the network is obtained. The CICADA multi-hop protocol [6] is used as base protocol. This is a cross-layer protocol that sets up a data gathering tree and offers low delay and high energy efficiency. The reliability of this protocol is analyzed and modifications are proposed to increase the reliability. It is shown that the reliability can be improved without affecting the energy consumption of lifetime of the network. In addition, the combined effect of the solutions is analyzed. This paper is an extension of [14] where we briefly discussed the advantages of the proposed modifications. In this paper, a more comprehensive analysis is performed, both by simulations and analytically.

The remainder of this paper is as follows. Section II gives an overview of the related work and Section III explains the current design of CICADA. Section IV gives a method to model the reliability and discusses the impact on the protocol design. We use this reliability model in our simulations, to evaluate two proposed techniques to improve reliability: scheme randomization and repeating the schemes (overhearing). The simulation set up is discussed in Section V and the techniques are the topics of respectively Sections VI and VII. In Section VIII the combined effect of the solutions is analyzed. Finally, Section IX discusses the general applicability of our results and Section X concludes the paper and describes future work.

## II. Related Work

Although a lot of projects currently try to implement BSNs, few protocols have been developed. Focus lies either on single-hop communication [7] or on multi-hop routing for embedded devices where the prime criterion is the reduction of heat produced in the devices [8], [9]. These protocols only try to improve the energy efficiency as a second criterion, while the reliability or quality of service is overlooked. The issue of tissue heating is less important with body mounted devices as these can emit their heat to the air. Only a few protocols have been proposed for multi-hop routing in BSNs that improve the lifetime of the network. Both [10] and [11] propose a data gathering protocol that uses clustering to reduce the number of direct transmissions. They do not consider the delay of their protocol and are not optimized for BSNs as they where developed for regular sensor networks. CICADA [6] and its predecessor WASP [12] are tree based protocols that aim for high network lifetime and low delay.

To the knowledge of the authors, currently only two protocols exist that take into account reliability and QoS.

B. Braem and C. Blondia are with the PATS research group, Dept. of Mathematics and Computer Science, University of Antwerp — IBBT, Middelheimlaan 1, B-2020, Antwerp, Belgium, firstname.lastname@ua.ac.be

B. Latré, I. Moerman and P. Demeester are with the Broadband Communication Networks group, Dept of Information Technology, Ghent University — IBBT, Gaston Crommenlaan 8, bus 201, 9050 Gent, Belgium, firstname.lastname@intec.ugent.be

BodyQoS [13] addresses three unique challenges introduced by BSN applications. It uses an asymmetric architecture where most of the processing is done at the central device. Second, the authors have developed a virtual MAC (V-MAC) that can support a wide variety of different MACs. Third, an adaptive resource scheduling strategy is used in order to make it possible to provide statistical bandwidth guarantees as well as reliable data communication in BSNs. The protocol has been implemented in NesC on top of TinyOS.

In [14] the reliability of CICADA was evaluated and additional mechanisms were proposed in order to improve the reliability, such as the randomization of schemes and overhearing the control messages sent by siblings. In this paper, the concepts presented in [14] are investigated more thoroughly. The simulations are more elaborated and the efficiency is considered analytically.

## III. CICADA

### A. General Overview

CICADA is a cross-layer protocol as it handles both medium access and the routing of data. The protocol sets up a spanning tree in a distributed manner, which is subsequently used to guarantee collision free access to the medium and to route data toward the sink. The time axis is divided in slots grouped in cycles, to lower the interference and avoid idle listening. Slot assignment is done in a distributed way where each node informs its children when they are allowed to send their data by using a scheme. Slot synchronization is possible because a node knows the length of each cycle. In each cycle, a node is allowed to send all of its data to its parent node. CICADA is designed in such a way that all the packets arrive at the source in only one cycle. Routing itself is not complicated in CICADA as data packets are routed up the tree which is set up to control the medium access, no special control packets are needed.

A cycle is divided in a control and a data subcycle. The former is used to broadcast a scheme from parent to child, to let the children know when they are allowed to send in the data cycle. In the data subcycle, data is forwarded from the nodes to the sink. In each data subcycle, a contention slot is included to allow nodes to join the tree. New children hear the scheme of the desired parent and send a JOIN-REQUEST message in the contention slot. When the parent hears the join message, it will include the node in the next cycle. Each node will send at least two packets per cycle: a data packet or a HELLO packet and a scheme. If a parent does not receive a packet from a child for $N$ or more consecutive cycles, the parent will assume that the child is lost. If a child does not receive packets from its parent for $N$ or more consecutive cycles, the child will assume that the parent is gone and will try to join another node.

An example of communication in CICADA is given in Figure 2 for a network of 5 nodes as shown in Figure 1. The control and data subcycles can be seen clearly: the communication goes from sink to node in the control subcycle and from node to sink in the data subcycle. As only schemes are sent in the control subcycle, the slot length can be up to 10 times smaller in the control subcycle compared to the slot length in the data subcycle. This improves the energy efficiency of the protocol as the node switches its radio off after the control subcycle.

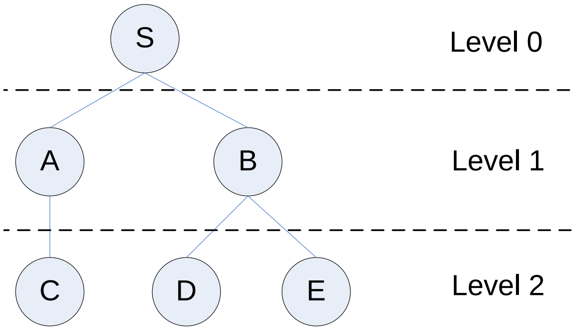


Fig. 1. Tree topology for a network of 5 nodes

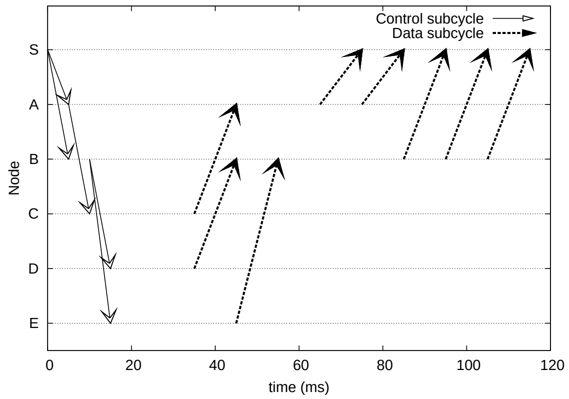


Fig. 2. Communication streams for the network in Figure 1. Notice the downstream and upstream cycles

### B. Algorithm

In order to inform its parent node of the number of slots a node needs, to send its own data and forward data coming from children, two parameters are calculated: $\alpha_n$ and $\beta_n$. The former gives the number of slots needed for sending data (including forwarded data), the latter gives the number of slots the node has to wait until it has received all data from its children. Based on the $\alpha_n$ and $\beta_n$ from its children, a node can calculate the slot allocation for the next cycle.

CICADA initially did not include reliability support. Two adaptations to add this are envisioned: scheme randomization and repeating the schemes received from a parent (also referred to as overhearing).

### C. Packet Formats

In order to be able to estimate the overhead of the solutions proposed below, the packet formats used in CICADA are described. In general, three types can be distinguished

TABLE I

PARAMETER VALUES FOR THE PATH LOSS MODEL (1)

| parameter | value LOS [4] | value NLOS [5] |
|---|---|---|
| $d_0$ | 10 cm | 10 cm |
| $P_{0,dB}$ | 35.7 dB | 48.8 dB |
| $\sigma$ | 6.2 dB | 5.0 dB |
| $n$ | 3.38 | 5.9 |

depending on the type of message sent: a control packet, a routing packet and a JOIN-REQUEST-message. The length of the node ID's is limited to 8 bits, allowing a network of 255 nodes. This is sufficient for a BSN.

Figure 3 shows the packet format for the different messages.

The routing packet is used for sending data to the next hop in the data subcycle. It contains the ID of the sending node (say node $n$) and the ID of the parent supposed to receive the message. Further, the node sends its $\alpha_n$ and $\beta_n$ to the parent followed by the data. This packet contains the ID where the data was originated, a message ID and the payload of the data. The length of the payload is variable and limited by the maximum packet size.

The control packet contains the ID of the sending node, followed by the control scheme and the data scheme. If bidirectional traffic is supported and data is sent during the control cycle, the `settings`-bit is set to 1. The data is then added after the data scheme.

The JOIN-REQUEST message only contains the ID of the sending node, the ID of the desired parent (Nexthop ID) and its $\alpha_n$ and $\beta_n$.

The HELLO-message is similar to a routing packet, without the data part. It is sent to ensure connectivity and buffer information propagation.

## IV. MODELING RELIABILITY

The path loss between the transmitting and receiving antenna for a BSN is subject of several studies. The line of sight (LOS) propagation was investigated in [4]. However, this model does not consider the communication between the back and torso for example nor does it take into account the curvature effects of the body. In [5] a higher path loss was found in non-line of sight (NLOS) situations around the torso. For our simulations, we will combine these models. Both models use the following semi-empirical formula for the path loss:

$$P_{dB} \ = \ P_{0,dB} \ + \ 10 \cdot n \cdot \log(d/d_0) \qquad (1)$$

where $P_{0,dB}$ is the path loss at a reference distance $d_0$ and $n$ is the path loss exponent, which equals 2 in free space. The parameter values for both models can be found in Table I.

In practice the average received power varies from location to location in an apparently random manner. This variation is well described by a lognormal distribution with standard deviation $\sigma$ and is called *shadowing* [15]. The magnitude of the standard deviation indicates the severity of signal fluctuations caused by irregularities in the surroundings of the receiving and transmitting antennas. It is crucial to account for this in order to provide a certain reliability of communication. This can be done by adding the shadowing component, represented by a zero-mean Gaussian random variable with standard deviation $\sigma$, $X_{\sigma,dB}$ to (1).

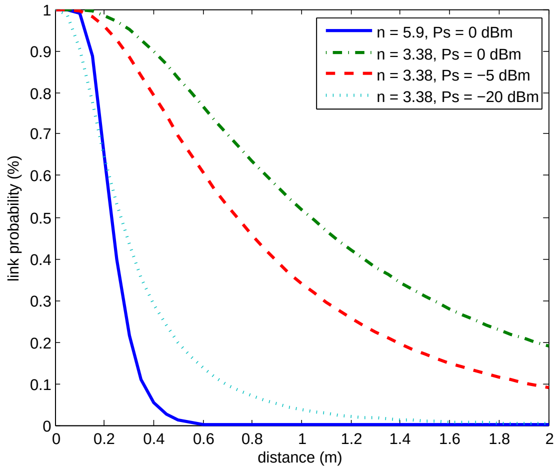


Fig. 4. Link probability for different path loss models and varying transmission power.

The received signal strength $P^j_{r,dB}$ at a node $j$ from a node $i$ sending with transmitting power $P^i_{s,dB}$ over a distance $d_{ij}$ can thus be written as:

$$P^j_{r,dB}(d_{ij}) \ = \ P^i_{s,dB} \ - \ PL_{dB}(d_{ij}) \ - \ X_{\sigma,dB} \qquad (2)$$

The condition for connectivity at the receiver is that $P^j_{r,dB}$ is higher than a certain threshold $P_{th}$ at the receiver. As a result, the probability $p(d_{ij})$ that two nodes $i$ and $j$ are connected can be formulated as [16]:

$$\begin{aligned} p(d_{ij}) \ &= \ \Pr\left[P^j_{r,dB}(d_{ij}) \ > \ P_{th}\right] & (3) \\ &= \ \Pr\left[X_{\sigma,dB} + \mu(d_{ij}) \ < \ 0\right] & (4) \end{aligned}$$

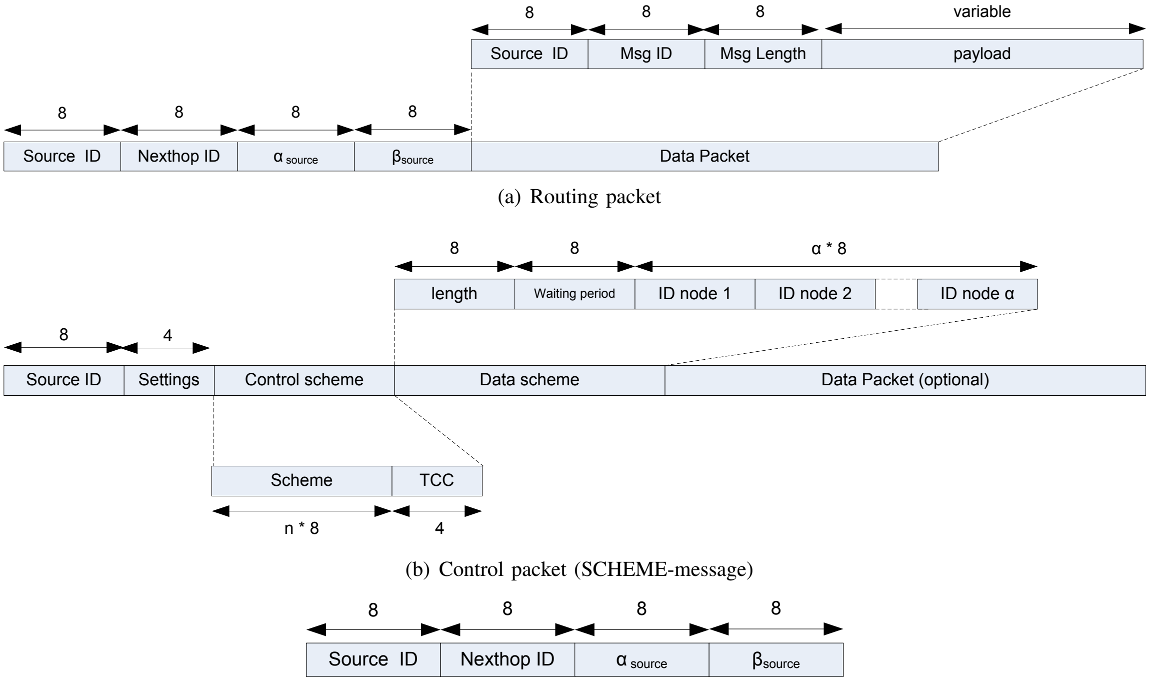
The left part can be seen as normally distributed with standard deviation $\sigma$ around the mean $\mu(d_{ij})$ where:

$$\mu(d_{ij}) \ = \ -P^i_{s,dB} + PL_{0,dB} + 10n\log_{10}(d_{ij}/d_0) + P_{th} \quad (5)$$

Consequently, (4) can be rewritten as

$$\begin{aligned} p(d_{ij}) \ &= \ \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{0} \exp\left[-\frac{(t \ - \ \mu(d_{ij}))^2}{2\sigma^2}\right] dt & (6) \\ &= \ \frac{1}{2} - \frac{1}{2}\,\mathrm{erf}\left(\frac{\mu(d_{ij})}{\sqrt{2\pi}\sigma}\right) & (7) \end{aligned}$$

The link probability for different path loss exponents is given in Figure 4. It is clear that the link probability also depends on the transmitting power ($P^i_{s,dB}$) and the receiving threshold. The latter can be defined using the parameters of the receiver. If its noise floor is -90 dBm and the desired signal-to-noise ratio is at least 20 dB, we can say that $P_{th}$ = -90 dBm + 20 dB = -70 dBm. The figure shows that if one wants a minimal link probability to ensure a certain reliability, the distance that can be covered is really small.



(a) Routing packet

(b) Control packet (SCHEME-message)

(c) JOIN-REQUEST-message

Fig. 3. Packet format in CICADA. The numbers indicate the length in bits. $\alpha_{source}$ is the $\alpha_n$ of node $n$ sending the packet.

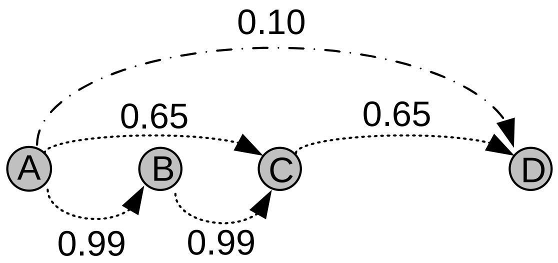


Fig. 5. Example of a connection probability in a single-hop and a multi-hop scenario. The distance between node $A$ and $B$ is 10 cm and between node $C$ and $D$ 20 cm

Due to the probabilistic connectivity, the boundaries of the area where the signals are received can no longer be represented by a circle with the sender in the middle. The boundaries fluctuate. This also means that bidirectionality is no longer guaranteed, which will complicate protocol design.

Figure 4 shows that for reliable communication, the covered distance is rather low in BSNs. From (7) we can derive the communication reliability when using a single-hop or a multi-hop architecture. In order to develop an intuition for why there might be room for improvement in multi-hoprouting, it is helpful to consider Figure 5. Different nodes are placed on one line and different routes are shown for communication between nodes $A$ and $D$. The numbers above the communication links show the link probability between the two nodes using (3) and the variables of Table I. At one extreme, node $A$ could send directly to $D$ in one hop and at the other extreme, $A$ could use the 3-hop route through $B$ and $C$. In the example, it is clear that the 3-hop communication has a communication probability of 63.7% whereas the single-hopcommunication only is 10%. On the other hand, in multi-hopcommunication nodes $C$ and $D$ will hear many of the packets sent from $A$ to $B$ and it is wasteful that node $B$ forwards these packets. In this example, the single-hop communication is more energy efficient but less reliable than the multi-hop communication and vice versa. This shows the trade-off between the reliability and the energy efficiency.

Using the formula for link probability, the condition to determine whether or not multi-hop communication should be used in terms of reliability if there are $n$ intermediate hops can be written as $p\left(\frac{d}{n+1}\right)^{n+1} > p(d)$. When applying this inequality, it turns out that the multi-hop path has the highest reliability. This is due to the fact that $p(d)$ is a monotonically decreasing function, as can be seen in Figure 4. One has to keep in mind that this holds as long as the intermediate hops are placed on the path between the sender and destination. Further, as the probability is a statistical value this will not always be the case. At a given moment in time, the reliability over the multi-hop path can be lower as for example the path between node $C$ and $D$ may experience high packet loss temporarily. Of course, when nodes $A$ and $D$ are sufficiently close to each other, the reliability of direct communication will be high enough to use it and the gain of using multi-hop communication will be negligible.

Based on the previous, it can be concluded that, in order to increase the reliablity, it is a good design choice to use a multi-hop architecture when developing a protocol for wireless BSNs. This view is also supported by [3] where the reliability was experimentally validated. However, the energy efficiency also needs to be taken into account. Preliminary studies in [2] have shown that lifetime of the network can be increased considerably by letting nodes cooperate intelligently and by introducing extra intermediate relay devices. A more detailed analysis is subject of future research. Summarizing, new protocols for WBANs should take both transmission reliability and the energy efficiency of the nodes into account. The rest of this paper we will focus on improving CICADA's reliability.

## V. Simulation set up

Our simulations were performed in a newly developed simulator written in Ruby [17], in order to have complete control of the simulation environment and to avoid overhead of classical simulators that are more tailored toward testing of routing protocols or mac protocols in large scale networks with specific data sources. The simulator correctness is verified by a large set of unit tests with a test coverage of 99.8%, as calculated by the Ruby rcov coverage tool, and a set of algorithm tests in a number of scenarios. The code has a number of built-in triggers that signal erroneous states, combined with the high number of random tests performed this gives us confidence in the simulator. Future work will include comparing results with the performance of classical sensor network MAC protocols in order to have even more confidence.

The path loss model (2), the link probability (7) and the improvements are implemented. The simulator was used to analyze the changes to the protocol. The nodes were randomly placed in a 2 by 2 meter area where the sink is positioned in the center. The distances between the nodes is at most 40 cm in a connected topology, i.e. every node is within transmission range of at least one other node so there is always a path to the sink. Nodes start randomly, they do not join the network all at once. All simulations were run during 10.000 slots for 1000 randomly generated topologies, while making sure the same topologies are used in comparisons. Each node generates one packet after a fixed number of data intervals. This data interval is equal to the number of nodes in the network. Thus, when one data interval is chosen, all nodes will generate data packets, at least one per cycle. The more data intervals are chosen between each data generation, the fewer the number of packets that will be sent.

## VI. Scheme Randomization

### A. Concept and Algorithm

In order to understand the benefits of using randomization, consider Figure 6 where a part of a topology is shown when the tree is set up. As can be seen, it might occur that two nodes $a$ and $b$ can hear each other, but actually have different parents, $c$ and $d$ respectively. This can happen when the link between $a$ and $c$ is more reliable than the link between $a$ and $d$. When the schemes are not variable, i.e. the node that joins first always sends first in each cycle and so on, it might well happen that nodes $a$ and $b$ will always interfere while sending their data. In order to try to decrease the overall interference, schemes are randomized. I.e. the sequence in which the children are allowed to send is randomized, while still respecting the rules of the CICADA protocol. The implementation is straightforward.

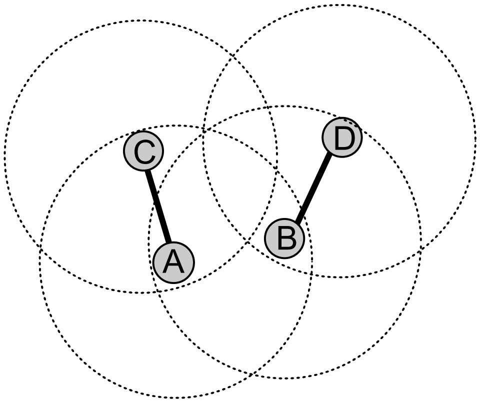


Fig. 6. Example of a topology where randomization can be useful. The dashed circles represent the node's send range. The straight line shows the established parent-child relation. It is clear that nodes A and B will interfere.

### B. Simulation Results

Results of the simulations are shown in Figures 7, 8 and 9 where the size of the network is varied from 5 to 30 nodes. The values represent the improvement in percentage between the results without and with randomization. We compare the number of packets received by the sink and the number of retransmissions. We also look at the number of slots the radio was on, averaged out over the number of runs, to study the impact on network lifetime. In order to take into account network saturation effects, we also vary the data generation interval to 1, 5 and 9 times the number of nodes in the network. E.g., for 7 nodes and a data interval 5, every 35 slots a data packet is generated at each node.

It can be seen that scheme randomization has a minor impact on the performance of the system. The number of packets that can be received by the sink stays constant for almost all network sites and data intervals. Yet, it can be seen that the number of retransmissions is larger. We do currently not understand why this behaviour arises, as there are no dependencies on node order in the system. In case of bad links, we would expect these effects to be cancelled by the number of simulations. For larger networks and larger data intervals, the absolute increase can be explained by the higher number of transmissions in the network when randomization is used. For small networks, little effect was found as the parent nodes have few child nodes to randomize.

It is important to notice that the figure also shows that the average time a radio is on, is slightly larger with randomization. This is to be expected as more packets are sent.

Overall, we think scheme randomization should be used to avoid fixed bad links, but we believe the protocol should be studied more to look into the retransmission increase.

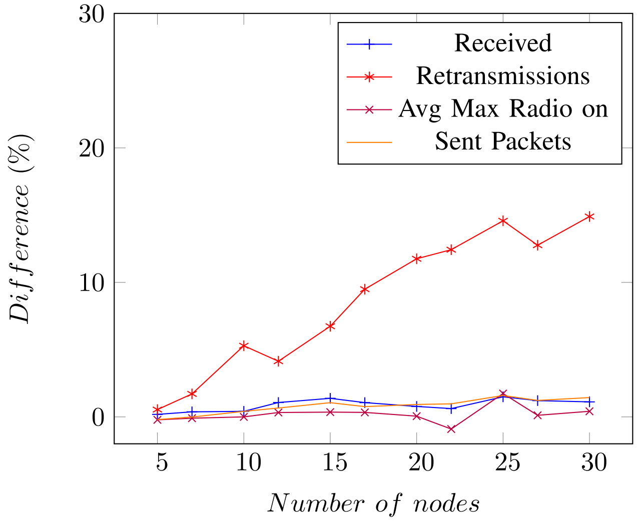


Fig. 7. Difference between whether the scheme randomization was used or not. Data interval: 1.

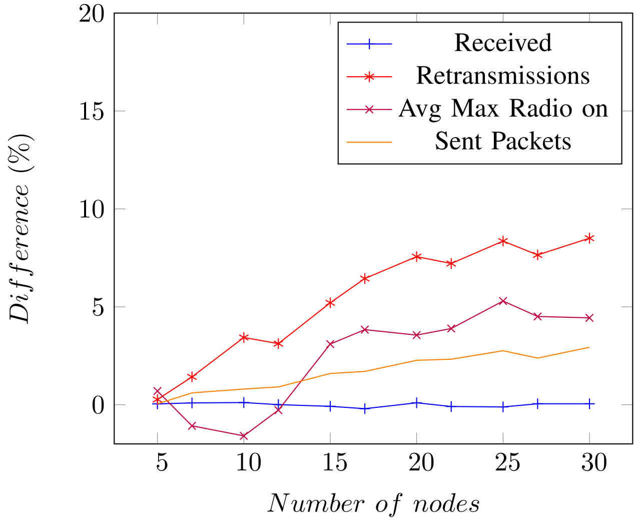


Fig. 8. Difference between whether the scheme randomization was used or not. Data interval: 5.

## VII. Overhearing

During our simulations we noticed that, from time to time, nodes miss a scheme packet from their parent, because of a link that is not very stable. The result is that this node and all nodes below it cannot do anything and must have their radio on until the next cycle. In order to tackle this problem, a child node repeats the scheme of its parent when it sends its own scheme, so siblings can exploit this information if they missed their parent's scheme. It is a way of avoiding the dependency on just one packet. Nodes now have multiple opportunities to synchronize their state on the network. This increases the reliability and the energy efficiency, as nodes that were not synchronized because of packet loss before now can overcome those inefficiencies.

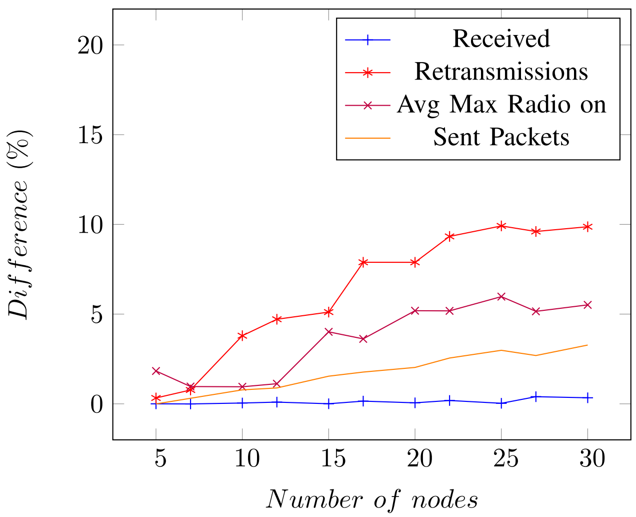


Fig. 9. Difference between whether the scheme randomization was used or not. Data interval: 9.

### A. Algorithm

In order to do overhearing, a node will not power down its radio when it has heard nothing within an expected number of slots. When receiving scheme packets, it will overhear packets from all nodes instead of dropping packets not coming from its parent. When receiving a packet, it will check whether the node transmitting the packet has the same parent. It will record the new control scheme with the control subcycle and data subcycle length and the current slot number. After correcting some offsets in the original control scheme packet, it can be passed to the routines responsible for doing regular processing of parent schemes.

### B. Analysis

A critical phase in the CICADA protocol is the control subcycle. If a node misses packets in this phase, i.e. it misses scheme messages, energy will be wasted as the node will keep its radio on until the next cycle. The impact of loss of other packet types is less critical. As a result, the overhearing algorithm only focuses on scheme packets.

When a node overhears a scheme packet, it can perform a recovery. Two types of recovery can be defined. In the first one, a node is not capable of transmitting its own scheme packet but it is aware of the current scheme. This means that the node can turn its radio off according to the scheme, but its children will not receive a scheme and cannot go to sleep mode. This state is referred to as partial recovery. In the second type of recovery, the node can transmit its scheme to its children. The children then know when to sleep. This is called full recovery.

In the following analysis, we will calculate the probability that a node $n$ is capable of sending its scheme by performing a full recovery and the probability that the node can do a partial recovery.

The packet reception probability between two nodes $x$ and $y$ is represented as $P_{x,y}$. In this analysis, it is assumed that all links are identical and that all the reception probabilities

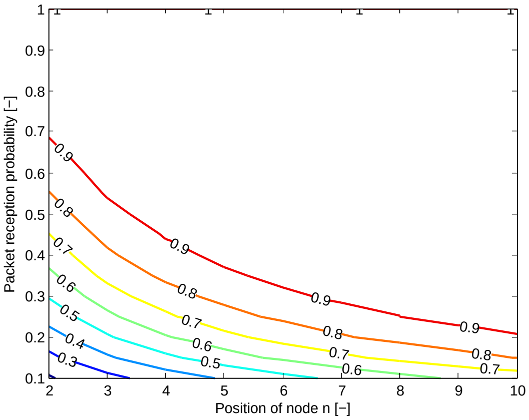


Fig. 10. Probability that node $n$ is capable of hearing its parent's scheme or to do a full recovery, calculated with (9). The curves connect the points with the same probability. The x-axis is the position of node $n$ in the control sub cycle, the y-axis is the packet reception probability. It can be seen that overhearing is useful.

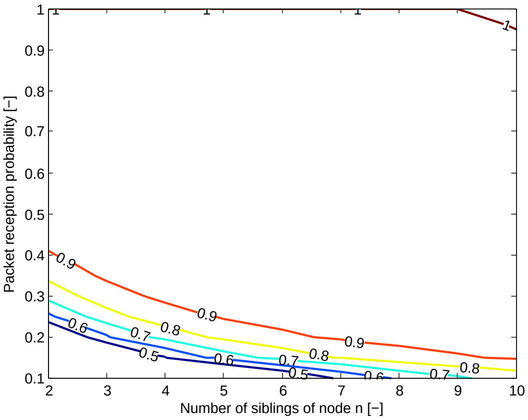


Fig. 11. Probability that node $n$ is capable of hearing its parent's scheme or to do a full recovery, calculated with (12). The curves connect the points with the same probability. The x-axis is the number of siblings of node $n$, the y-axis is the packet reception probability. It can be seen that overhearing is useful when more nodes have the same parent.

are independent and identically distributed. Hence, $P_{x,y}$ can be written as $P$. Further, it is assumed that node $n$'s parent node has $N+1$ children.

The probability that no recovery is needed simply is $P$, namely the probability that the scheme packet from the parent is received. This means that for a good link, overhearing will not even start in many cases. Similar, the probability that the node does not receive its parent's packet, is $1-P$.

The probability that no scheme packet is received and that neither full nor partial recovery is possible, given $N$ siblings, is then

$$(1-P) \times (1-P)^{N} \tag{8}$$

This means that the node does not receive the scheme packet from its parent or from any of its sibling. Thus, when a link is very bad, the overhearing success probabilities depend on the number of siblings. When a node has more siblings, they can help cooperate and recover the state.

Let's define the position of a node in the control subcycle as $C$, where $0 < C \leq N+1$. Further assume that the node is able to perform partial recovery, thus the node has overheard the retransmitted scheme of one of its siblings that is positioned before the node in the control subcycle. Then, the probability that a node is able to send its scheme to its children is given by

$$P \, + \, (1-P) \times (1-(1-P)^{C-1}) \tag{9}$$

This is shown in Figure 10 where the avantage of using overhearing can be clearly seen. For example, when node $n$ is on position 5, the probability that the node can send its own scheme rises to more than 90%, even when the packet reception probability is a little bit less than 50%. Further we can see that if the node's position is early in the control subcycle, the probability of being able to perform full recovery is smaller.

Given a uniform distribution of the position of a node in the control subcycle, the average full recovery probability is given as

$$\frac{1}{N+1}\sum_{C=1}^{N+1} P + ((1-P) \times (1-(1-P)^{C-1})) \tag{10}$$

$$= P \, + \, (1-P) \times (1 - \frac{\sum_{C=1}^{N+1}(1-P)^{C-1}}{N+1}) \tag{11}$$

$$= P \, + \, (1-P) \times (1 - \frac{1-(1-P)^{N+1}}{(N+1)P}) \tag{12}$$

This is shown in Figure 11. The more siblings a node has, the higher the probability of doing a full recovery.

The probability that a node at position $C$ can do partial recovery, is then given as

$$(1-(1-P)^{N+1-C}) \tag{13}$$

Thus, the probability that node $n$ receives its parent's scheme and can turn its radio off when possible in the data cycle is given by

$$\begin{gathered} P \, + \, (1-P) \times [(1-(1-P)^{C-1}) + \\ \ldots + (1-P)^{C-1} \times (1-(1-P)^{N+1-C})] \end{gathered} \tag{14}$$

For partial recovery, once again the number of siblings is important. When this is high, the number of nodes behind the node will generate a high probability that partial recovery is still possible.

### C. Overhead

This solution seems very easy but also increases the overhead. In order to analyze the overhead, we need to know the size of a control packet. It is assumed that the length of an address is 8 bits. This allows us to identify 255 nodes in the network, which is sufficient for a BSN. Further, it is assumed that a node has $x$ children. In a control packet, a parent nodes sends the scheme for the control subcycle as well as for the data subcycle. A control packet is made up from the following elements, as can be seen in Figure 3 :

1) ID sender: 8 bits
2) Control Scheme: $x$ * 8 + waiting period (8 bits)
3) Control subcycle length (settings) + tree depth (8 bits together)
4) Data scheme: $x$ * 8 + waiting period (8 bits) + length field (8 bits)

This gives a total of $40 + 2 \cdot x \cdot 8$ bits or $5 + 2 \cdot x$ bytes. When the scheme of the parent is included, an additional of $4 + 2 \cdot x_p$ bytes are added, where $x_p$ indicates the number of children of the parent. If we assume that in a network each node has a maximum of 10 children, the length of the control packet will change from 25 bytes to 49 bytes. This means that the length of the slot size in the control cycle needs to be increased, which will have an impact on the energy efficiency. However, these influences are minor as long as the length of the control slots is more than ten times smaller than the length of the data slots (proof omitted due to reasons of brevity). Hence, if a data slot can hold a message of 500 bytes, the influence of adding your parent's scheme is minor.

### D. Simulation Results

Again, simulations were performed in our own simulator, with the same settings. This time, the data interval was set to 1, 3, 5, 7 and 9 times the node count to take into account possible network saturation effects. The results are shown in Figures 12,13,14,15 and 16. We see that the sink receives about the same amount of data, regardless of the data interval. The number of sent packets increases slightly with the data interval, as there is more room to use the recovered slots. But while sending more packets, nodes clearly have their radio switched off more, especially when the number of nodes in the network is large. In that case, the number of siblings to overhear control packets is larger, thus increasing the probability of recovery. This clearly shows that overhearing to do recovery works very well.

In order to evaluate the overhead, Figure 17 shows the ratio of the number of nodes in the schemes. As expected, the number of nodes in the schemes increases with the number of nodes in the network.

## VIII. Combined solutions

In this section, the two mechanisms are combined and it is investigated how they influence each other. The results are shown in Figures 18, 19 20, 21 and 22. In saturated networks, where the data interval is small, the combined solution suffers from the retransmissions and advantages are not clear. In non-saturated networks, the combination of both solutions clearly works: the number of retransmissions and the use of the radio decreases. For larger networks, the number of retransmissions once again increases due to the large number of links.

In Figure 17 it can be seen that the combined solution has little influence on the overhead caused by repeating the parent's scheme. This is expected as the tree structure is not fundamentally changed by any of the mechanisms.

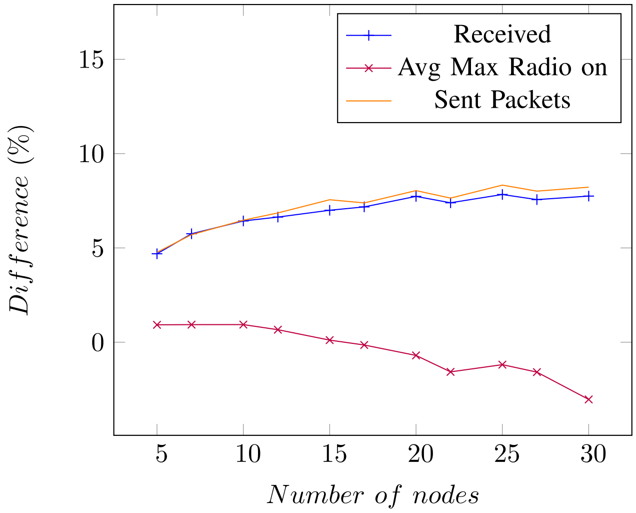


Fig. 12. Difference between whether or not overhearing was used. Data interval: 1.

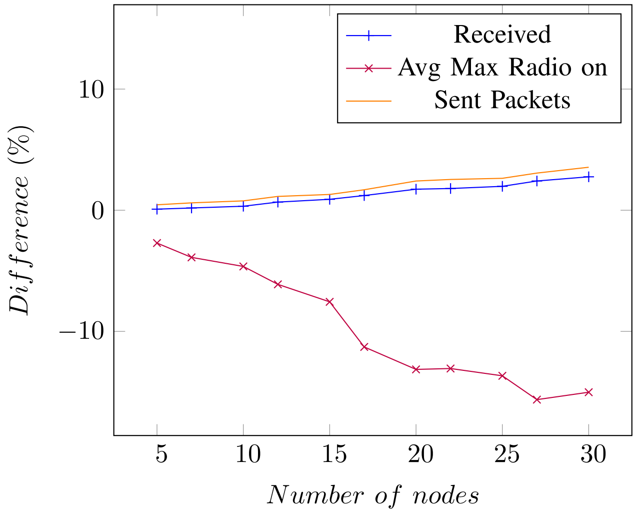


Fig. 13. Difference between whether or not overhearing was used. Data interval: 3.

## IX. Applicability of Results

One should notice that although both approaches are implemented for the CICADA protocol, they are applicable to other sensor network protocols as well. Scheme randomization comes down to avoiding fixed allocation of slots, as this can become a fixed source of interference. Especially distributed TDMA schemes should avoid this.

The idea of overhearing is also interesting for other protocols. Depending on just one packet to synchronize one or more nodes is dangerous because of packet loss. Retransmitting the required protocol information does include an overhead, but it ensures better reliability. Nodes really collaborate to make sure all nodes in the network can send their data properly.

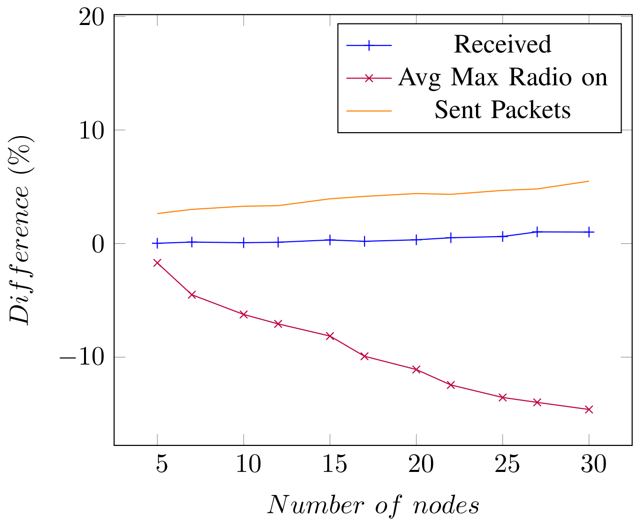


Fig. 14. Difference between whether or not overhearing was used. Data interval: 5.

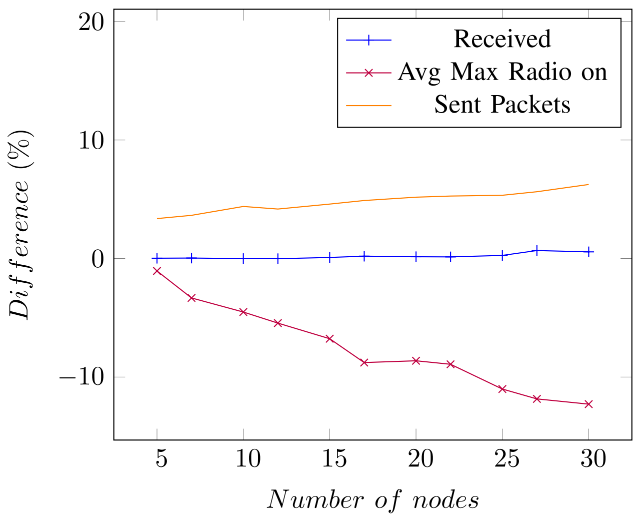


Fig. 16. Difference between whether or not overhearing was used. Data interval: 9.

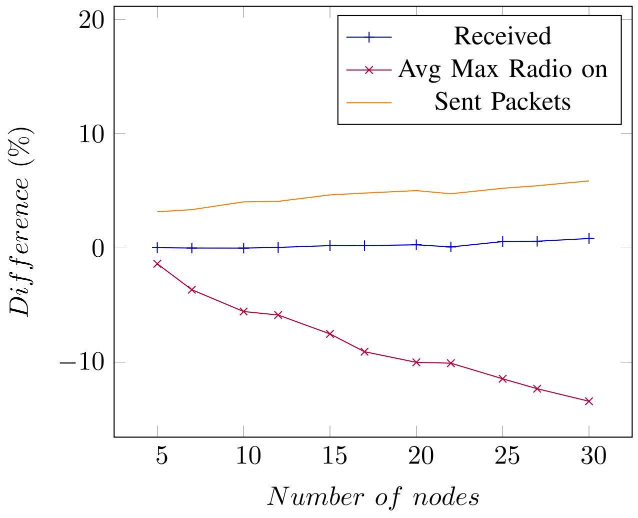


Fig. 15. Difference between whether or not overhearing was used. Data interval: 7.

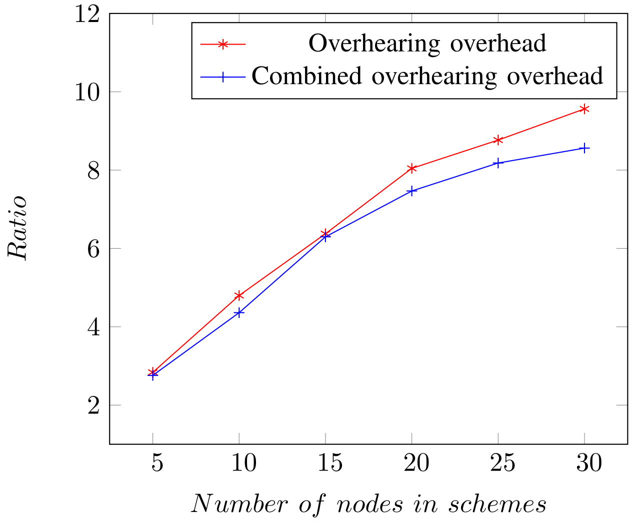


Fig. 17. Evaluation of overhearing overhead (with/without)

## X. Conclusions and Future Work

In this paper, we have presented two mechanisms to improve the reliability of CICADA, a multi-hop protocol for BSNs.

First, we have modeled the reliability of a link in a BSN. This was done based on path loss models available in literature. The proposed model uses a lognormal distribution for determining the range of a node instead of a circular coverage area. Doing so, a more realistic view of the network is obtained. This model was subsequently used for evaluating the proposed reliability mechanisms. The scheme randomization does lead to better results, although the number of retransmissions increases for reasons that are not clear. Adding the parent's scheme to the control message increases the reliability even further.

In the future we will further look into the randomization effects, an explanation for the increase in the number of retransmissions has to be found.

We will also test our simulator for more protocols, to be able to validate it completely. We then will try incorporating the improvements into those protocols as well, to study effects there. We also consider releasing the simulator as a fast, open source alternative to existing general simulators.

## Acknowledgments

This research is partially funded by the Fund for Scientific Research – Flanders (F.W.O.-V.,Belgium) project G.0531.05 (FWO-BAN) and by the Flemish IBBT project IM3.

## References

[1] C. Otto, A. Milenkovic, C. Sanders, and E. Jovanov, "System architecture of a wireless body area sensor network for ubiquitous health monitoring," *Journal of Mobile Multimedia*, vol. 1, no. 4, pp. 307–326, 2006.

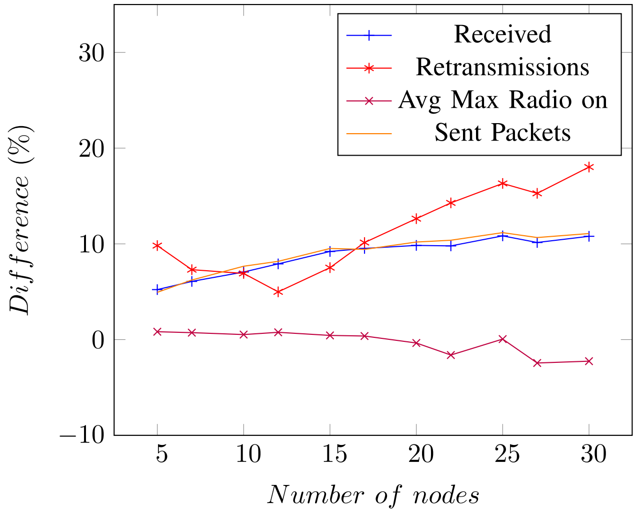


Fig. 18. Evaluation of combination of scheme randomization and overhearing (with/without), data interval: 1

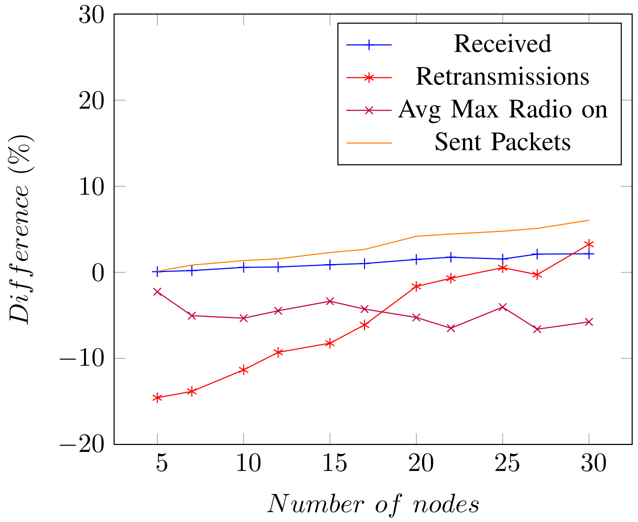


Fig. 19. Evaluation of combination of scheme randomization and overhearing (with/without), data interval: 3

[2] B. Braem, B. Latré, I. Moerman, C. Blondia, E. Reusens, W. Joseph, L. Martens, and P. Demeester, "The need for cooperation and relaying in short-range high path loss sensor networks," in *Accepted at 2007 International Conference on Sensor Technologies and Applications (SENSORCOMM 2007)*, OCT 2007.

[3] A. Natarajan, M. Motani, B. de Silva, K.-K. Yap, and K. C. Chua, "Investigating network architectures for body sensor networks," in *HealthNet '07: Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*. New York, NY, USA: ACM, 2007, pp. 19–24.

[4] E. Reusens, W. Joseph, G. Vermeeren, and L. Martens, "On-body measurements and characterization of wireless communication channel for arm and torso of human," in *4th Internation Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, March 2007, pp. 264–269.

[5] A. Fort, C. Desset, P. De Doncker, P. Wambacq, and L. Van Biesen, "An ultra-wideband body area propagation channel model - from statistics to implementation," *IEEE Trans. Microwave Theory and Tech.*, vol. 54, no. 4, pp. 1820–1826, Apr. 2006.

[6] B. Latré, B. Braem, I. Moerman, C. Blondia, E. Reusens, W. Joseph, and P. Demeester, "A low-delay protocol for multihop wireless body area networks," in *Mobile and Ubiquitous Systems: Networking & Services, 2007 4th Annual International Conference on*, Philadelphia, PA, USA, August 2007.

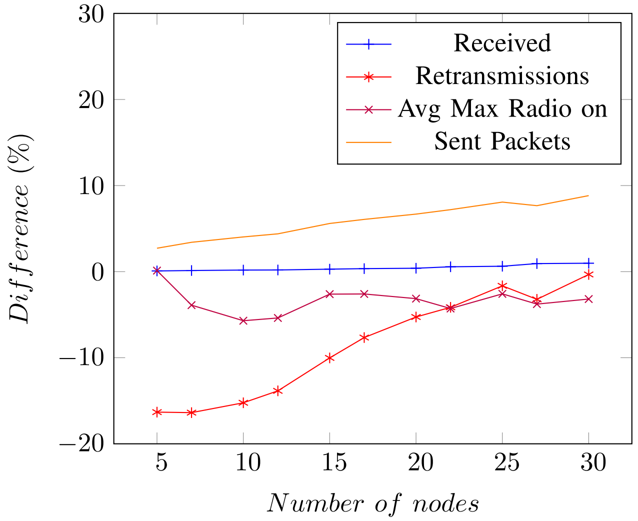


Fig. 20. Evaluation of combination of scheme randomization and overhearing (with/without), data interval: 5

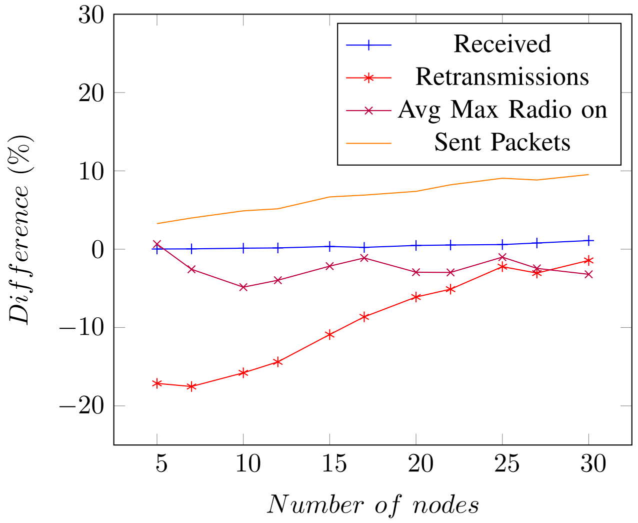


Fig. 21. Evaluation of combination of scheme randomization and overhearing (with/without), data interval: 7

[7] H. Li and J. Tan, "Heartbeat driven medium access control for body sensor networks," in *HealthNet '07: Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*. New York, NY, USA: ACM, 2007, pp. 25–30.

[8] A. Bag and M. A. Bassiouni, "Energy efficient thermal aware routing algorithms for embedded biomedical sensor networks," in *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, Vancouver, BC,, Oct. 2006, pp. 604–609.

[9] D. Takahashi, Y. Xiao, and F. Hu, "Ltrt: Least total-route temperature routing for embedded biomedical sensor networks," in *IEEE Globecom 2007*, November 2007.

[10] T. Watteyne, . Augé-Blum, M. Dohler, and D. Barthel, "Anybody: a self-organization protocol for body area networks," in *Second International Conference on Body Area Networks (BodyNets)*, Florence, Italy, 11-13 June 2007. 2007.

[11] M. Moh, B. J. Culpepper, L. Dung, T.-S. Moh, T. Hamada, and C.-F. Su, "On data gathering protocols for in-body biomedical sensor

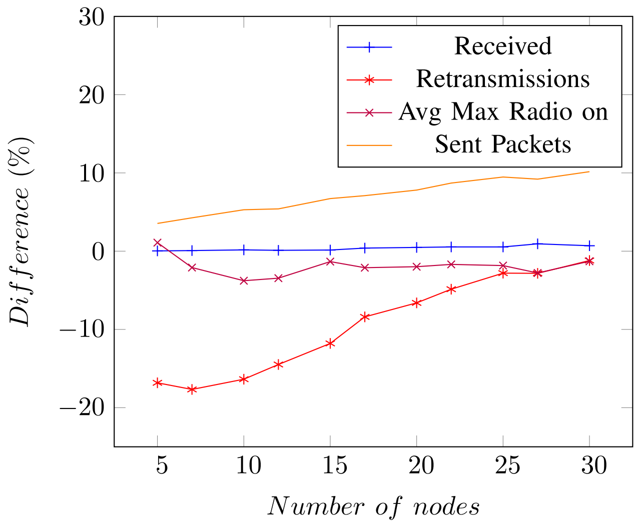


Fig. 22. Evaluation of combination of scheme randomization and overhearing (with/without), data interval: 9

networks," in *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, vol. 5, Nov./Dec. 2005.

[12] B. Braem, B. Latré, I. Moerman, C. Blondia, and P. Demeester, "The Wireless Autonomous Spanning tree Protocol for multihop wireless body area networks," in *Proceedings of the First International Workshop on Personalized Networks*. San Jose, California, USA: ICST, 2006.

[13] G. Zhou, J. Lu, C.-Y. Wan, M. Yarvis, and J. Stankovic, "Bodyqos: Adaptive and radio-agnostic qos for body sensor networks," April 2008, pp. 565–573.

[14] B. Braem, B. Latre, C. Blondia, I. Moerman, and P. Demeester, "Improving reliability in multi-hop body sensor networks," Aug. 2008, pp. 342–347.

[15] Saunders, *Antennas and propagation for wireless communication systems*. West Sussex, England: Wiley, 1999.

[16] R. Hekmat and P. V. Mieghem, "Connectivity in wireless ad-hoc networks with a log-normal radio model," *Mobile Networks and Applications*, vol. 11, no. 3, pp. 351–360, 2006.

[17] Ruby Programming Language [online] http://www.ruby-lang.org/.

# A Traffic Engineering proposal for ITU-T NGNs using Hybrid Genetic Algorithms

Alex Vallejo[1], Agustín Zaballos[1], David Vernet[1], Albert Orriols-Puig[1] and Jordi Dalmau[2]

[1] Enginyeria i Arquitectura La Salle
Universitat Ramon Llull (URL)
Barcelona, Spain
{avallejo, zaballos, dave, aorriols}@salle.url.edu

[2] Abertis
Barcelona, Spain
jordi.dalmau@abertistelecom.com

***Abstract*—Routing optimization is a key aspect to take into account when providing QoS in next generation networks (NGN), especially in access networks. The problem of weight setting with conventional link state routing protocols for routing optimization has been studied in order to adjust link's utilization and it has been object of study by a few authors. Among different approaches, GAs have been devised as one of the most appealing methodologies to tackle this problem since it becomes NP-hard when applied to large networks. In particular, some authors have used hybrid GAs (memetic GAs) which incorporate local search procedures in order to optimize the GA results.**

**This paper has proposed and implemented the integration of routing optimization using HGA with the ITU-T architecture for QoS resource control in Next Generation Networks (NGN). The implementation has been done over an IPv6 Linux testbed with OPSPv3 using the ITU-T proposed COPS-PR protocol for the policy delivery, in this case the weight setting delivery.**

***Keywords- QoS; PBNM; ITU-T; NGN; traffic engineering; routing optimization; OSPF; IPv6; hybrid genetic algorithm; local search***

## I. INTRODUCTION

The Telecommunication Standardization Sector of the International Telecommunications Union organization (ITU-T) has developed a generic end-to-end architecture for QoS resource control in Next Generation Networks (NGNs) [1]. This architecture proposes a centralized management of quality of service (QoS) through policy based network management (PBNM). For the intra-domain's policy delivery the ITU-T recommends among others the use of COPS-related protocols for the resource and admission control functions (RACF), which carries out the resources and admission control of the transport subsystem (QoS) within access and core NGNs.

One of the key factors when providing QoS is traffic engineering, given the overflows that certain links may have due to fluctuating traffic demands which can occur even in well dimensioned networks [2]. In order to tackle this problem, routing optimization aims at the optimization of networks so that more traffic can be routed in providing a possible solution to this problem. One way of achieving it is to modify the link weights and therefore the metrics. Depending on the dimensions of the topology, this weight setting problem may become a NP-hard problem [3], which can be solved through heuristics with artificial intelligence techniques. Among the different possible approaches, GAs have been shown to be appealing techniques to solve those type of NP-hard problems [4][5].

The purpose of this paper is to obtain a GNU/Linux system to optimize routing of NGNs with HGA algorithms using ITU-T specifications for QoS policy delivery. In order to optimize routing in NGNs with the aforementioned requirements we propose to use a centralized architecture, where a decision server (RACF entity) applies an offline routing protocol over the known network topology and uses an hybrid genetic algorithm (HGA) to decide the optimal weight setting of the links, which are later delivered to the nodes by the Common Open Policy Service (COPS) protocol [6] as it can be seen in Fig. 1. Thus using the centralized architecture recommended by the ITU-T. This paper extends a previous work by the authors [7] by integrating the ITU-T architecture and applying the results to a GNU/Linux testbed.

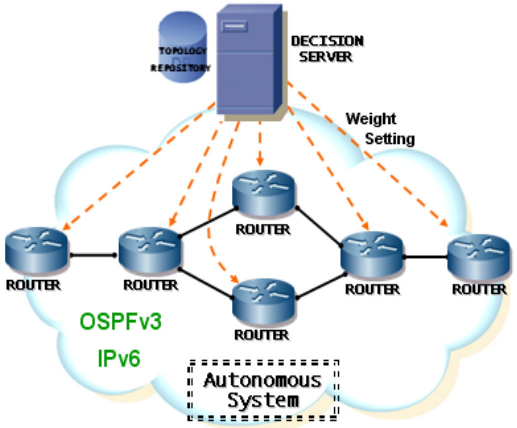


Figure 1. Network architecture for an offline routing optimization in NGNs

Therefore, the paper is structured as follows: In section II the ITU-T's RACF entity for QoS resource control management is introduced. In section III the routing fundamentals needed to understand the principles of the OSPF weight setting problem are described, in section IV the genetic algorithms (GA) for its application on routing optimization are introduced, and in section V the effects of a local search algorithm applied to the GA are illustrated. In section VI most relevant HGA algorithms are evaluated over a unique scenario which provides us with comparative data. In section VII a routing optimization proposal according to ITU-T specifications to be implemented in a GNU/Linux testbed is presented and finally, in section VIII the conclusions are provided.

## II. QoS MANAGEMENT IN ITU-T'S NGNS

The generic end-to-end architecture of the ITU-T for the QoS resource control in NGNs has been developed summarizing the local efforts of different agents in their respectively field (3GPP, DSL forum, WiMAX forum, CableLabs, etc.) and the ETSI-TISPAN's generic access networks architecture.

This architecture incorporates the IP Multimedia Subsystem (IMS) architecture, developed by 3GPP, as a support of session-based services and other session initiation protocols (SIP) [8]. Therefore the QoS management architecture proposed by ITU-T is completely integrated and interoperable with IMS and can provide the management of new services and multimedia communications through diverse NGNs.

The ITU-T NGN architecture presents a stratified division between the transport plane in the lower layer and the service plane in the upper layer with a session control plane in the intermediate control sub layer (Fig. 2). The transport plane supports the transport functions and the resource and admission control functions defined in the RACF entity [1]. This entity is capable of managing the end-to-end QoS through both core and access heterogeneous networks and therefore is responsible of managing the QoS resources of every domain.

The resource and admission control functions (RACF) provide the service layer (in concrete to the SCF) with an abstract vision of the network infrastructure through a unique contact point. Thus when a service request is received from a user, the current network resource usage status is checked to see whether the requested service can be offered with a guaranteed QoS.

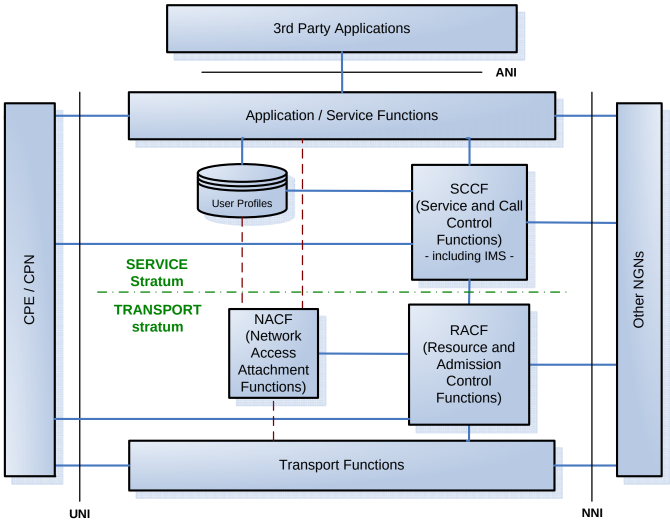

3rd Party Applications
ANI
Application / Service Functions
User Profiles
SCCF
(Service and Call Control Functions)
- including IMS -
SERVICE Stratum
TRANSPORT stratum
NACF (Network Access Attachment Functions)
RACF (Resource and Admission Control Functions)
Transport Functions
CPE / CPN
Other NGNs
UNI
NNI

Figure 2. ITU-T NGN Framework Architecture

### A. The RACF architecture

The RACF entity has adopted the policy-based network management in order to provide an efficient and centralized system to control the QoS of each domain and to provide an end-to-end QoS management tool for the end-to-end multimedia sessions which traverse domains without service stratum. This entity carries out the policy-based physical resource control, establishes the availability of these resources, decides on admission and applies the controls required to enforce the accomplishment of the policy decisions.

RACF uses reference points to manage the negotiated QoS through the session signaling and the flow control at a network level (Fig. 3). This architecture is presently under discussion and therefore some of the reference points are still to be specified. Even though different international organizations (ITU-T, ETSI-TISPAN, MSF, etc.) have proposed many alternative protocols to the network policies delivery in use in the Rw, Rc and Rn interfaces, industry has not opted for any of them.

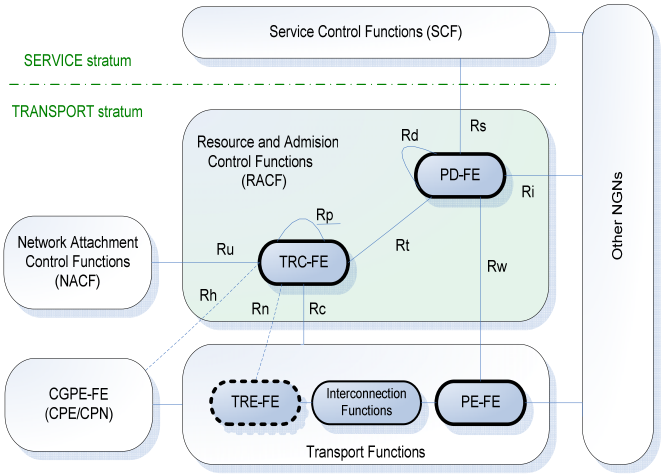

Service Control Functions (SCF)
SERVICE stratum
TRANSPORT stratum
Resource and Admision Control Functions (RACF)
Rd
Rs
PD-FE
Ri
Rp
Network Attachment Control Functions (NACF)
Ru
TRC-FE
Rt
Rw
Rh
Rn
Rc
CGPE-FE (CPE/CPN)
TRE-FE
Interconnection Functions
PE-FE
Transport Functions
Other NGNs

Figure 3. ITU-T RACF architecture

The Policy Decision Functional Entity (PD-FE) takes the final decision over the resource and admission control and delivers it to the corresponding Policy Enforcement Functional Entity (PE-FE) through the Rw interface. ITU-T is currently studying three alternatives for this interface, as specified in the Q.3323.x sub-series [9]: COPS-PR, H.248 and DIAMETER. The three of them are revisions of the Q.3303.x series and are still under discussion (targeted for December 2009).

The Transport Resource Control Functional Entities (TRC-FE) deals with the control of the resources which depend on transport technology. These entities are responsible for preserving and maintaining the network topology and resource database (NTRD) of each subdomain. The Rc interface is used to check the network topology and the status of network resources. The TRC-FE assigns resources to each QoS requesting flow. ITU-T has approved two alternatives for this interface, which have been specified in the Q.3324.x sub-series [10]: COPS-PR and SNMP. These specifications are revisions of the Q.3304.x series and are targeted for December 2009.

The scope and functions of the Rn and the Rh interfaces are also still under study, though it does clarify that one of the functions of the TRC-FE entity is to assign the network resources for its application and is not any of the Rc interface functions. Although ITU-T has not made any formal proposals some of the protocols which meet the requirements include SNMP and COPS-PR [11].

In this paper we are only concerned with intra-domain QoS policies, given that routing optimization will only be applied to interior gateway protocols (IGPs) and that neither the service stratum nor the user profile are required. The rest of the interfaces defined by [1] are therefore out of the scope of this paper.

Rw, Rc and Rn interfaces provide centralized management of QoS resources inside a domain and the protocol and procedures used for this task may be also be used for routing optimization. The most appropriate protocol to be used must then be determined.

### B. Intra-domain control of QoS resources

Two possible scenarios are defined by the RACF entity for the QoS resource control which are based on the type of sessions established by the user. These scenarios may be in push mode or in pull mode depending on the different QoS signalling capabilities of the costumer premises equipment (CPE) that initiate the sessions and the access technology (Fig. 4).

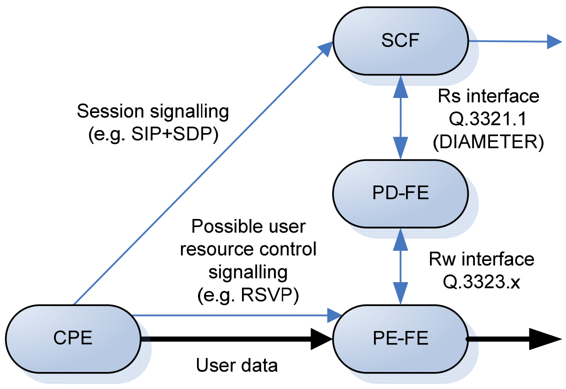


Figure 4. Push or pull operation depending on the type of user terminal

When user QoS control signalling does not exist a push mode scenario is initiated through the session signalling. The SCF are responsible for deriving the QoS needs of the requested service and sending the request to the RACF (PD-FE) for QoS authorization and reservation, sending the QoS policy to the network transport equipment (PE-FE). This mode is employed by CPE without QoS negotiating capacity (type-1 CPE) or by those with only service stratum negotiating capacity (type-2 CPE).

When the transport functions require signalling to perform a flow (e.g. RSVP o NSIS) a pull mode is initiated. In this mode it is the PE-FE which sends a QoS resource request to the PD-FE through the Rw interface, so that the RACF may take the appropriate authorization decision and replay with the final policy decision to be applied. This mode is used by CPE that explicitly request the QoS resource request through path-coupled QoS signalling (type-3 CPE).

As discussed in [12] the two protocols proposed by the ITU-T to manage the Rw interface in IP networks, DIAMETER and COPS-PR, have similar problems due to the natural client-server role (pull mode). The COPS-PR protocol works efficiently in pull mode but not so well in push mode. On the other hand, the original definition of the DIAMETER protocol also works correctly in pull mode but in this case the push mode is not contemplated. Moreover, even though most of the protocols defined in RACF interfaces and routing elements of the transport layer are still under definition, the common protocol for all of them will probably be COPS-PR. An additional factor to consider is that the COPS-PR which manages policy delivery on a native level is already supported by most IP-IP gateways (routers) in present-day networks.

### C. The COPS-PR protocol

The COPS-PR protocol [11], the provisioning model, is the COPS protocol variation created to deliver QoS policies between the PD-FE and the PE-FE for the DiffServ model. Within this model, the PD-FE proactively sends the network policies to be applied to the PE-FE.

In this model the PD-FE may proactively provision the PE-FE and both have a virtual container called PIB (Policy Information Base) where the policies are stored. This PIB has a tree structure formed by PRovisioning Classes (PRCs) which contain PRovisioning Instances (PRIs) [13]. Once the PE-FE has been initiated, and whenever there are updates, the appropriate policies are sent out by the PD-FE (Fig. 5). This way the PD-FE keeps the two PIBS synchronized.

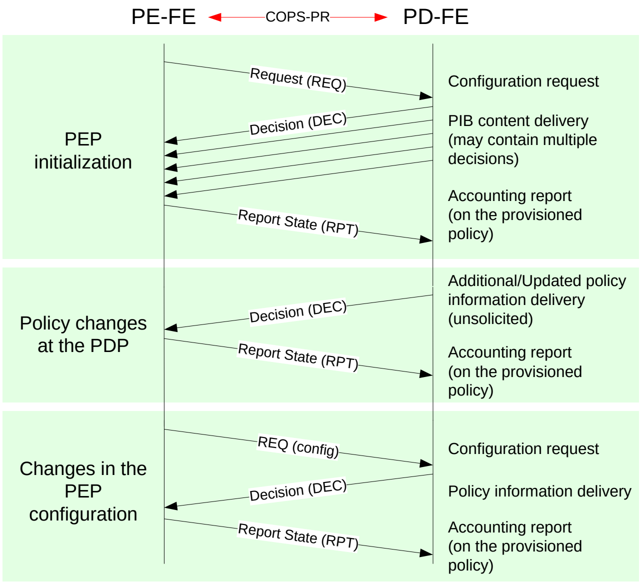


Figure 5. COPS-PR signaling between the PDP and the PEP

As stated before, when COPS-PR is applied to the RACF entity, the protocol works effectively in pull mode, but not so well in push mode, even though it does work correctly. In Fig. 6 we can see how this latter mode incorporates appended messages between the PD-FE and the PE-FE when the states of the different events which appear (detected by the PD-FE) are associated to the specific operations of the protocol.

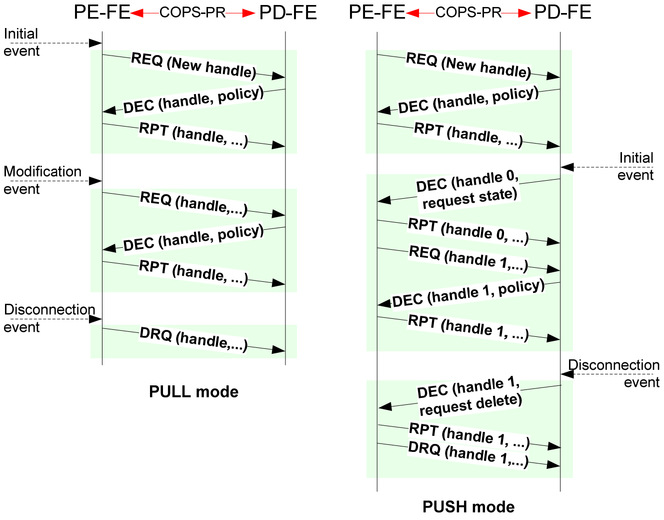


Figure 6. COPS-PR signalling in pull and push modes

In this paper, we propose the incorporation of the router link weights of the managed domain selected for routing optimization into the PIB policies with the aim of setting all the QoS parameters to the router through a single protocol, COPS-PR.

## III. Routing Optimization Fundamentals

In traffic engineering, a network optimal performance is generally accepted as being one where network congestion has been minimized in all the links so that all of them are equally congested. This is done from the existing resources utilization in a domain and the traffic demand matrices. One possible way of optimizing this performance is to manipulate the routing process of the packets, that is to say, by modifying the routing protocol.

The function of the routing protocols is to find the best route between an origin node and a destination node, from a minimization of the cost function. Given that the metrics provide a comparative measure to decide which path is better, we must make sure that the values of their components are properly adjusted so that they improve the global performance. Therefore to optimize routing we need to define an objective function which takes into account the link's usage in a quantifiable way and hence considers the routing cost.

The general routing problem can be described as the problem of optimizing the minimization of an objective function from a given network topology and a traffic demand matrix. Under stationary conditions the problem can be solved with linear programming (LP) and the result will be the best possible routing for all the possible flows so all the traffic will be globally optimized for all the networks in the domain.

Existing research lines tend to use the OSPF and ISIS protocols. The advantage of these protocols is that they incorporate IPv6 versions broadly used by commercials routers (RFC-5340 and RFC-5308). Moreover both protocols have versions oriented to traffic engineering with support for IPv6 (RFC-5329 and RFC-5305). These characteristics make them the most commonly-used protocols in NGN-related research. In this paper we have used OSPF, in concrete its version 3 (RFC-5340).

In order to fully understand how an effective approach to optimize overall network performance with OSPF and before considering the options available to achieve this optimization, we must first explain the how the OSPF protocol behaves.

### A. *Link state routing*

The OSPF protocol is one of the most common link state routing protocols in packet networks. This protocol works with a metric based in a cost value associated to each link and applies Dijkstra's shortest path algorithm [14] to find out the shortest path to each network based on these costs. In OSPF the metric of a path to a given network is the sum of all the costs to that network.

In the case where multiple paths exist to destination with equal metrics, OSPF can balance the load with equal cost multiple path (ECMP) so that the traffic flows will be theoretically evenly split between all the paths with the same metrics. Even though it is typically impossible to guarantee an exact even split of the load, it was decided to compare OSPF with ECMP and OSPF without ECMP using an exactly even traffic distribution to achieve the simulations of this paper.

Once the initial convergence phase has finished, any change which occurs in any link, for example a weight modification, will result in only the affected link's modification being flooded. Each one of the routers will have to decide whether or not to recalculate all the information in the routing table. Therefore if the number of changes of the link weights is high, it may lead to an inefficient use of the net's resources, as well as in bandwidth as in CPU.

In this paper the inverse of the link capacity has been used as a default configuration of the link weights (or costs). This methodology was first proposed by Cisco [15] and according to [3][16] is the best way to adjust the link weights in default configurations with OSPF.

### B. *The OSPF weight setting problem*

Given a known network topology and a predictable traffic demand, the OSPF weight setting problem (OSPFWS) is to find a set of weights which optimize the network performance and therefore minimize the cost function [3][17]. This problem, as stated previously, can be NP-hard depending on the dimensions of the topology. As this kind of problems can not be solved in a polynomial time, then heuristic search methods must be employed to find the most optimal solutions.

The use of local search heuristics, which apply an iterative process to solve the problem, only work for medium sized networks at the most and they do not guarantee the best possible solution [18]. The most commonly-used algorithm is the one proposed by Fortz and Thorup [16], which proposes minimizing a function that summarizes all the link weights so that it optimizes the global performance of the domain. This proposed cost function is convex, incremental, lineal, continuous and piece-wise, which assigns low costs to the infrequently used links and high costs to the overloaded links. If the problem needs to be solved for medium-big sized networks it is necessary to use artificial intelligence, given that using a local search heuristic is not viable in computing time terms.

## IV. Genetic Algorithm heuristics for routing optimization

Genetic algorithms [4][5] are methods for search, optimization and machine learning which are inspired by natural principles and biology. Differently from other optimization methods, GAs do not assume any structure or underlying distribution of the objective function and employ random, local operators to evolve a population of potential solutions. Since GAs have demonstrated to be able to solve complex problems that previously eluded solution [19], we have chosen to adopt this optimization model in our design. In the following, we first present the basic mechanics of GAs and then explain different GA implementation to solve the routing optimization problem.

### A. *Mechanics of genetic algorithms*

GAs evolve a *population* of individuals, where each of them represents a potential solution to the problem. Analogous to genetics, individuals are represented by *chromosomes*, which encode the decision variables of the optimization problem with a finite-length string. Each of the atomic parts of the chromosome is referred to as *genes*, and the values that the gene can take are addressed as *alleles*. To implement the principles of natural selection and competition among candidate solutions, GAs incorporate an *evaluation function* that gives a certain value of *fitness* to each individual, which indicates the quality of the given individual.

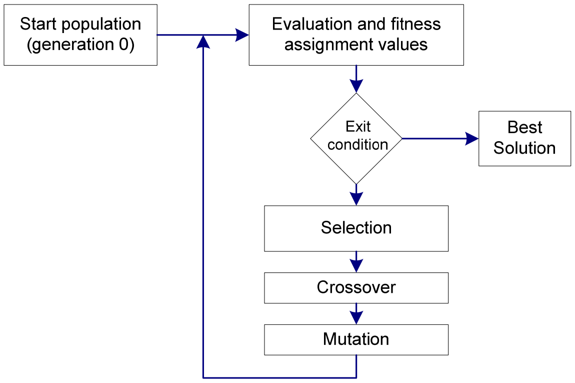

Start population (generation 0)
Evaluation and fitness assignment values
Exit condition
Best Solution
Selection
Crossover
Mutation

Figure 7. Flow chart of a Genetic Algorithm

Then, this population of individuals, which is usually initialized randomly, is evolved by a continuous process of *selection*, *crossover*, *mutation*, and *replacement* of individuals. That is, firstly the selection operator chooses the fittest individuals in the population, simulating the survival-of-the-fittest mechanism. Then, the crossover operator takes two or more of the selected individuals and recombines their genetic information in order to generate new, possibly better offspring. Afterwards, mutation introduces random errors on the transference of genetic information from parents to children, and finally, the offspring population replaces the original one. This process is repeated until a stop criterion is met; usually, the process is run during a prefixed number of iterations. Fig. 7 schematically illustrates this process.

The synergy of all these operators pressures toward the evolution and selection of the best solutions, which are recombined yielding new promising offspring. In [19], Goldberg emphasized the idea that, while selection, crossover, and mutation can be shown to be ineffective when applied individually, they might produce a useful result when working together. This was explained with the fundamental intuition of GAs, which supports the following two hypotheses. The first hypothesis is that the combination of the selection and crossover operators introduces a process of innovation or cross-fertilizing by generating new solutions from the fittest individuals in the population. As a consequence, new individuals are expected to be different from and better adapted than their parents. The second hypothesis is that the combination of selection and mutation represents a process of continuous improvement or local search. Thence, this process searches around the best solutions in the population with the aim of finding better solutions that are close to the parents.

### B. *Design of a genetic algorithm for routing optimization*

With the basic mechanisms of GAs in mind, now we are in position to proceed with the description of how GAs have been applied to the routing optimization problem. In what follows, we present the typical representation employed by several authors, and discuss which types of genetic operators have been used in different approaches [20][21][22][23][24].

In order to solve the OSPFWS problem, the solution must contain the weights of each link of the network. Therefore, the individual is typically represented as a vector that contains all the link weights of the domain, which range in [1, $w_{max}$]. Thus, each individual provides a complete solution to the problem. This population is typically initialized with the default weight setting with the inverse capacity procedure as proposed by Cisco [15]. On the other hand, several evaluation functions that provide a measure of the link performance of the domain, showing the most overloaded or the global average performance, have been used by different authors. In the following section three of these functions are explained in more detail.

Different genetic operators have been used in different GA implementations for this problem so far. Two selection schemes were employed in the approaches studied in this paper: rank selection and proportionate or roulette wheel selection. Rank selection ranks the individuals of the population according to their fitness, and those with better ranking are selected to be in the next generation. An especial case of rank selection is tournament selection, which uses a set of *s* randomly selected individuals for ranking instead of considering the whole population. On the other hand, proportionate selection gives each individual a selection probability that is proportional to its fitness with respect to the fitness of the other individuals in the population. Optionally in this phase a technique named elitism can be used, which consists in always passing at least one copy of the best chromosome to the next population, so the best individuals are not lost due to the effect of the genetic operators.

Then, crossover and mutation reproduce the parent population as follows. Crossover is applied to each pair of parents with a certain probability. If applied, crossover randomly generates a cut point and uses this cut point to shuffle the genetic information of the parents. Therefore, crossover creates two new individuals that mix the genetic information of the parents. If crossover is not applied, the offspring are exact copies of the parent. Thereafter, the offspring undergo mutation. That is to say, for each gene, a random number is generated and, if it is lower than the probability of applying mutation, the gene is mutated by assigning a new randomly selected value in the interval [1, $w_{max}$]. After mutation, the new population replaces the original one.

A representative example of a GA applied to routing optimization is provided by Ericsson et al. [20]. In this case the GA is based on the idea proposed by the heuristic search in [3][17] and it applies the same cost function. The representation of the population's individuals is formed by the set of all the link weights in a vector. The population initialization is randomly generated and the selection method is the rank selection where it divides the population in three sets of $\alpha$=20% (elitism), $\beta$=70% (crossover) and $\gamma$=10% (discarding), in respect the total population size between 50 and 500 individuals. The crossover probability is 70%, mutation probability is 1% and the number of iterations is variable between 500 and 700.

Throughout this section, we have explained the process organization of GAs, have intuitively discussed how and why they work, and have shown how GAs have been applied to the routing optimization problem. In the next section, we take these ideas and explain how GAs can be enhanced by incorporating a new local search procedure that enhances the original local search mechanism of GAs – that is, mutation – in order to converge quicker to the objective.

## V. Local search with GA for routing optimization

The hybrid genetic algorithms (HGA) [21], or memetic algorithms, are distinguished from the GA because they append a local search heuristic applied during the evolutionary cycle, as can be seen in a typical HGA flowchart in Fig. 8. The objective of this local search procedure is to improve the effectiveness and efficiency of a GA when converging to an optimal solution of the problem. In this section we provide an inedited theoretical comparative analysis of the main three HGA proposals.

The hybrid genetic algorithms obtain better results when optimizing the global performance of the domain with OSPF routing process rather than the simple GA [22][23][24]. Some benefits that the local search addition provides are acceleration in the optimization process (in computational time) and improvement in the quality of the solutions, which are more optimized, that is to say better, as demonstrated in [23] and [24]. However, a disadvantage is the potential loss of the global maximum, getting stuck in a local maximum, as happens when there is an abuse of the genetic operators.

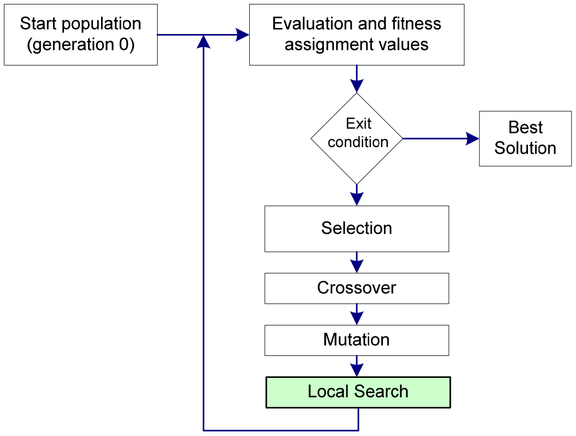

Start population (generation 0)
Evaluation and fitness assignment values
Exit condition
Best Solution
Selection
Crossover
Mutation
Local Search

Figure 8. Flow chart of a Hybrid Genetic Algorithm

A theoretical analysis of the HGA algorithms proposed in [22], [23] and [24] has been carried out. Table I presents a summary of the most representative genetic parameters of the three proposals. All of them have used the same values in the crossover parameter and mutation, but they differ in the selection method, weight representation and overcoat, where to apply the local search procedure and the fitness function.

TABLE I. Parameters of the Hybrid Genetic Algorithms

| | **Mulyana & Killat [22]** | **Buriol et al. [23]** | **Riedl & Schupke [24]** |
|---|---|---|---|
| **Population Size** | 50 | 50 | 20 |
| **Chromosome Representation** | Set of all domain's link weights in a vector | | |
| **Weight Representation** | [1, 99] | [1, 20] | [1, 20] |
| **Number of Iterations** | 200 | 200 | 200 |
| **Selection Method** | Rank Selection: $\alpha$=20%, $\beta$=70%, $\gamma$=10% | Rank Selection: $\alpha$=25%, $\beta$=70%, $\gamma$=5% | Roulette Wheel Selection |
| **Crossover Probability ($P_c$)** | 0.7 | 0.7 | 0.7 |
| **Mutation Probability ($P_m$)** | 0.01 | 0.01 | 0.01 |
| **Heuristic Search** | Best individual | Individual generated from the crossover | All individual |

The fitness function used by Buriol et al. in [23] is the same as the one used by Resende et al. in [20], in the first application of a GA to the optimization problem and has been introduced in the previous section. Both of them use the convex cost function proposed by Fortz and Thorup [16].

$$\min \phi = \sum_{a \in A} \phi_a \tag{1}$$

subject to

$$\sum_{u:(u,v)\in A} f^{(st)}_{(u,v)} - \sum_{u:(u,v)\in A} f^{(st)}_{(v,u)} =$$

$$= \begin{cases} -d_{st} & \text{if } v = s, \\ d_{st} & \text{if } v = t, \quad v, s, t \in N, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

$$\ell_a = \sum_{(s,t)\in NxN} f^{(st)}_a, a \in A, \tag{3}$$

$$\phi_a \geq \ell_a, a \in A, \tag{4}$$

$$\phi_a \geq 3\ell_a - 2/3c_a, a \in A \tag{5}$$

$$\phi_a \geq 10\ell_a - 16/3c_a, a \in A \tag{6}$$

$$\phi_a \geq 70\ell_a - 178/3c_a, a \in A \tag{7}$$

$$\phi_a \geq 500\ell_a - 1468/3c_a, a \in A \tag{8}$$

$$\phi_a \geq 5000\ell_a - 19468/3c_a, a \in A \tag{9}$$

$$f^{(st)}_a \geq 0, a \in A; s, t \in N \tag{10}$$

Later, Mulyana and Killat [22] tackled the General Routing Problem by minimizing the following fitness function, which takes into account a weighted addition of the global average and the maximum link utilization.

$$min \left[ (a_t \cdot t) + \frac{1}{|E|} \sum_{ij} \sum_{uv} \frac{\ell^{uv}_{ij}}{c_{ij}} \right]$$

$$\forall (i,j) \in E, \forall (u,v) \in V \times V \tag{11}$$

$$\delta_{un} f_{uv} + \sum_{m \in V} \ell^{uv}_{mn} = \delta_{nv} f_{uv} + \sum_{m \in V} \ell^{uv}_{nm}$$

$$\forall (u,v) \in V \times V, \forall n, m \in V \tag{12}$$

$$\sum_{uv} \frac{\ell^{uv}_{ij}}{c_{ij}} \leq t, \forall (i,j) \in E$$

$$\ell^{uv}_{ij} \geq 0, \forall (i,j) \in E, \forall (u,v) \in V \times V \tag{13}$$

Finally, the fitness function employed by Riedl and Schupke in [24], only considers minimizing link maximum utilization with the application of an exponential scalability factor. This way, routing solutions with smaller maximum link utilization receive higher fitness values and a greater chance to be reproduced in the new generation.

$$fitness = \left( \frac{1}{\rho_{max}} \right)^p, p > 0. \tag{14}$$

subject to:

$$\rho_{max} \geq \rho_{ij} \quad \forall (i,j) \in \varepsilon \tag{15}$$

$$\rho_{ij} = \sum_{u \in V} \frac{f_{ij,u}}{C_{ij}} \quad \forall (i,j) \in \varepsilon \tag{16}$$

## VI. Evaluation of the HGAs

The aim of this section is to choose one of the HGAs proposed in the previous section and consequently implement it in a real testbed, providing a comparative analysis of the main three HGA proposals. The main problem when comparing the different algorithm's proposals of the authors is the fact that each one uses its own network topologies and traffic matrices. As the objective is to evaluate comparing the algorithms it is necessary to determine a common topology and traffic demand matrix to apply and test them. When this comparative was first proposed by the authors in [7] there was a physical limitation of twelve routers creating a restriction regarding the topology of the test. Therefore the network topology N11 used in [24] has been selected. This network topology, which has 11 nodes and 48 unidirectional links, can be seen in Fig. 9.

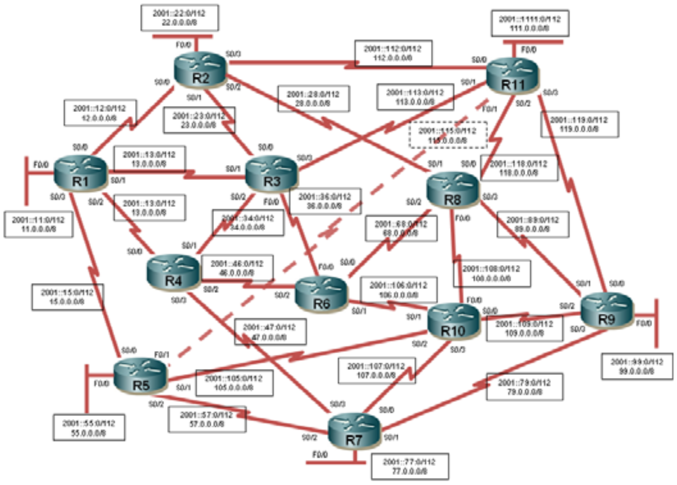


Figure 9. Network topology N11 used to evaluate the routing optimization

In order to evaluate the proposals comparisons between the different approaches have been made: the default inverse capacity metric (InvCap) [15], the local search algorithm proposed by [17], the GA proposed by [20] and the three HGAs proposed by [22], [23] and [24]. The parameters proposed by the authors of the HGA algorithms have been used in this evaluation. Algorithms with a maximum number of iterations of 250 and a maximum percentage of total weight change of 30% from initial configuration were used in order to avoid excessive flooding which would have consequently led to excessive routing re-calculations. Six traffic demand matrices with increasing traffic have been generated.

The graphs show how, in general, the HGA algorithms improve local search and simple GA algorithms' results. The two graphs, OSPF without ECMP (Fig.10) and OSPF with ECMP (Fig.11), show how the HGA algorithm proposed by Buriol et al. [23] is the one that most effectively minimizes the maximum usage of the most congested link. Overcoat in the worst traffic cases when under default conditions (InvCap) there is link overloaded, even though there exist minimum differences with respect to the other two HGA proposals.

On the other hand the HGA algorithm proposed by Mulyana and Killat [22] is the one which best minimizes, albeit marginally, the average usage of the links in all cases, as can be seen in the graphs of OSPF without ECMP (Fig.12) and OSPF with ECMP (Fig. 13). In the case of average usage of the links, even though the HGA algorithms always provide better results, the deviation with the others are minimal in percentage

According to results minimal variations between the three HGA proposals exist, both in the average usage as well as in the maximum usage. It was decided to use the algorithm proposed by Buriol et al. [23] because it presents slightly better results in this latter aspect.

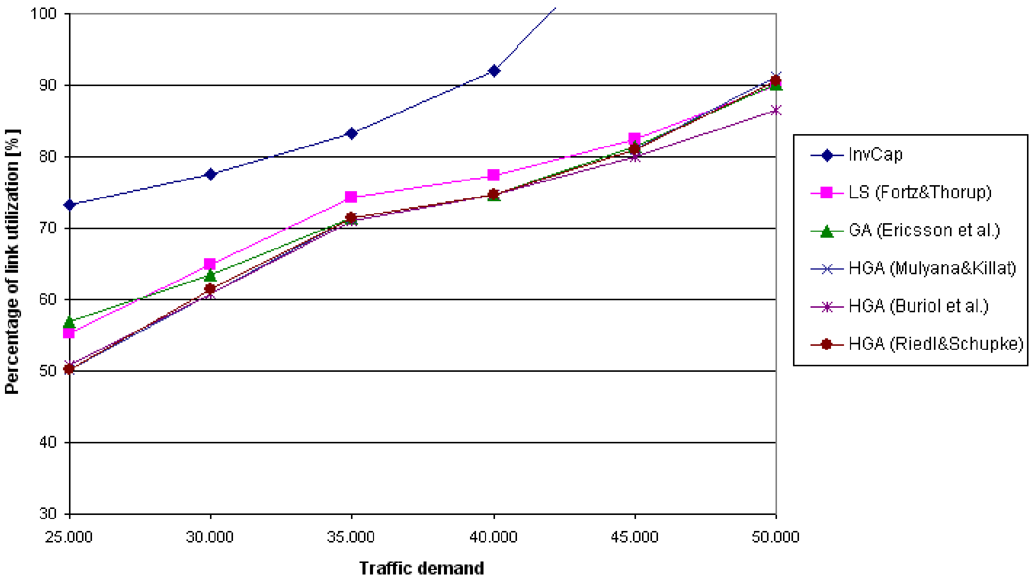


Figure 10. Maximum link utilization without ECMP

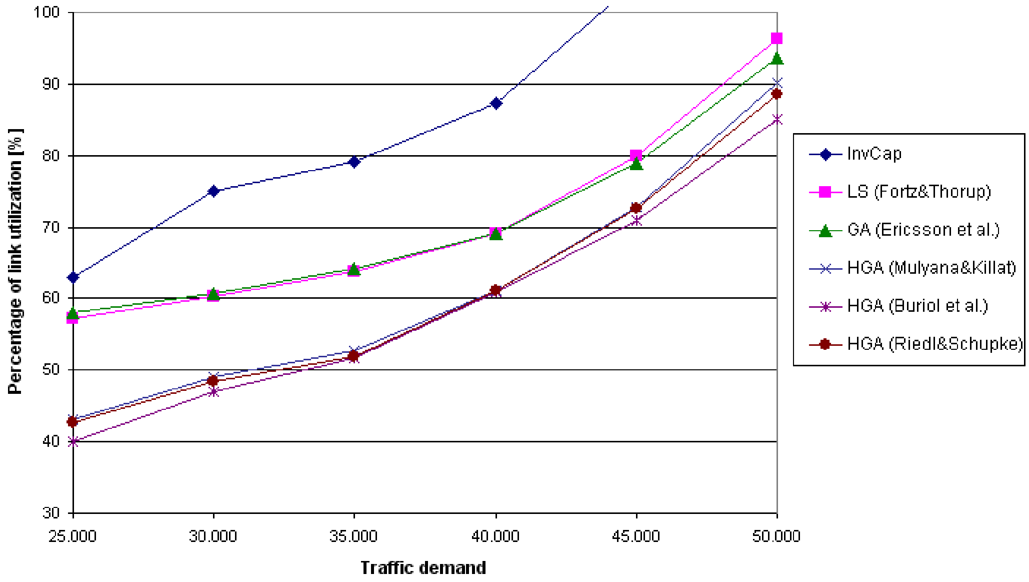


Figure 11. Maximum link utilization with ECMP

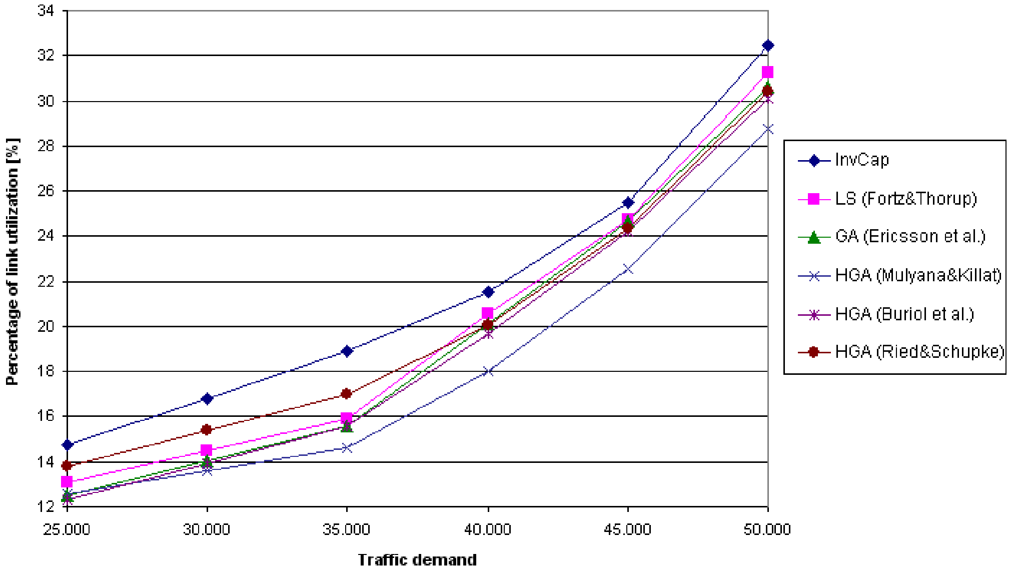


Figure 12. Average link utilization without ECMP

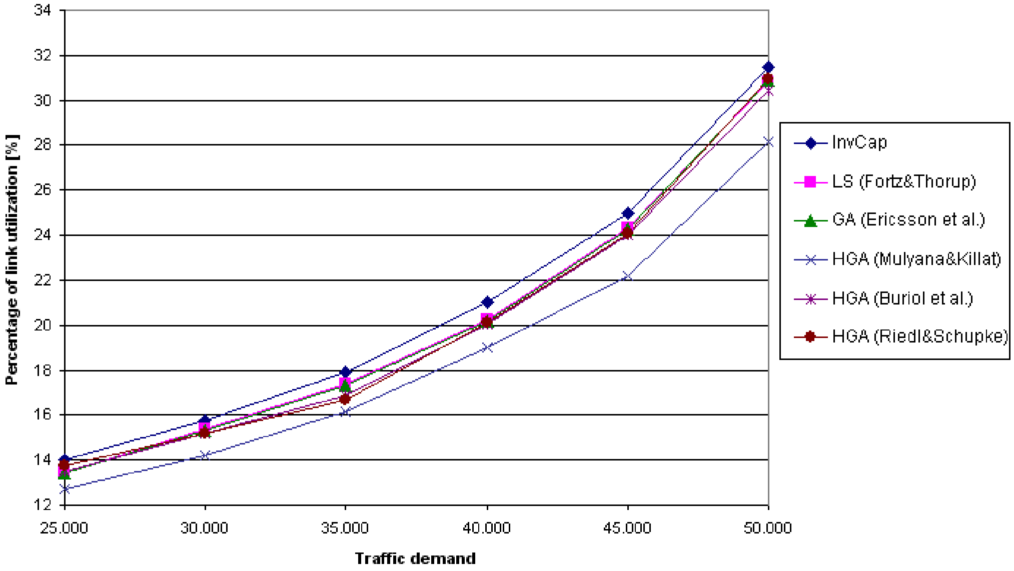


Figure 13. Average link utilization with ECMP

## VII. TESTBED IMPLEMENTATION

This section presents a real testbed implementation where it has been upgraded the routing optimization presented in [7] to meet the ITU-T requirements. Thus, in order to deliver the link weights to the routers over a NGN testbed we have used the COPS-PR protocol

The authors presented in [7] a successful implementation of routing optimization in an IPv6 domain with eleven commercial 262x Cisco routers and with a centralized GNU/Linux device. The task of this centralized device was computing the link weights through a HGA algorithm and sending them to domain routers, where they were configured automatically (Fig. 1 shows the architecture). In the aforementioned paper the sending of the optimized link weights was carried out via the SSH and the routing protocol used was OSPFv3.

In this paper we have integrated the application developed in [7] into the NGN testbed presented in [12], where we implemented an end-to-end QoS management signaling proposal for the ITU-T NGN architecture. In this latter paper the COPS-PR protocol was used to manage the intra-domain policies and resources with the GNU/Linux routers (Rw, Rc and Rn interfaces) and the COPS-SLS protocol was used to dynamically negotiate the inter-domain policies among the centralized devices of each domain (Ri interface). These devices are formally the RACF entity and have been named as QoS Brokers (QoSBv6).

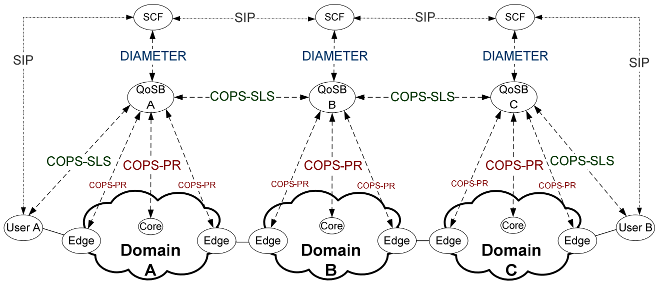


Figure 14. Proposed architecture in [12] for end-to-end signalling

We have used an application in the QoSBv6 to compute the offline routing from the domain's topology and the traffic matrix with the OSPFv3 protocol. This application was developed in [7]. The default link weights of the GNU/Linux router interfaces are obtained by using the link's inverse capacity [15] and applying the HGA algorithm proposed by Buriol et al. [23] these link weights can be modified, to obtain the new "genetically" optimized weights. The sending of these new OSPF weights to the GNU/Linux router interfaces is carried out via the COPS-PR protocol.

This way, whenever the administrator decides to optimize routing, which is relatively infrequent, the application applies the HGA to the current topology and sends the new weights to the system Database, which will be copied into the COPS-PR PIB and, later, delivered to the routers with COPS-PR.

### *A. System Database*

The database stores the domain policies such as service level agreements (SLA) with the client traffic demand matrix of the client, the network topology (nodes, link's capacities and costs) and the addresses of the QoSBv6s of adjacent domains. The client traffic demand matrix is a maximum traffic configuration, required in order to assure client's contract traffic.

PostgreSQL v.7.4.6. is used because its JAVA connector supports IPv6 connections. The SLS policies and the link weights are transferred to the PIB for their distribution through COPS-PR to the routers or Policy Enforcement Physical Entity (PE-PE). Therefore, the relation of routers of the domain permits the installation of the network policies to the routers.

### *B. COPS-PR module and the PIB*

The protocol used for the intra-domain communication is COPS-PR. Besides the support of IPv6 protocol, we implement the whole protocol, including the keep-alive function, the synchronization function for the PDP-PEP disconnection case and the PEP-redirect function in the case of failure. This latter function also supports PDP redirection with IPv6 addresses. Support for the PIB defined by the DiffServ WG [25] has also been included in addition to the previously determined RAP WG PIB definition [26], which always supports IPv6. Our PIB completely fulfils them.

We have modified the COPS-PR Policy Information Base (PIB) so the router link weights are sent as a piece of the policies. COPS-PR allows unsolicited sending of policies from the PD-FE to the PE-FE and, moreover, it only allows sending parts of the PIB for efficiency improvements. Therefore whenever it is necessary to change the link weights, only a small part of the PIB is sent through COPS-PR, which will be only the modified links weights.

### *C. The GNU/Linux router*

The IPv6 protocol is currently supported by the main software and hardware components and is present in the main worldwide networks. Furthermore, as a result of the work carried out by the USAGI project [27], which has merged its work into the official Linux kernel, IPv6 stack in Linux OS is now fully compliant with advanced IPv6 conformance and interoperability tests. Therefore NGN testbeds using the IPv6 protocol with Linux OS can be implemented with performance guarantees.

The PE-PE module installed in every router fully supports COPS-PR over IPv6 and is responsible for the configuration of the policies in the PE-PE's PIB and the link weights in a computer running GNU/Linux as a router. The GNU/Linux kernel has been configured for IPv6 support with all the QoS functionalities available to be used in IPv4 as well as IPv6.

The load balancing in the routers is automatically activated if the routing table has multiple paths to a destination. In this testbed per-destination load balancing has been used, where the router distributes the packets based on the destination address. Another option could have been per-packet load balancing which guarantees equal load across all links but there is the possibility that the packets may arrive out of order at the destination if differential delay exists within the network and it is a processor intensive task which may impact the overall forwarding performance. We should underline the fact that even though per-destination load balancing is an improvement over per-packet, it is still not very good because if substantially more packets are sent to one destination than to another, the overall bandwidth utilization will be uneven.

### *D. From theory to real world*

Thus far, we have proposed a traffic engineering method to be applied under ITU-T NGN specifications and we have proven its viability by using COPS-PR for the router link weights delivery. We have successfully optimized weights of a single GNU/Linux router applying the offline algorithm successfully proven in simulations and using the NGN testbed successfully deployed in [12]. Despite this headway progress, in this paper we have not optimized and evaluated a complete system with this new proposal even though we are currently working on it.

Nevertheless it must be remarked the great difficulty of the hand-on testbed versus simulation implementations. Moreover the fact of haven built our testbed over physical machines, it provides an extra difficulty over the emulated solution. The preliminary results show the viability of our proposal in a real environment and using open systems. Therefore, this paper represents the baseline for future work detailed in the next section.

## VIII. Conclusions and further work

The ITU-T MS/NGN architecture must provide support for the QoS of the multimedia sessions. This support includes the QoS negotiation, admission and resource control for different end-to-end QoS models. Therefore, it seems reasonable to integrate traffic engineering for intra-domain optimization into this architecture. Assuming this hypothesis, this paper has presented the implementation of a traffic engineering proposal in ITU-T NGN environment by means of routing optimization. This optimization has been done through the application a HGA algorithm to offline routing with the aim of modifying the OSPF link weights and therefore minimizing the maximum utilization of the domain's links.

A comparative study of the various proposals to solve the OSPFWS problem has been carried out and it has been demonstrated that HGA algorithms provide good solutions to the complex problem, better than those provided by GA algorithms. Among those, the one which provided slightly better results was the Buriol et al. proposal and which was hence selected to be implemented in the testbed.

The testbed implementation has been done with GNU/Linux routers and with a centralized QoSBv6 device. This device had the task of computing the link weights through the elected HGA algorithm and sending them to the domain's routers, where they are configured automatically. OSPFv3 is the routing protocol which decides the routing inside the autonomous system implemented with IPv6 and computed with the optimized weights. The sending of the optimized link weights has been carried out with the COPS-PR protocol, proposed by the IMS/NGN architecture to manage domain's internal policies with PBNM.

We are currently working on extending the work presented in this paper to optimize and evaluate a complete system with this new proposal. On the other hand, given that one of the functions of the QoSBv6 in [12] is to manage QoS DiffServ model policies inside a domain, in [28] a description of how routing optimization can be applied to the DiffServ model is provided. This testing will be also undertaken in further work.

## Acknowledgment

We would like to thank "Enginyeria i Arquitectura La Salle" (University Ramon Llull) for their encouragement and assistance, especially C. Duz and D. Cutiller for his support throughout the development of this paper.

## References

[1] ITU-T Rec. Y.2111, "Resource and Admission Control Functions in NGN (version 2)". November 2008.

[2] X. Xiao, L. Ni, "Internet QoS: A Big Picture", IEEE Network, pp. 8-18, March 1999.

[3] B. Fortz, M. Thorup, "Internet traffic engineering by optimizing OSPF weights", Proceedings of IEEE INFOCOM 2000, pp. 519-528, March 2000.

[4] J. H. Holland,"Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artiçial Intelligence", MIT Press/ Bradford Books edition, 1992.

[5] D.E. Goldberg, "Genetic Algorithms in Search, Optimization & Machine Learning", Addison-Wesley, Massachusetts, 1989.

[6] D. Durham, Ed., J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, "The COPS (Common Open Policy Service) Protocol", IETF RFC 2748, January 2000.

[7] A. Vallejo, et al., "Implementation of Traffic Engineering in NGNs Using Hybrid Genetic Algorithms", Proceedings of ICSNC 2008, pp.262-267, October 2008.

[8] ITU-T Rec. Y.2021 "IP Multimedia Subsystem for NGN". September 2006.

[9] ITU-T Draft Recommendations Q.3323.x: "Protocol at the interface between Policy Decision Physical Entity (PD-PE) and Policy Enforcement Physical Entity PE-PE (Rw interface) version 2", 2009.

[10] ITU-T Draft Recommendations Q.3324.x: "Protocol at the interface between Transport Resource Control Physical Entity (TRC-PE) and Transport Phisical Entity (T-PE) (Rc interface) version 2", 2009.

[11] K. Chan, et al., "COPS Usage for Policy Provisioning (COPS-PR)", IETF RFC 3084, March 2001.

[12] A. Vallejo, A. Zaballos, X. Canaleta and J. Dalmau, "End-to-end QoS management proposal for the ITU-T IMS/NGN architecture", Procedings of SoftCOM 2008, pp 147-151, September 2008.

[13] K. McCloghrie, et al., "Structure of Policy Provisioning Information (SPPI)", IETF RFC 3159, August 2001.

[14] E. Dijkstra, "A note on two problems in connection of graphs", Numerical mathematics, Vol. 1, pp. 269-271, 1959.

[15] Cisco Systems, "Configuring OSPF", Cisco Systems, Inc., San Jose, USA, August, 2006.

[16] B. Fortz, J. Rexford, M. Thorup, "Traffic Engineering with Traditional IP Routing Protocols", IEEE Communications Magazine, Vol. 40, N. 10, pp. 118-124, May 2002.

[17] B. Fortz, M. Thorup, "Increasing Internet capacity using local search", Computacional optimization and applications, Vol. 29, N. 1, pp. 13-48, October 2004.

[18] M. Söderqvist, "Search Heuristics for Load Balancing in IP Networks", SICS. Technical Report T2005:04, March 2005.

[19] D.E. Goldberg, "The Design of Innovation: Lessons from and for Competent Genetic Algorithms", Kluwer Academic Publishers, 2002.

[20] M. Ericsson, M.G.C. Resende, P.M. Pardalos, "A Genetic Algorithm For The Weight Setting Problem in OSPF Routing", Journal of Combinatorial Optimization, Vol. 6, N. 3, pp. 299-333 , September 2002.

[21] P. Moscato, "On Evolution, Search, Optimization, Genetic Algorithms and Material Arts: Towards Memetic Algorithms, Caltech Concurrent Computation Program, C3P Report 826, 1989.

[22] E. Mulyana, U. Killat, "A Hybrid Genetic Algorithm Approach for OSPF Weight Setting Problem", Proceedings of the 2nd Polish-German Teletraffic Symposium PGTS 2002, September 2002.

[23] L.S. Buriol, M.G.C. Resende, C.C. Ribeiro, M. Thorup, "A hybrid genetic algorithm for the weight setting problem in OSPF/ISiS routing", Networks, Vol. 46, N. 1, pp. 36-56, 2005.

[24] A. Riedl, D. A. Schupke, "Routing optimization in IP networks utilizing additive and concave metrics", IEEE/ACM Transactions on Networking, Vol. 15, N. 5, pp. 1136-1148, 2007.

[25] K. Chan, R. Sahita, S. Hahn, K. McCloghrie, "Differentiated Services Quality of Service Policy Information Base", IETF RFC 3317, March 2003.

[26] R. Sahita, Ed., S. Hahn, K. Chan, K. McCloghrie, "Framework Policy Information Base", IETF RFC 3318, March 2003.

[27] University of Tokyo, USAGI (UniverSAl playGround for Ipv6) Project, December 2008, [Online]. Available: http://www.linux-ipv6.org.

[28] B. Fortz, M. Thorup, "Optimizing OSPF/ISIS weights in a changing world", IEEE Journal on Selected Areas in Communications, Vol. 20, N. 4, pp. 756-767, May 2002.

# Interactive Internet Television for Mobile Devices and Large-scale Areas

Radim Burget, Dan Komosny, Jakub Müller

***Abstract*—In coming years it is supposed that importance of the Internet television technology will grow and also number of households and subscribers paying for this service will be increasing. It is also supposed that this technology will enable a new kind of services, where one of them is an interaction of customers with content provider. For this purpose a method of hierarchical aggregation for a feedback transmission has been proposed, which is in comparison to classical real-time control protocol quite scalable. This paper describes integration of hierarchical aggregation with internet coordinate systems, which can make communication between session members more efficient. It also describes some advantages of this integration and a prototype of such system is introduced. Furthermore, it describes some use examples and options for extensions of such architecture.**

***Index Terms*—Quality of Service, Global network positioning, Real-time control protocol, Real-time protocol, Hierarchical aggregation**

## I. Introduction

The Internet Protocol TeleVision (IPTV) market has achieved a great attention in recent years. According to several independent market analyses, IPTV technology will soon take a significant additional market share among China, Europe, USA and even other areas. IPTV will enable a new kind of services such as interactive TV. However, to be really interactive, there is need to transmit all the user action in some limited time period and this could become a problem. Consider an example where we have IPTV session with two million of subscribers. When all the subscribers decide to vote for some kind of poll, this would lead for ten millions times sending a message of at least 64 bytes (assuming a presence of User Datagram protocol (UDP) header, a packet header to distinguish it from standard Real-time Transport Protocol / Real-time Control Protocol (RTP/RTCP) header and, also, there can be not even simple YES/NO votes), this would lead to need to transmit about 600 MB. A general view of the term *interactive TV* is a communication between content provider and subscribers, where the content provider announces some poll and waits for some time for answers from subscribers. We are in this work motivated not by a need to enable just simple request/reply model, where also simple HTTP protocol based approaches can work sufficiently. We are in this work motivated by a vision to enable conveying of subscriber actions continuously during entire TV program time, from . In such cases especially for a bigger numbers of receivers it will pay of to provide optimized and relatively well scalable backchannel technology. In the opposite case it may arise some traffic peaks, which potentially may cause loss of votes, harm other already running services on the network or even the IPTV broadcasting itself. One of the promising technologies for this purpose is so-called hierarchical aggregation (HA) [3], [4], [5], [13], [14].

This paper describes how the hierarchical aggregation (HA) can be integrated with coordinate systems to save bandwidth and proposes a real architecture for IPTV systems. HA is inspired by principle often utilized by wireless sensor networks (WSN) and it is used to gather huge amount of data in a short amount of time. However in WSN, there is emphasis on energy efficiency. In the field of the Internet there less need to take emphasis on the energy consumption. However compared with WSN, there is need to take a better emphasis on complex Internet topology.

The integration of internet coordinating systems with HA can make communication even more effective and it can enable not only simple interaction as a question/response in some kind of polling, but even continuous connection of subscribers with content provider and convey their opinion during entire time of the session. It is also scalable enough for any further growth of number of receivers in the session and even mobile devices in future.

What should be emphasized here, this paper does not deal with security issues. The securing the communication can be often simplified by the identification of paying subscribers and can significantly vary from case to case.

The first part of this paper is involved in RTP/RTCP protocols defined in RFC 3550 [17] specification and its mathematical foundation. It also gives a brief overview of the HA. The next section describes internet coordinate systems and gives a brief comparison of these algorithms concerning HA. The following section proposes an architecture for integration of HA with selected coordinate system. The last section describes how it could be further extended to estimate positions of subscribers.

## II. Introduction to Real-time Protocol and Hierarchical Aggregation

RTP and RTCP [1] are protocols designed for data delivery in real-time and, among other things, to measure the quality of service (QoS). This couple of protocols is today used for almost all transmission of time sensitive data such as audio, video, subtitles, etc. This is the reason why the paper will describe RTCP protocol rather than IPTV service.

The RTCP protocol uses receiver reports (RR-RTCP) and sender reports (SR-RTCP), which are sent between sender and receiver and which contain necessary information to evaluate e.g. RTP packet loss, jitter, round trip delay time, etc. In the RTCP protocol, the maximal consumed bandwidth is limited to 5 % of the total session reserved bandwidth. To meet this limitation, the period for transmitting RTCP messages must exactly fulfill the following equations [17]. These equations compute the period for transmitting RR-RTCP messages ($T'_{\mathrm{RR}}$), SR-RTCP messages ($T'_{\mathrm{SR}}$) and RSI-RTCP messages ($T_{\mathrm{RSI}}$). All of them are described by equations (1), (2), (3) and (6). When the number of users is low the period will be evaluated as too short and this will lead to unnecessarily wastage of bandwidth. For this reason the final values of equations $T_{\mathrm{RR}}$, and $T_{\mathrm{SR}}$ are limited by their lower bounds by the constant of 5 seconds. Finally the compensation factor $\mathrm{C}$ is added; see equations (6), (5), (7) (or see [15], [17] for more detailed information) to take also into consideration empirical experiences and long-term observations.

$$T'_{\mathrm{RR}} = \frac{L_{\mathrm{RR}} \cdot n}{75\ \% \cdot B_{\mathrm{RTCP}}} \qquad (1)$$

$$T'_{\mathrm{SR}} = \frac{L_{\mathrm{SR}}}{25\ \% \cdot B_{\mathrm{RTCP}}} \qquad (2)$$

$$B_{\mathrm{RTCP}} = B \cdot 5\ \% \qquad (3)$$

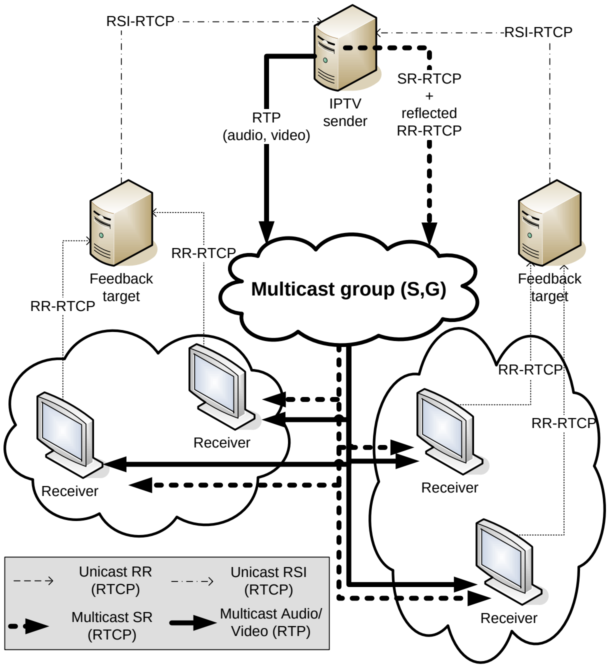


Fig.1. The RFC 3550 RTP/RTCP protocol improved with hierarchical aggregation. RR-RTCP stands for receiver report [17], SR-RTCP stands for sender report [17], RSI-RTCP stands for receiver summary information packet [15]

$$T_{\mathrm{RR}} = \frac{\max\ (T'_{\mathrm{RR}}; 5\ \mathrm{sec})}{\mathrm{C}} \qquad (4)$$

$$T_{\mathrm{SR}} = \frac{\max\ (T'_{\mathrm{SR}}; 5\ \mathrm{sec})}{\mathrm{C}} \qquad (5)$$

$$T_{\mathrm{RSI}} = 1.5 \cdot T_{\mathrm{SR}} \qquad (6)$$

$$\mathrm{C} = \mathrm{e} - 1.5 = 1.21828 \qquad (7)$$

*L* stands for the packet length of a message where its index denotes the packet type, *B* stands for the total session bandwidth, $B_{\mathrm{RTCP}}$ for the bandwidth reserved for RTCP protocol, and *n* is the total number of receivers in the whole session. As follows from equation (1) and (4), for a large number of receivers *n* the period $T_{\mathrm{RR}}$ can become pretty long and this leads to averaged values from longer time period, which can be useless for some kind of applications, especially an interactive ones. One way to cope with this problem is to break RTP/RTCP recommendations and use more than 5 % of session bandwidth. Another approach is to use method such as HA.

HA is one of the improvements for the RTCP protocol that has been recently introduced. Thanks to HA the idea of redundant data flow reduction has been advanced even further than any other RTCP improvement. It uses feedback targets and these feedback targets are organized hierarchically. With their help data redundancy can be removed at a short distance from the receiver and this gives us the ability to construct topologies ready for large-scale deployment where a huge number of receivers can be connected at the same time with low bandwidth consumption for RTCP protocol transmission. As described in detail in [3], HA can give even up to 100 times faster signaling gathering in comparison with the RFC 3550 RTCP standard, when in both cases 51 kbps bandwidth, and $10^5$ receivers is present in the session [18]. The value 51 kbps equals the IPTV streaming with 1 Mbps reserved bandwidth for the service (i.e. 5 % of the whole service, as defined in RFC 3550).

In HA three types of members exist: sender, feedback targets and receivers. The sender transmits multimedia data, sender reports (SR-RTCP) and receiver summary packets (RSI-RTCP) to a multicast channel. The receivers receive multimedia data from the multicast channel and transmit receiver reports (RR-RTCP) to a feedback target via a unicast channel. These receiver reports contain information about the quality of reception and they can be also extended by an additional content (e.g. vote). And finally, feedback targets receive receiver reports (RR-RTCP) and they create statistics about QoS of these reporting receivers. RSI-RTCP messages are then created from many of these reports and they are transmitted to sender or another feedback target, when multilevel hierarchical aggregation is used (see Fig. 1).

In HA the receivers and feedback targets have to be organized in a hierarchical tree structure and the sender has to be informed about the size of each subgroup below feedback target, in other words, about how many members share the $B_{\mathrm{RTCP}}$ bandwidth. It is necessary to know this to be able to calculate the period lengths $T_{\mathrm{RR}}$, and $T_{\mathrm{RSI}}$ as shown in equations (8), (9), (10) and (11):

$$T'_{\mathrm{SR}} = \frac{L_{\mathrm{SR}}}{25\ \% \cdot B_{\mathrm{RTCP}}}, \tag{8}$$

$$T'_{\mathrm{RSI\_S}} = \frac{L_{\mathrm{RSI}}}{75\ \% \cdot B_{\mathrm{RTCP}}}, \tag{9}$$

$$T'_{\mathrm{RSI\_FT}} = \frac{L_{\mathrm{RSI}} \cdot n_{\mathrm{FT}}}{75\ \% \cdot B_{\mathrm{RTCP}}}, \tag{10}$$

$$T'_{\mathrm{RR}} = \frac{L_{\mathrm{RR}} \cdot n_{\mathrm{G\_R}}}{75\ \% \cdot B_{\mathrm{RTCP}}}, \tag{11}$$

where $T'_{\mathrm{RSI\_S}}$ stands for the time interval for sending RSI-RTCP message transmission from the sender, $T'_{\mathrm{RSI\_FT}}$ stands for the time interval of the RSI-RTCP packet transmission from the feedback targets, $n_{\mathrm{FT}}$ and $n_{\mathrm{G\_R},}$ give the number of neighbouring feedback targets or receivers that have a common feedback target in a single subgroup (see Fig. 1). All the formulas have to be compensated by the compensation factor $C$ and its lower bound is limited to a constant of 5 seconds (4), (5) [17].

From these equations it is quite clear that especially for large scale sessions with huge number of receivers the traffic load over a network can be better spread and thanks to the aggregation also some degree of bandwidth reduction is possible.

## III. Coordinate systems

The Internet coordinate systems are quite new approach how to localize hosts even in fixed networks. The major motivation for utilizing these methods is to optimize communication in the network and reduce bandwidth used in the same session size. And this is also our motivation of utilizing them in the field of HA. In the next few paragraphs the coordinate system types will be assessed from the perspective of HA.

The coordinate system methods can be divided into two basic groups: central based which utilizes landmarks and distributed one which are commonly based on physical model of spring network, which produces tension between hosts. [19]

### A. *Centralized coordinate systems*

The Global Network Positioning (GNP) algorithm runs through two separate steps: first a set of landmarks is established and secondly the host position prediction is done. The landmarks are a subset of hosts which have a special role in the network and they create the backbone for the whole algorithm. Using them the hosts can predict their position while no high network traffic is generated.

The equation establishing the landmarks is a matter of seeking the minimum of the following function:

$$f\left(c_1^S; \dots; c_N^S\right) = \sum_{\mathcal{L}_i,\mathcal{L}_j \in \{\mathcal{L}_1 \dots ,\mathcal{L}_N\} | i<j}^{N} \varepsilon\left(d_{\mathcal{L}_i\mathcal{L}_j}; d^{\mathrm{S}}_{\mathcal{L}_i\mathcal{L}_j}\right), \tag{12}$$

$$N = n_{\mathcal{L}} * D, \tag{13}$$

the variable $n_{\mathcal{L}}$ stands for a number of landmarks, $D$ is the space dimension, $c_i^S$ is a coordination of a landmark $L_i$ in synthetic space, $d_{L_iLj}$ stands for the distance measured between the landmarks $L_i$ and $L_j$, $d^S_{L_iL_j}$ stands for calculated distance between the landmarks $L_i$ and $L_j$, $\varepsilon$ stands for the square of error, and function $f$ stands for the total sum of errors, for which we seek the minimum. See [1], [2], [6], [7] for a more detailed explanation of the algorithm. The measurement of distances is performed using the Internet Control Message Protocol (ICMP) protocol [16]. This protocol is used, for example, by the ping tool, which is available under many operating systems. The protocol measures the time delay between the initial packet transmission request and receiving the echo from the "pinged" host. This time is the so-called round-trip time (RTT).

Although the equation seems to be quite complex, it is based on a simple idea. The known variables are the distances measured and the unknown variables are the coordinates of landmarks, which will best fit the values measured. The number of unknown variables is expressed by formula (13). Each dimension of each landmark stands for a variable. The best host placement is found when the total function error is minimal. The equation thus takes the matrix of distances measured between all the landmarks, compares these values with the matrix of computed values and creates the sum of square of these deviations. Seeking optimal landmark coordinates is a matter of seeking for the minimum of function (12). Using this equation, we are even able to establish a set of landmarks from regular hosts, whose position is not know, but without any relevance to a real coordinate system (e.g. position on the map).

The second part of the algorithm localizes regular hosts. It is similar to the previous one, but now we aim to estimate the coordinates of a single host. The known variables are the RTT distances between the host and each of the set of landmarks that the host can measure. Then the estimation of the host coordinates is a matter of seeking the minimum of the following function:

$$f(c_H^S) = \sum_{\mathcal{L}_i \in \{\mathcal{L}_1 \dots ,\mathcal{L}_N\}}^{N} \varepsilon(d_{\mathcal{L}_iH}; d^{\mathrm{S}}_{\mathcal{L}_iH}) \tag{14}$$

In the case of equation (14), it is a D-dimensional function (see Equation (13)) and the total deviation between computed and measured values between the host $H$ and landmarks $L_i \in \{L_1 \dots, L_N\}$ is computed.

### B. *Distributed coordinate systems*

Other approaches can also use distributed coordinate systems. Their major representative is so-called Vivaldi method [19], which has also many variants and improvements that have been introduced recently, such as Myth [20], Pharos [21] and others. All can improve its accuracy and shorten the time of convergence to accurate values. These methods are commonly based on theoretical physical model of spring mesh, which are placed between hosts and the tension among these imaginary springs leads to minimize energy in the set of localized hosts. In terms of its accuracy the distributed coordinate systems are comparable with GNP, however its time of convergence to relatively accurate values is in case of Vivaldi significantly higher. Another issue is that it generates permanent traffic during the time of the session. When the network conditions are static, there is no need to measure the RTT distances again. On the other hand, the advantage of it is, that the structure of network is continuously adapted to the current network conditions, and therefore in dynamic network environments it would give better results. Furthermore, it seems that in decentralized coordinate it systems is quite difficult to utilize HA overhead for round-trip delay time measurement, which is necessary for every coordinate systems.

In some cases there might be beneficial to use a hybrid approach – distributed coordinate systems for feedback target (FT) stations and centralized GNP for receivers. Suitability of this approach strongly varies from case to case depending on the network type and network environment. If there are expected some network changes it is suggested to consider to use a distributed Vivaldi version where the session will adapt to actual network conditions. The drawback of this approach is its overhead traffic, which is for the immutable networks needless.

Currently we expect that most of the cases of HA deployment will be in static environments, and therefore the version described here is the centralized one. In this case it is also possible to periodically reset previously determined values and force to reinitialize the coordinates of all the hosts in the session. Thus it can, to some degree, also dynamically adapt to changing environments like the distributed ones.

## IV. Integration of Host Position Prediction into Hierarchical Aggregation

In this section the integration of GNP method with HA is proposed. Because of hierarchical structure of HA, the coordinate system integration quite differs from the most common cases. In the first section we introduce a new session member type: so-called feedback target manager. The next sections describe tree initialization process which is needed for registration of FT hosts to the session.

### A. Feedback Target Manager

As described in section II, in HA method feedback targets (FT) forms a tree, which is able to transmit signaling from huge number of receivers in a short time. This set of FT can be shared among several IPTV broadcasting and several parallel trees exists there. To organize these FT in the desired tree structure we introduce a new member type – so-called feedback target manager (FTM). This is a standalone application, which can be possibly run on the same hardware together with FT. However because of possibility of high network load, it is suggested to place FT on standalone station to reduce the risk of FTM service unavailability.

### B. Tree Initialization

At the start of a session all the FTs need to be registered to FTM. Thus the FTM knows about them and when requested by an IPTV server it can create a new hierarchical tree. As mentioned earlier, there can be several trees sharing this set of FTs. Each tree is identified by a unique number and thus they can be distinguished between each other. During the time of a session, it is also possible that a new FT can join or, on the other hand, a FT can leave. FT leaving from a session can occur due to maintainers request or, of course, unexpectedly due to FT or network failure. For such events, we proposed a protocol, which monitors set of FTs and can detect breakdown of any of them in reasonable amount of time.

### C. First Step – Landmark Backbone Establishment

As said in section III, the GNP algorithm proceeds in two steps. In the first step the landmarks positions are predicted and receivers are informed about the results. In the second step all the receivers predict their positions and select FTs where they will send their feedback. Each receiver selects its feedback according to the distance to FT and according to the number of receivers sending feedback to this FT (in other words, with how many other receivers or feedback targets the newly connected receiver will have to share the $B_{\text{RTCP}}$ bandwidth). The ratio of these two parameters can be changed dynamically.

At the first glance it might seem that we would need an additional set of stations distributed over the Internet for host position prediction. Fortunately, for this purpose we can

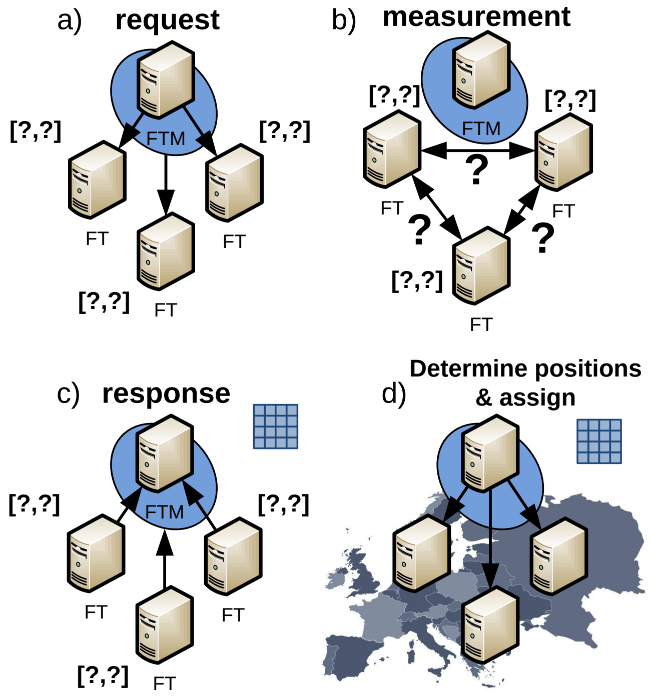


Fig. 2. Hierarchical aggregation scheme with many feedback targets.

utilize an existing set of FT stations as they are already distributed over the broadcasting area (or the whole Internet) and distance measurement generates relatively low overhead traffic in the network. This will have an effect that the traffic load will be distributed uniformly and the overlay network will be better organized, and consequently we can reduce risk of high traffic peaks in the network when some attractive poll is announced.

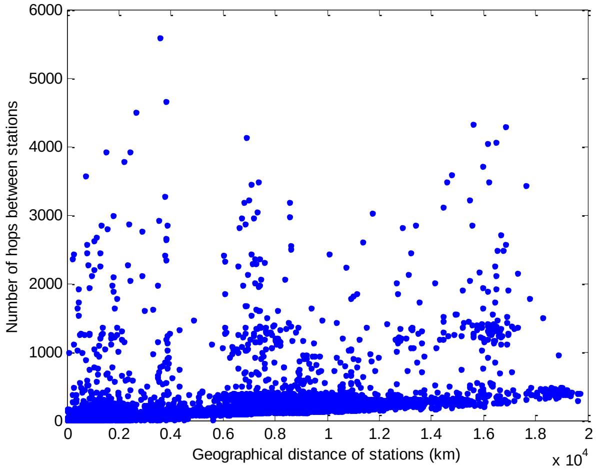


Fig. 3. Dependency of RTT measured between two hosts and their geographical distance. Results were obtained from the Planetlab experimental network. From the graph may not be noticeable, but 90 % of all measurements are in close linearly dependent area

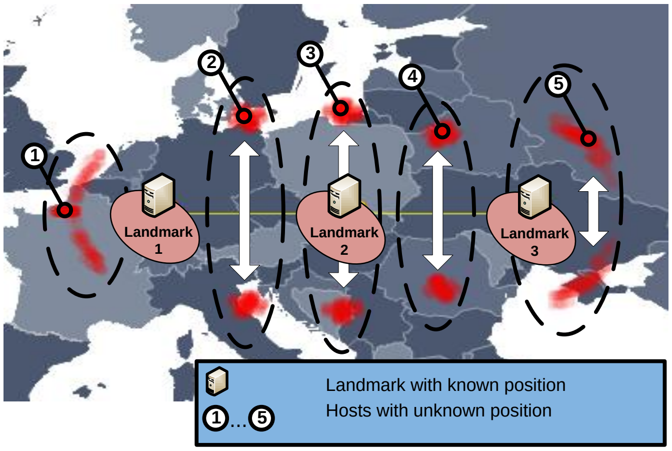


Fig. 4. Inaccurate position estimation when the triangular inequality condition is not fulfilled.

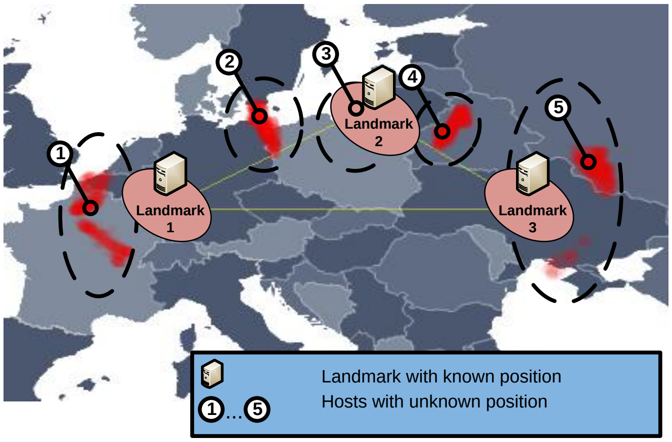


Fig. 5. Inaccurate position estimation when the triangular inequality condition is fulfilled.

Let us look closer at the first step of the algorithm. It consists of 4 subparts: a) request from sender, b) measurement of RTT distances, c) establishment of distance matrix, d) and finally the localization of feedback targets (see Fig. 2). The sender request is transmitted in the RSI packet as a new block type of the RSI sub-report block [15]. When the feedback targets receive the request packet from sender, all the feedback targets will start measuring the RTT distances. To prevent network overflow, this measurement should be spread over the time length of $T_{\mathrm{RSI\_F}}$. Furthermore to reduce the risk of measurement during temporal network problems, the measurement should be repeated several times (e.g. three times). The resultant value should be the minimum of these values measured. Unfortunately, the risk cannot be completely eliminated. In such cases it fails it will affect the position estimation accuracy. As stated before, this measurement of RTT distances is performed using the ICMP [16] protocol. When all the feedback targets have measured the RTT distances to the other feedback targets, they will transmit these so-called vectors back to the sender. When the sender has a complete matrix of distances between the feedback targets, the sender will predict feedback target coordinates using equation (16) and transmit them via a multicast channel, to all the feedback targets and the receivers.

Informing the whole session about the landmark positions may seem to be a waste of bandwidth. However, the RTCP standard recons with possibility of there being one or more senders and therefore 25 % of the total RTCP bandwidth is reserved.

### D. Second Step – Hosts Position Prediction

When the position of unknown landmarks has been predicted and all the landmarks form something like a basic network backbone of the whole algorithm, the position of regular hosts is to be predicted. Thanks to the fact that the number of landmarks is relatively low, the generated network traffic will not be very high.

When a receiver position has been predicted, the receiver should redirect its feedback to the best FT. The optimality of choosing a FT is here a matter of the distance to the receiver and the number of the other receivers reporting to the same FT.

In the section VI. it is also described a way how synthetic coordinate space can be mapped to a real map and how it can be utilized for statistics, e.g. for number of connected subscribers from different areas.

## V. Feedback Target Placement Problem

At first glance it may seem quite surprising that the estimation of a host position can work, although it is based on measurement of RTT between two hosts and the structure of the Internet is quite complex. The Internet topology is based on a tree structure, rather than on the 2D space. However in spite of it, according to several measurements it has been approved [1], [2], [10], [11] that such approach can estimate relative distances between hosts in the network and thus save significantly amount of bandwidth, especially in large-scale environments. In Fig. 4 are depicted results obtained from measurements among approximately 350 stations in the Planetlab network, which spread worldwide. As you can see the dependency of distance on the RTT values are obvious and the most of the measurements forms a linear dependency.

### *A. Triangle inequality problem*

In the coordinate systems is quite complex high-dimensional space mapped into some low-dimensional one. As these spaces are not homogenous, it might cause some degree of errors in some cases. Another factor that strongly affects resulting error of position estimation is the placements of landmark stations (or, in the case of HA, FT stations). The problem lies in the triangle inequality problem – when the landmark stations are in line, the algorithm gives exactly the same probability for prediction on the correct position as for a mirrored position (see Fig. 4). This is caused because the algorithm only considers the distance from host to landmark one, two and three and this value equals, also, the mirrored position. Naturally, the RTT measurement is also not absolutely accurate because of routers and switches latency and other unpredictable conditions and it is supposed that the total error is in average about 9 % ±3 % of the measured distance.

In the Fig. 4 are depicted five examples with error rate 9 % and normal distribution of this error with standard deviation 3 %. The probability of position estimation is depicted as the red area where its saturation stands for the probability of estimation on this position. As you can see on examples 2, 3, 4, 5, when the localized hosts are beside landmarks line, the probability of mirroring its real position is quite high. When the predicted host lies in landmarks line, the probability of host position estimation will not be mirrored, however also the accuracy is quite low (see hosts 1 in Fig. 4).

### *B. Effect of removing triangle*

Now let us compare the previous results obtained with another, slightly modified, selection of landmarks. Rather than choose the landmark 2 in Slovakia (see Fig. 4), we placed it to Poland (see Fig. 5). Thus the triangular inequality condition was fulfilled and even the position of hosts 1, 2, 3, 4, 5 was kept the same as in the previous case. The results of host positions prediction is considerably better.

Of course, it is not always possible to choose from several FTs, especially in smaller networks. However if so, there is still high probability that the hosts will be formed in more effective manner than would be formed when chosen randomly. However, often there is a possibility to choose from several hosts and as such topology should serve for IPTV sessions for a long time. Therefore there could be motivation to address this issue. Because of presence of RTT measurement inaccuracy, it is not possible to rely on classical mathematical condition of triangular inequality. An example of this can be found in Fig. 5 – the landmark nodes 1, 2, 3 are not in line, however you can notice, that host no. 5 has some probability to be mirrored and the position can be predicted under the triangle consisting by vertexes landmark 1, landmark 2 and landmark 3.

### *C. Triangle inequality identification*

To better evaluate if a selection of landmarks is good or not, the modified version of triangular inequality condition is introduced here in equation (15). Input parameters of this function $a, b, c$ are RTT distances between any three hosts, where $c$ must be greater or equal to $a$ and $b$. $T$ stands for a threshold. When $T$ equals zero, all the combinations of hypotenuse and catheti will be considered as to be correct and the function $I$ will return 0 (false) as the condition was not be violated.  On the other hand, when threshold $T$ equals one, only the equilateral triangle will be detected as to be correct.

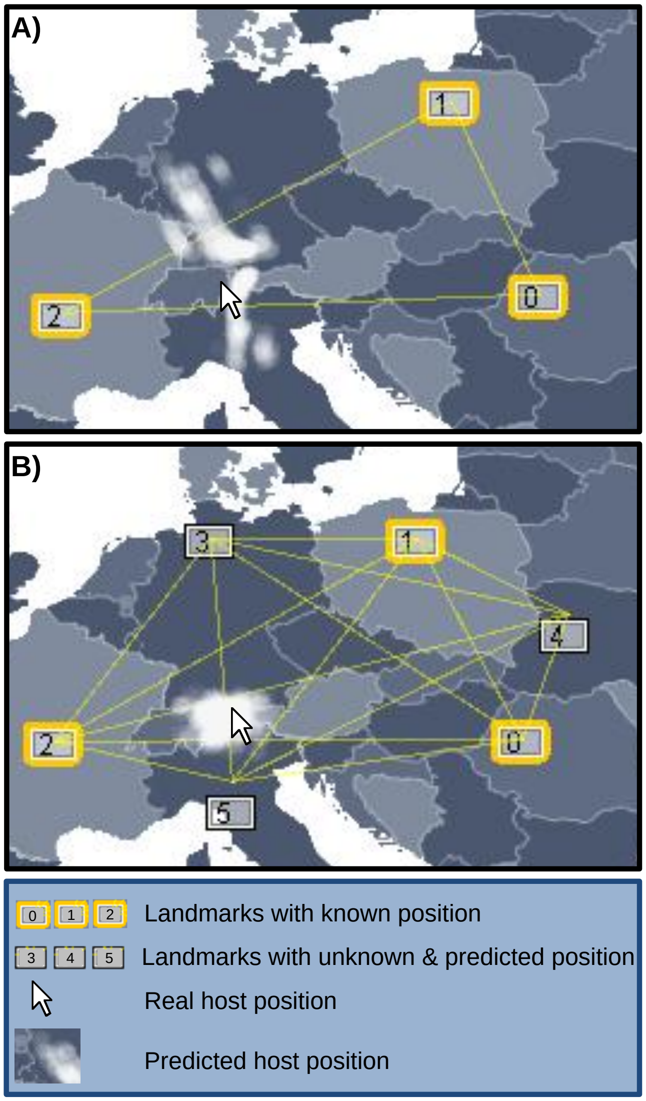

A)
1
2
0
B)
3
1
4
2
0
5
0 1 2 Landmarks with known position
3 4 5 Landmarks with unknown & predicted position
Real host position
Predicted host position

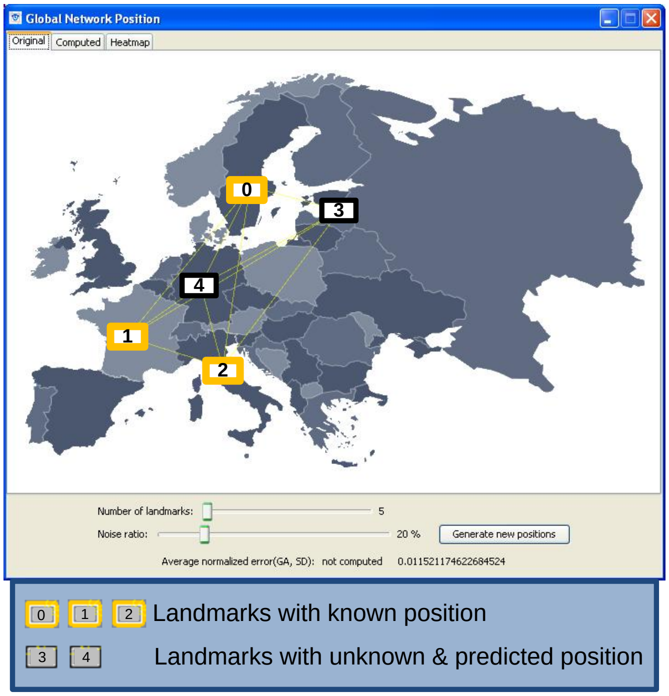
Fig. 7. Inaccurate host position prediction and its improvement with computationally predicted three new landmarks. Noise ratio was set to 20 %.

$$I(a,b,c) = \begin{cases} 0, & if\ \dfrac{a+b}{c} - 1 \geq \mathrm{T} \\ 1, & \text{otherwise} \end{cases} \qquad (15)$$
$$\text{where } a \leq c \ \wedge b \leq c, T \in < 0,1 >$$

According to our empirical experiments it seems that value of 0.4 is sufficient, but it strongly depends on concrete landmarks placement. Of course, it strongly depends on a particular host positions and with more than 3 landmarks the results are also slightly different. To identify possible problems when deploying FT stations (landmarks) we developed a simulation with tool which is appropriate for a given type of coordinate system[1].

### D. Triangle Inequality Identification

For the purpose of identifying the origins of the resulting prediction error when the GNP algorithm is used, a simulation tool has been developed, which has several options to be set (see Fig. 6). They are: the number of landmarks; noise ratio; and actual position of each landmark. The number of landmarks can grow from three, which is the minimal usable value for estimation in 2D space, up to one hundred. The landmarks positions can be set by mouse dragging and each landmark can be marked to indicate whether its position is known or unknown.



Number of landmarks: 5

Noise ratio: 20 %

Average normalized error(GA, SD): not computed 0.011521174622684524

Landmarks with known position

Landmarks with unknown & predicted position

Fig. 6. GNP simulation tool with a config panel and landmarks position.

Known landmarks are shown with bold yellow borders (landmarks no. 0, 1, 2) (see Fig. 6) and unknown landmarks are shown with thin black borders (landmarks no. 3, 4, 5; see Fig. 6). The positions of these landmarks will be predicted in the first part of the algorithm (4). If the distances between landmarks and hosts were measured absolutely accurately, also the RTT distances would correspond accurately to the map distances. However this does not correspond to reality. In real network conditions, the round-trip time and the real-map distance cannot be mapped absolutely accurately because of the difference between Euclidean space and network structures. To take this inaccuracy into account the option "noise ratio" assists. It can scale from the value of 0 %, which stands for absolutely accurate measuring, to 100 %. The value 100 % says that such a virtually measured RTT value is obscured by noise ranging from ±100 % which means range from 0 % to 200 % of its original distance.

Let us assume a case when we know the position of only a limited number of landmarks (e.g. three) and they are placed as depicted in Fig. 7A – one in Romania, one in Poland and one in France and all the RTT measurement is obscured by 20 % noise. In this case the position prediction for the host placed in Germany is quite inaccurate (see Fig. 7A). The probability of position prediction is spread over a huge area beginning in the centre of Germany through Switzerland, Austria up to Italy.

As the simulation results have shown, the algorithm, for such a configuration, does not give very good outcomes. Especially when the host is near the French landmark, the prediction can be affected by quite big error with a big diversion from the real position. It is obvious that a new landmark should be added, which should be placed somewhere near Germany. If we had enough landmarks and their positions, it would be quite easy. We would just select a suitable passive landmark from this area and then activate it. However, when the network structure is dynamically changing and new landmarks must emerge dynamically, it might be problem.

## VI. Mapping from Imaginary Space onto Real Position Space

Except reduction of bandwidth used, the integration of HA with coordinate systems can also offer estimation of subscriber positions. For this purpose there is better idea to use instead of a synthetic coordinate space use a one mapped to a real world coordinates, e.g. geographical map or the GPS space.

What should be also emphasized is that internet coordinate systems were not proposed for accurate position prediction on a geographical map and, therefore, they do not give such accurate results as other methods can give. Their main objective is to allow building more effective overlay structure. As they are already deployed, they can give approximate estimation of receiver positions with only a little overhead.

In a real network, a set of all landmarks $\mathcal{L}_\mathrm{A} = \{\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_N\}$ can be divided into two separate subsets: set of landmarks, whose positions are unknown (denoted $\mathcal{L}_\mathrm{U}$) and a set of landmarks whose positions are well known (denoted $\mathcal{L}_\mathrm{K}$). Obviously, for hosts with known position we do not need to estimate their coordinates. When we take this fact into account, equations (12), (13), and (14) can be changed to the following forms:

[1] http://adela.utko.feec.vutbr.cz/projects/global-netwok-positioning.html

$$f_{\mathrm{obj}}\left(c^{\mathrm{S}}_{\mathcal{L}_{\mathrm{U1}}},\dots,c^{\mathrm{S}}_{\mathcal{L}_{\mathrm{UN}}}\right)=\varepsilon_{\mathrm{CD}}+\varepsilon_{\mathrm{DD}},\quad \mathcal{L}_{\mathrm{U1}},\mathcal{L}_{\mathrm{U2}},\dots,\mathcal{L}_{\mathrm{UN}}\in\mathcal{L}_{\mathrm{U}} \tag{16}$$

$$\varepsilon_{\mathrm{CD}}=\sum_{\mathcal{L}_i\in\mathcal{L}_K,\mathcal{L}_j\in\mathcal{L}_U}\varepsilon\left(S\cdot d_{\mathcal{L}_i,\mathcal{L}_j},d^{\mathrm{S}}_{\mathcal{L}_i,\mathcal{L}_j}\right), \tag{17}$$

$$\varepsilon_{\mathrm{DD}}=\sum_{\mathcal{L}_i,\mathcal{L}_j\in\mathcal{L}_U|i<j}\varepsilon\left(S\cdot d_{\mathcal{L}_i,\mathcal{L}_j},d^{\mathrm{S}}_{\mathcal{L}_i,\mathcal{L}_j}\right), \tag{18}$$

$$f_{\mathrm{obj}}\left(c^{\mathrm{S}}_{\mathrm{H}}\right)=\sum_{\mathcal{L}_i\in\{\mathcal{L}_1,\mathcal{L}_2,\dots,\mathcal{L}_N\}}\varepsilon(d_{\mathrm{H},\mathcal{L}_i}\cdot S,d^{\mathrm{S}}_{\mathrm{H},\mathcal{L}_i}), \tag{19}$$

$$S=\frac{\sum_{\mathcal{L}_i,\mathcal{L}_j\in\mathcal{L}_{\mathrm{K}}}\left(\frac{d^{\mathrm{S}}_{\mathcal{L}_i,\mathcal{L}_j}}{d_{\mathcal{L}_i,\mathcal{L}_j}}\right)}{\binom{N}{2}}=\frac{2\cdot\sum_{\mathcal{L}_i,\mathcal{L}_j\in\mathcal{L}_{\mathrm{K}}}\left(\frac{d^{\mathrm{S}}_{\mathcal{L}_i,\mathcal{L}_j}}{d_{\mathcal{L}_i,\mathcal{L}_j}}\right)}{N^2-N}, \tag{20}$$

where $\varepsilon_{\mathrm{CD}}$ stands for the total sum of squares of deviations of computed distances from measured distances. These distances are computed among the relation $\mathcal{L}_{\mathrm{K}}\times\mathcal{L}_{\mathrm{U}}$, i.e. between hosts from set of known landmarks $\mathcal{L}_{\mathrm{K}}$ and from a set of unknown landmarks. $\varepsilon_{\mathrm{DD}}$ stands for the total sum of square deviations between computed and measured distances. Here the distances are computed among the relation $\mathcal{L}_{\mathrm{U}}\times\mathcal{L}_{\mathrm{U}}$, i.e. between all the hosts from a set of unknown landmarks.

To make it clearer, see Fig. 8, where hosts with known positions are marked (K) and hosts with unknown and predicted positions are marked (U). What should be emphasized is the fact that distance between the known landmarks (K) is used only for the computation of scale factor *S* (20), and not for the estimation of landmarks position, where it is obtained by calculating an average ratio between RTT distances and distances in the coordination space.

What is newly introduced here is the scale factor *S* and the function $f_{\mathrm{obj}}\left(c^{\mathrm{S}}_{\mathcal{L}_{\mathrm{U1}}},\dots,c^{\mathrm{S}}_{\mathcal{L}_{\mathrm{UN}}}\right)$, which computes the total deviation of computed-space and RTT-space distances among set of known landmarks and a set of unknown landmarks. The scale factor is a mean of the RTT among all the landmarks of the set of known landmarks $\mathcal{L}_{\mathrm{K}}$. The equation compares the measured distances $d^{\mathrm{S}}$ with the real distance *d*, e.g. on a map. By the use of it, the measured values can be scaled to be comparable with the real positions on a map. In the case of GNP positioning, the algorithm is based on imaginary values that have no reference to any real distance or position. As the RTT distances and the real distances may not be homogenous spaces, the *S* may involve an error. At any rate, the value of round-trip time correlates with the distance values and therefore can be used as estimate for the network position.

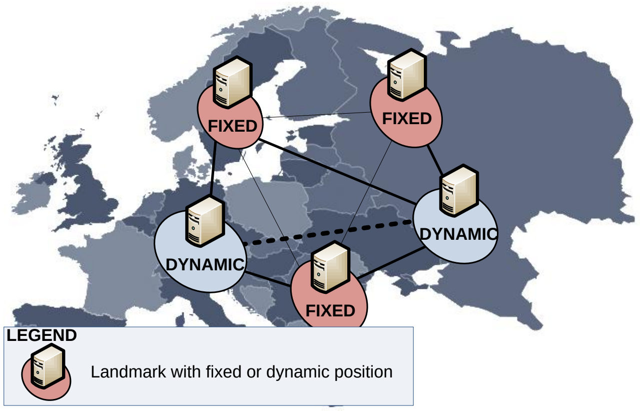


Fig. 8. Set of landmarks with known position (K) and set of landmarks with unknown position (U).

Equation (16) has a similar function as original equation (12) except that it does not change the position of known landmarks $\mathcal{L}_{\mathrm{K}}$. It only predicts the position of the unknown landmarks, which belong to the set $\mathcal{L}_{\mathrm{U}}$. Furthermore, thanks to the scale factor, there are not RTT units of milliseconds but some other units (kilometers / meters) that fit better to the distance quantity.

With these equations the section IV. D can be extended with following few things: all the hosts receive from the multicast channel information about the network scale factor *S* (20), about all the landmarks and their coordinates and, in addition, they can measure the RTT distances to the landmarks (or FTs in case of HA). This is all that is needed to predict their own positions (see formula (19)). As mentioned before in this text, the RTT measurement should be performed several times to minimize the chance of affecting the RTT measurement by some network problems. When the receiver knows the RTT distances to all the landmarks, it can start predicting its own position using formula (19). In fact, it is a matter of seeking such coordinates for which the equation gives a minimum error. For this purpose, some multidimensional optimization algorithm should be used such as the simplex downhill, a gradient method or a kind of genetics algorithm, which can give most accurate results especially for bigger numbers of FTs.

As the hosts positions are scaled using the scale factor *S*, the values can have relevance to some real positions.

## VII. Benefits of Integration of Hierarchical Aggregation with Internet Coordinate Systems

The effect of integrating internet coordinate systems with HA as shown in Fig. 9. In this figure all the stations are in both cases A) and B) on the same position. The difference is that in the case A) receivers select the target for their feedback reports randomly. In the case of B) all the receivers select the nearest one using internet coordinate system. It is obvious, that the communication in the case B) is significantly more effective and in this particular case the bandwidth on some routing points has been saved by up to 37 %.

Of course it is also possible to localize receivers simply by measuring RTT distance and selecting the one, which is the nearest one. A narrow neck of such a solution is that when the set of FTs is shared among several IPTV broadcasting, all the time when the program is switched, it might use a different stations and the measurement have to be repeated. Second and even more important issue arise when the receiver is mobile and its position changes in time. In this case the measurement should be periodically repeated and this would generate significant additional overhead traffic not even at the beginning of the session but continuously during the entire time of the session.

## VIII. Scalability Scenarios

When the classic Hypertext Transfer Protocol (HTTP) is used, the speed of signaling transmission is limited only by the capacity of a network. This approach has a disadvantage in that it may lead to traffic peaks and might affect other services, in particular when an attractive program is broadcasted. The RTCP protocol is designed to deal with such an issue. It uses a constant bandwidth and when the number of users grows, the time period for receiver signaling grows too. Thus the traffic is spread in time and traffic peaks can be avoided. However, the disadvantage of such an approach is that for a big number of receivers the resulting period might become rather long. A simple solution can be assigning more bandwidth. It is expected that the feedback channel will be used not only for a simple monitoring of QoS, as used today in the RTCP protocol, but also for new value-added services such as interactive TV. Therefore it is supposed that it will not be a big barrier for IPTV service providers to assign more bandwidth than currently defined in standard RTCP. However, especially in bigger countries and in the case of multination programs (sporting events), the number of viewers can achieve even tens of millions of viewers at a time. Particularly in such scenarios, the compromise between bandwidth and signaling propagation period is not sufficient and can lead to high bandwidth consumption and long propagation time periods.

The HA brings a new architecture where compromising between time and bandwidth is extended to a number of FTs. The advantage of HA is that it can, in addition, significantly reduce the traffic in the network a) by spreading the load between several FTs and b) by aggregating receiver signalizations at the nearest FT. Here the aggregation can significantly reduce the length of the message (it is a kind of histogram and thus the length of the packet can remain constant for almost any number of receivers in the session). In Fig. 10, several scenarios of the dependence of resulting signaling propagation time on the number of receivers and bandwidth assigned is depicted. All of them suppose that the HA tree is ideally balanced and the number of receivers is, except Figure 10 b), in the range of 1 to 25 million and the feedback channel bandwidth scales from 128 kbps to 3.2 Mbps. The cases a) and b) depict exactly the same scenarios, the only difference is that b) is focused on the area where the resulting propagation time is below 15 seconds and thus limited to the number of receivers from 1 to 250 000. The cases with 1 FT, 10 FTs, 30 FTs, 50 FTs, 100 FTs, 200 FTs and 500 FTs are depicted, where the case with 1 FT stands for RTCP standard. The area where the resulting signaling propagation time is below 15 seconds is marked in blue color.

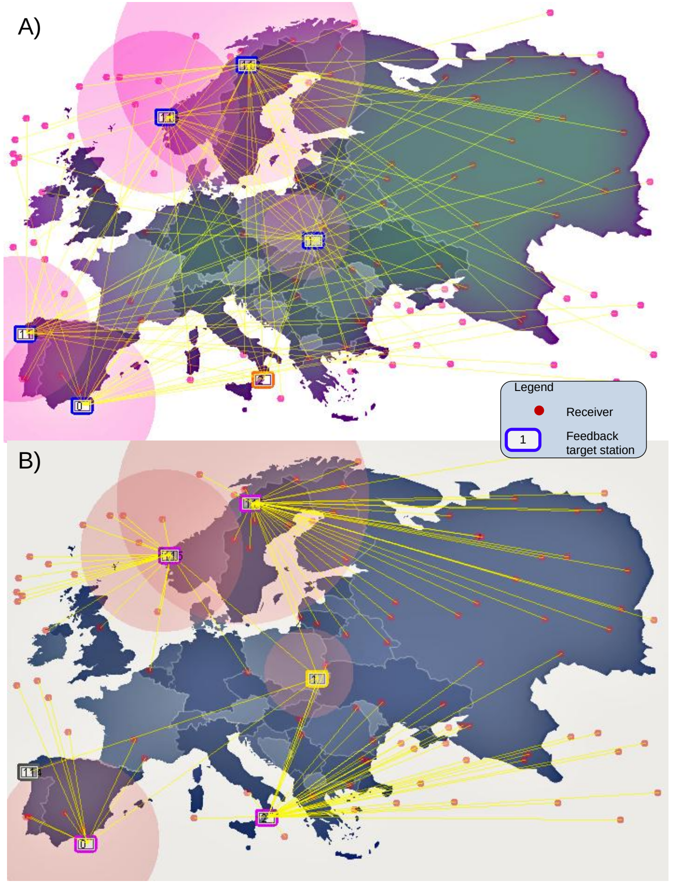


Fig. 9. Example of A) randomly selected FT stations and stations and B) selection using internet coordinate systems. The positions of receivers and FT stations are the same in both cases.

## IX. Deployment of HA in a Real IPTV Infrastructure

The original motivation of this work was targeted to design a scalable structure for needs of interactive IPTV service. The idea behind it is to provide to an IPTV service provider a technology, which will be capable to efficiently transmit receiver signaling and can enable fast interaction between viewers and a content provider. Common IPTV service consists of RTP and RTCP protocols. HA builds on the basis of RTCP protocol and only a few changes are needed. Namely adding the internet coordinates system support to receivers and, of course, adding support for a new type of packets and new type of blocks in RTCP messages.

Second scenario is to target the feedback channel to a content provider rather than to each IPTV service provider. Subscribers of several IPTV service providers complemented by regular TV subscribers equipped with access to the Internet can make up a number of viewers and their votes are related only to distributed content.

The proposed idea is also quite general and might be used not only in the field of IPTV service, but also in any case where there is a need for transmitting the signaling from a number of receivers to a single point.

## X. Conclusion

Nowadays we can be witnesses of the growing influence of IPTV on almost all parts of the developed world. According to several independent marketing analyses it seems that this trend will remain at least for further several years. This, of course, will mean more IPTV subscribers. Furthermore, if we take an expansion of mobile multimedia devices into account the growth of number of IPTV subscribers can even accelerate.

HA will provide facilities for future growth in the number of subscribers and will be scalable enough not only for time to time sending response on some poll or question, but can provide a continuous and scalable feedback transmission for all the receivers in the session, which can convey their opinion during the entire duration of a television program. This will also provide a new kind of knowledge as it will be cheap, really fast and easy to get an opinion from all the subscribers.

There are already several solutions how to enable interaction between subscribers and a content provider. In this paper was introduced an improvement of HA method and described the proposed prototype of hierarchical aggregation with internet coordinate systems.

## Acknowledgment

This work was supported by Grant Agency of the Czech Republic - project No. 102/07/1012.

## References

[1] T. S. Eugene Ng and Hui Zhang, "Predicting Internet Network Distance with Coordinates-Based Approaches", INFOCOM'02, New York, NY, June 2002

[2] T. S. Eugene Ng and Hui Zhang, "Towards Global Network Positioning", Extended Abstract, ACM SIGCOMM Internet Measurement Workshop 2001, San Francisco, CA, November 2001

[3] KOMOSNY D., NOVOTNY V. Tree Structure for Source-Specific Multicast with feedback Aggregation, in ICN07 - The Sixth International Conference on Networking . Martinique, 2007, ISBN 0-7695-2805-8

[4] BURGET, R., KOMOSNY, D. Real-time control protocol and its improvements for Internet Protocol Television. International Transaction on Computer Science and Engineering, ISSN 1738-6438, 2006, roč. 2006, č. 31, s. 1 - 12.

[5] NOVOTNY, V., KOMOSNY, D. Optimization of Large-Scale RTCP Feedback Reporting in ICWMC 2007. ICWMC 2007 - The Third International Conference on Wireless and Mobile Communications. Guadeloupe, 2007, ISBN: 0-7695-2796-5

[6] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for Internet applications,”in Proceedings of ACM SIGCOMM’01, San Diego, CA, Aug. 2001.

[7] J.A. Nelder and R.Mead, “A simplex method for function minimization,”Computer Journal, vol. 7, pp. 308–313, 1965.

[8] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, “Topologicallyaware overlay construction and server selection,” in Proceedings of IEEE INFOCOM’02, New York, NY, June 2002.

[9] J. Postel, Internet control message protocol, RFC792 (1981) (September).

[10] M. Costa, M. Castro, A. Rowstron, P. Key, PIC: practical internet coordinates for distance estimation, in: International Conference on Distributed Systems, Tokyo, March 2004.

[11] P. Francis , C. Jamin, C. Jin, Y. Jin, Y., D. Raz, Y. Shavitt, L. Zhang, IDMaps: A Global Internet Host Distance Estimation Service. EEE/ACM Trans. on Networking, Oct. 2001.

[12] S.M. Hotz. Routing information organization to support scalable interdomain routing with heterogeneous path requirements, 1994. Ph.D. Thesis (draft), University of Southern California.

[13] KOMOSNY, D., NOVOTNY, V. Analysis of bandwidth redistribution algorithm for single source multicast In Proceedings of the Sixth International Network Conference. Sixth International Network Conference. United Kingdom: University of Plymouth, 2006, s. 45 - 52, ISBN 1-84102-157-1

[14] J.Chesterfield and E.Schooler, "An Extensible RTCP Control Framework for Large Multimedia Distributions", Proceedings of 2nd International Symposium on Network Computing and Applications (NCA'03), Cambridge, MA, 16 - 18 April, 2003.

[15] J. Ott, J. Chesterfield, E. Schooler, "RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback", IETF draft, AVT-RTCP-SSM, March 2007.

[16] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, USC/Information Sciences Institute, September 1981.

[17] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP - A Transport Protocol for Real-time Applications," RFC 3550 (STD 64), July 2003.

[18] NOVOTNY, V., KOMOSNY, D. Optimization of Large-Scale RTCP Feedback Reporting in ICWMC 2007. ICWMC 2007 - The Third International Conference on Wireless and Mobile Communications. Guadeloupe, 2007, ISBN: 0-7695-2796-5

[19] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: a decentralized network coordinate system," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 15-26, October 2004. [Online]. Available: http://dx.doi.org/10.1145/1030194.1015471

[20] Yang Chen; Genyi Zhao; Ang Li; Beixing Deng; Xing Li, “Myth: An Accurate and Scalable Network Coordinate System under High Node Churn Rate”, Networks, 2007. ICON 2007. 15th IEEE International Conference, Volume , Issue , 19-21 Nov. 2007 Page(s):143 - 148

[21] Yang Chen Yongqiang Xiong Xiaohui Shi Beixing Deng Xing Li , “Pharos: A Decentralized and Hierarchical Network Coordinate System for Internet Distance Prediction”. Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE

**Radim Burget** (born in 1982 in the Czech Republic,)

He graduated from Brno University of Technology, Faculty of Information technology (2003).

He is engaged in research focused on signaling for IPTV systems and data aggregation in sensor networks. Now he works with Dept. of Telecommunications, Brno University of Technology, Purkynova 118, 612 00 Brno, Czech Republic.
E-mail: burgetrm@feec.vutbr.cz

**Dan Komosný** (born in 1976 in the Czech Republic)

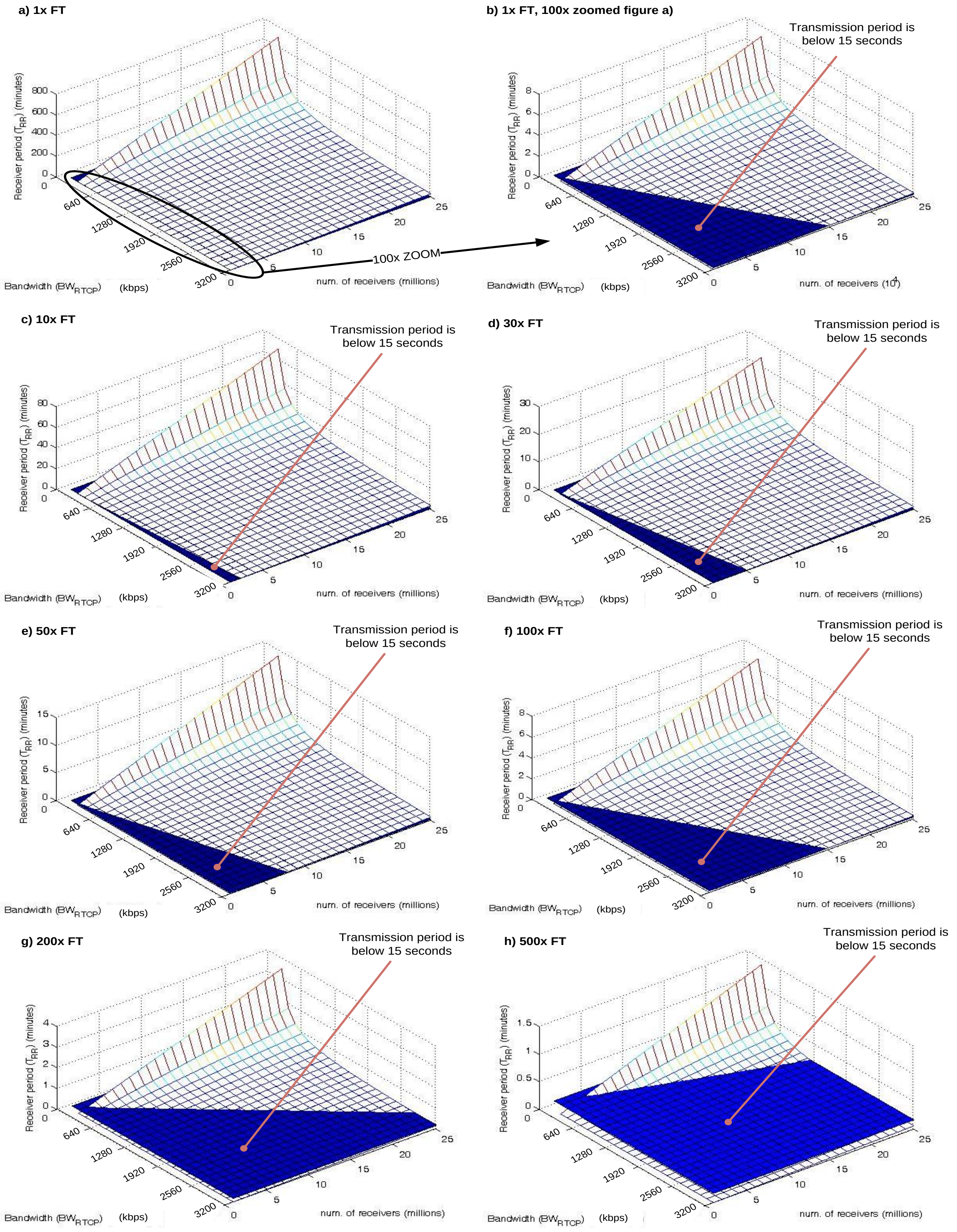
He graduated from Brno University of Technology, Faculty of Electrical Engineering and Computer Science in the field of Electronics and Communications (2000), Ph.D. in Teleinformatics (2003).

He is engaged in research focused on voice transmission over IP network (VoIP). He also focuses on the development of e-learning tools using formal and visual languages – the SDL object-oriented design language and the MSC trace language. Now he works with Dept. of Telecommunications, Brno University of Technology, Purkynova 118, 612 00 Brno, Czech Republic.
E-mail: komosny@feec.vutbr.cz

**Jakub Müller** (born in 1984 in the Czech Republic)

He graduated from Brno University of Technology, Faculty of Electrical Engineering and Computer Science in the field of Electronics and Communications (2008).

He is engaged in research focused on measurement of quality and localization of members in IP network. Now he works with Dept. of Telecommunications, Brno University of Technology, Purkynova 118, 612 00 Brno, Czech Republic.
E-mail: mullerj@feec.vutbr.cz



a) 1x FT

b) 1x FT, 100x zoomed figure a)

c) 10x FT

d) 30x FT

e) 50x FT

f) 100x FT

g) 200x FT

h) 500x FT

Fig. 10. Dependency of signaling propagation time on number of receivers and bandwidth assigned to feedback channel. The blue color marks the areas, where the resulting propagation time is below 15 seconds (10 seconds on the level of receivers and 5 seconds on the level of FTs). The bandwidth is in from 128 kbps to 3.2 Mbps. The number of users are in millions, except of b) where there is depicted the 100x zoomed a) figure.

# Seamless Handover in Heterogeneous Networks using SIP A Proactive Handover Scheme with the Handover Extension

Elin Sundby Boysen
*Norwegian Defence Research Establishment (FFI)*
*UNIK - University Graduate Center Kjeller*
*Kjeller, Norway*
*Email: elin-sundby.boysen@ffi.no*

Torleiv Maseng
*Norwegian Defence Research Establishment (FFI)*
*Kjeller, Norway*
*Email: torleiv.maseng@ffi.no*

## Abstract

*Mobile users move across different types of network, such as WiFi, WiMAX and UMTS. The mobile equipment is already capable of connecting to different network types simultaneously, but in such a heterogeneous environment, session continuity is still a challenge when changing connection from one network to another. The differences in properties on the physical layer and link layer promote higher-layer solutions for handover. Several architectures have already been proposed in either the network layer, in the application layer or as hybrid solutions. The application-layer-based Session Initiation Protocol (SIP) supports terminal mobility, but this procedure suffers from long handover delays. To ensure session continuity during handover we propose a new SIP extension, the Handover header that enables seamless handover. In the handover scheme we propose in this paper, we use the SIP Handover extension with a back-to-back user agent (B2BUA) that is deployed in the home domain of the mobile user. The handover can be assisted by either the B2BUA or the correspondent node.*

*Keywords: SIP; Mobility; Seamless handover; 3G; WiFi*

## 1. Introduction

Modern users of laptops and other mobile equipment want to be online anywhere, anytime using the cheapest or most suitable network, and they expect the technical solution to be ready within a very short time. Many new phones and computers are already equipped with multiple interfaces that allow you to be connected to multiple networks at the same time, like handsets with UMTS and WiFi interfaces. However, a user that switches an ongoing session from one network to another, can experience delays and possibly even session loss. Such delays can be a nuisance when using applications like email or web browsers, but are particularly problematic for real-time sessions such as voice or video, where the user experience can be severely degraded due to the handover delays. How can seamless handover between different types of networks, so-called heterogeneous networks, be accomplished? Although the business side of this is still a matter of discussion, different technical solutions are definitely on their way. In this paper we look deeper into some of the challenges around mobility in heterogeneous networks and suggest an improved version of our Proactive Handover scheme using SIP [1] as one possible solution.

In heterogeneous networks we differ between vertical and horizontal handover. Horizontal handover is between two access points of the same kind, for example WiFi to WiFi handover. In this paper we will focus on vertical handover - the handover between access points of different types like for example UMTS and WiFi. When performing vertical handovers the mobility management protocol must not only provide location transparency, but also network transparency.

To obtain session continuity, there are two main approaches. Either to solve it on the network layer with Mobile IP or on the application layer with augmented existing protocols such as H.323 or Session Initiation Protocol (SIP) [2]. Both the network layer and the application layer approaches have their advantages and drawbacks, and the one does not necessarily exclude the other [3], [4]. We have chosen to focus on the application-layer approach because of its flexibility and ease of implementation.

The rest of this paper is structured as follows. First, in Section 2 we present issues concerning mobility in general and the basics of the SIP protocol. Then we present related work and describe the novelty of this contribution in Section 3. In Section 4, our Proactive Handover scheme is explained and we suggest some changes to it by introducing a new SIP Extension: the Handover header. The proposed solution is discussed and areas of future work are identified in Section 5. Finally, in Section 6 we make some concluding remarks.

## 2. Mobility and SIP

This section will provide the basic concepts of mobility in general, some of the challenges when considering Mobile IP for terminal mobility, and an introduction to Session Initiation Protocol (SIP).

## 2.1. Mobility

Mobility while providing session continuity has until recently been a virtue mainly reserved for operator-controlled mobile systems like GSM and UMTS. These systems consist of overlapping and interconnected base stations and support handover of calls between adjacent cells. A mobile node (MN) report a set of parameters, including their signal strength, and the base stations monitor the traffic load in their cells. The handover is managed by the network and initiated by the base stations.

The Internet had originally no support for session continuity during mobility. The IP addresses are used to describe the point-of-attachment (PoA) of a unit, ergo the location, not the identity of the unit itself. This is not a problem when the units attached to the Internet are stationary, but when the units are moving and need a new PoA, the IP address also usually needs to be updated. This makes session-oriented communication difficult. Mobile IPv4 [5] was introduced as a solution to this problem. Mobile IPv4 allows a node to move from one network to another while keeping the same home address. When the node is not in its home network it is given a care-of-address that is associated with the home address. The corresponding node (CN) uses only the home address. With IPv6 comes also Mobile IPv6 [6]. However, neither Mobile IPv4 nor Mobile IPv6 supports seamless handover. When changing PoA, the MN must realize that the connection with the first PoA is lost before it connects to the new. Lee et al. report average handover delays of 1896ms and 2470ms for two different test cases using Mobile IPv6 [7]. For real-time services this is too long. Maximum handover delays should ideally be less than 100ms, not more than 200ms [8].

Another challenge is that an operational infrastructure that supports Mobile IP must be in place. As these two important challenges –the handover delay and the need for infrastructure– are not yet properly solved; other solutions are being investigated concurrently. One of these is the Proactive Handover scheme. Instead of solving the handover problem on the network layer, proactive handover using SIP is proposed on the application layer. Proactive Handover was first presented in [1] and is explained in more detail in this paper.

A handover has three main phases: *handover initiation* when the mobile node searches for and discovers a new network, *handover preparation* when the node sets up a new link, and *handover execution* when the handover signalling take place and the connection is transferred to the new link. Using SIP we propose a solution to reduce and in most cases eliminate the delays due to each of these phases.

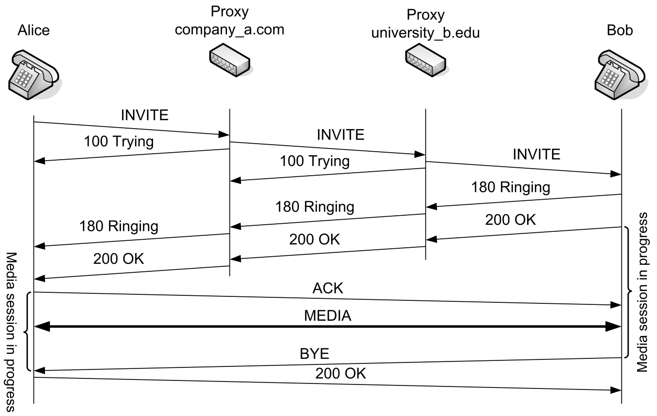


Figure 1. Standard setup of a session using SIP messages

## 2.2. Introduction to SIP

SIP is a protocol designed to establish, modify and terminate multimedia sessions. In short, SIP has been created to make it possible for end points to localize each other. It is an application-layer protocol and runs on top of transport layer protocols like TCP (Transmission Control Protocol), UDP (User Datagram Protocol) or SCTP (Stream Control Transmission Protocol). SIP works with both IPv4 and IPv6.

SIP is based on a request-response transaction model. A client sends a request and a server responds. A SIP message consists of the method name and set of header fields. The six basic methods are REGISTER, INVITE, ACK, OPTIONS, BYE and CANCEL. Later methods like REFER, NOTIFY, MESSAGE, SUBSCRIBE and INFO have been added. The SIP messages do not contain any information about the session itself. Instead, this information is provided in SDP (Session Description Protocol) messages contained in the SIP body.

Figure 1 depicts the signalling flow when setting up a session using SIP messages. In Figure 2 we can see an example of a message. This is a 200 OK response from Bob to Alice answering an INVITE request that was sent from Alice to Bob. In this message, the first line contains the response code, 200 OK. The rest of the message lines are header fields. The header field values of *To*, *From*, *Via*, *CSeq* and *Call-Id* are copied from the incoming INVITE request. In the *From* header field, Alice has included a *tag* parameter. Bob adds his tag parameter in the *To* header field. The two tags and the call id will together make the dialog identity. The three *Via* fields are added by Alice's SIP phone and the two proxies that forward the messages so that the response can take the same route back to Alice. In the *To* and *From* header fields the addresses are on a general form defining a logical recipient. Bob has added the *Contact* header field with his current location 'sip:bob@192.0.2.4' so that subsequent messages can be sent directly to him (as

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP server10.university_b.edu
   ;branch=z9hG4bKnashds8
   ;received=192.0.2.3
Via: SIP/2.0/UDP bigbox3.site3.company_a.com
   ;branch=z9hG4bK77ef4c2312983.1
   ;received=192.0.2.2
Via: SIP/2.0/UDP pc33.company_a.com
   ;branch=z9hG4bK776asdhds ;received=192.0.2.1
To: Bob <sip:bob@university_b.edu>
   ;tag=a6c85cf
From: Alice <sip:alice@company_a.com>
   ;tag=1928301774
Call-ID: a84b4c76e66710@pc33.company_a.com
CSeq: 314159 INVITE
Contact: <sip:bob@192.0.2.4>
Content-Type: application/sdp
Content-Length: 131
```

Figure 2. Example of a SIP response (200 OK).

the ACK, BYE and 200 OK messages in Figure 1) and not through proxies.

A proxy on the initial route may require that it be in the signalling path throughout a session. In this case, the proxy will add the header field *Record-Route* and its IP address or a URI resolving to the address in the INVITE request. As the header field will also be copied into the 200 OK response, both end point will eventually know that messages should be routed through the proxy. The media packets, however, will still go directly between the end points.

SIP consists of different logical elements such as UACs (User Agent Clients), UASs (User Agent Servers), registrars, redirect servers, back-to-back user agents (B2BUA) and proxies. The end points are referred to as UAs (User Agents) and consist of a UAC and a UAS. Proxies can be either statefull or stateless. A stateless proxy server will only forward incoming requests and responses. A statefull proxy on the other hand, will maintain a state for each transaction, -that is which requests and responses belong to that transaction. Redirect servers receives requests and responds to the requester where it should send its request. The B2BUA is an element that can be described as a concatenation of a UAS and a UAC. It acts as a UAS when receiving a request and processing it. When forwarding the request to the corresponding node it acts as a UAC. Unlike a proxy server, it maintains not only transaction state, but also dialog state and *must* participate in all requests sent on the dialogs it has established. Often, B2BUAs also terminate and bridge the media streams to have full control over the whole session. This makes B2BUAs well suited for transcoding between two dialogs, to hide network internals, and for network interworking as it can have protocol adaptation.

SIP inherently supports personal mobility. This means that a user can be found using a single identifier regardless of which location or device (such as PCs, PDAs or phones) he or she is currently at [5]. Terminal mobility is more relevant when introducing wireless access and is the topic of this paper. Terminal mobility allows the user to move around with the device, and the device will roam between different IP subnets. We differ between pre-call mobility and mid-call mobility or in-session mobility. Pre-call mobility is the easiest part, as the MN will only need to re-register its new IP-address with the home registrar each time it changes IP-subnet. The focus on this paper is on mid-call mobility.

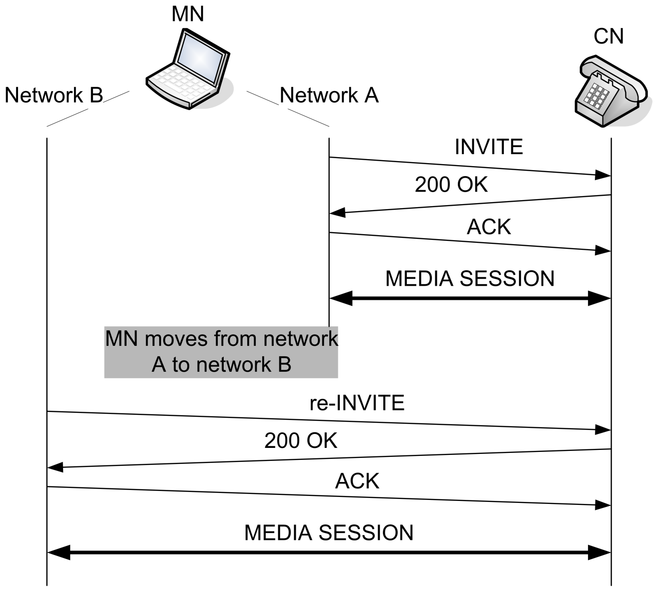


Figure 3. Setup and modification session according to [2] and [9].

The first suggestion for mid-call mobility support in SIP was presented by Wedlund and Schulzrinne in 1999 [9]. When the mobile node (MN) moves from one network to another it simply send a re-INVITE to its corresponding node (CN) telling it about its new IP address. This solution has been included in [2] and it is shown in Figure 3. When a new INVITE message is sent to change an existing dialog, a full description of the session is sent, not only a description of the changes. One challenge by using re-INVITE as described in [9] and [2] is that the handover delay when moving from one network to another can become too long. Referring to the three phases mentioned in Section 2.1, the MN will be disconnected through both the handover initiation (the MN must first realize that it has lost connection with the first network), the handover preparation (it must acquire a new IP address in the new network), and most of the handover execution phase where it sends the re-INVITE.

## 3. Related work and contributions

In this section we will first present some related work before we present the outline of the improved handover scheme presented in this paper.

### 3.1. Related work

Nakajima et al. [10] have analysed the handover delay for SIP mobility in IPv6. They measured handover delays from about 2s to 40s and have shown that this delay is mostly induced because of the Duplicate Address Detection (DAD) in IPv6. DAD imposes a delay from a user receives a Router Advertisement (RA) until the user can send its packets over the interface. The purpose of the DAD is to confirm the uniqueness of the autoconfigured IPv6 address on the link. With a modified kernel that omits DAD Nakajima at al still experience handover delays around 171ms for signalling and around 400ms for the media UDP packets. As in [10] Yeh et al. implements SIP terminal mobility over IPv6 [11]. Their solution also show very long handover delay due to DAD, reaching 1822ms before the media transmission is resumed. Without DAD they experience delays around 218ms for media resumption. The SIP mobility procedure in IPv4 network shows approximately the same delays as in IPv6 without DAD. In [12], Fathi, Chakraborty and Prasad studies SIP session setup delay over UMTS wireless networks with and without Radio Link Protocols (RLPs). They model and evaluate the protocol stacks SIP/UDP/RLP and SIP/TCP/RLP and show that the session setup delay (from INVITE is sent from UAC until ACK is received by CN) is lower for SIP over UDP than for SIP over TCP, around 4.6s using 9.6kbps bandwidth and 2.9s using 19.2kbps. The reason for considering these low bit rates, is that they assume the bandwidth allocated for SIP signalling in UMTS systems to be around this magnitude. These results confirm the results found by Wu et al. in [13], where handover between WLAN and WWAN is modelled. Here, a 128kbps channel in the UMTS network gives a handover delay of approximately 1.5s due to channel loss. For handover from WWAN to WLAN, the delay induced by for instance the DHCP address assignment is more important than the transmission delay of the new INVITE message over WLAN that is less than 1ms.

Several architectures and implementations have been suggested to overcome the challenges of too long handover delays. Some propose a combination of SIP and Mobile IP as Wang and Abu-Rgheff in [4], however most of the effort has been in providing new schemes and architecture to improve SIP.

We differ between *soft* ("make-before-break") and *hard* ("break-before-make") handover. In the hard handover all resources in the first connection are released before establishing a new connection. During soft handover, the equipment is able to communicate over multiple interfaces and thus using resources in both networks simultaneously. Some examples of hard handover are [2], [14], [15] and [16] and an example of soft handover is presented in [17]. Chahbour et al. [14] put forward Hierarchical Mobile SIP enforced by a predictive address reservation to reduce handover delay. Banerjee et al. suggest in [17] to let each base station in each of the wireless technologies (GPRS, CDMA, WLAN etc) be equipped with a back-to-back user agent (B2BUA). On the initiation of a handover the B2BUA of the old access network duplicates the incoming RTP packets and sends them to the B2BUA in the new access network. When the mobile node receives packets through the new interface, it releases the old B2BUA. Another solution that also suggests new entities in the subnets is presented in [15]. Here, Bellavista et al. introduce application-layer middleware to support session continuity. Their Mobile agent-based Ubiquitous multimedia Middleware (MUM) described in [15] and [18] consists of a Proxy Switch (PS) at the ingress of each domain and a Proxy Buffer in each subnet. A Handover Agent Activator (HAA) present in each subnet can activate a Handover Agent (HA) in conjunction with a B2BUA in the Proxy Buffer when a MN enters the subnet. The solution supports both vertical and horizontal handover and is very relevant for data streaming. Packets are being buffered in both the old and the new domain ensuring that no packets are lost while the MN is disconnected during the actual handover. While this solution ensures zero packet-loss, the disconnection time may be problematic for real-time sessions. Tsiakkouris and Wassell suggest in [16] to use location information from the Access Point Location Protocol (APLP) in combination with SIP to anticipate handovers and thus reduce handover delays. By introducing a SIP Mobility Anchor Point in the different domains that can forward media packets from the old to the new address while the MN informs the CN about its new location, packet loss can be avoided.

IEEE 802.21 [19], Media Independent Handover (MIH) is an emerging standard created originally to support handover and interoperability in heterogeneous networks consisting of different technologies in the IEEE 802-series. Later, handover between 802 technologies and non-802 technologies like cellular systems has also become part of the standardization work in 802.21. The scope of IEEE 802.21 is to assist in handover performed on layer 3 and above. It provides link layer triggers or events describing changes in link state or link quality, and network information about available networks and neighbour maps. It can also provide information about load balancing. A node can use the information to perform its own handover procedure or it can in some cases use the 802.21 specific handover commands. 802.21 also makes network initiated handover possible. The use of the 802.21 framework requires that the access points can provide link information through MIH messages. Containers for these messages are currently defined in 802.11u and 802.16g.

### 3.2. Improving SIP Handover

Many of the mentioned solutions for SIP handover provide a faster handover that reduces the handover delay or packet loss. Some of the solutions also introduce new network elements that must be present in either access points or in the subnets. We wish to provide a handover solution that is easily implemented and deployed, and that can support both soft and hard handover. We do not assume that we can deploy network elements in all the subdomains our user might enter, and rely instead on a B2BUA in the home network. We have already introduced our Proactive Handover scheme in [1]. It is summarized here and we propose a new header field, the *Handover* header, to improve the handover scheme. As the SIP protocol is flexible and easy to extend, we suggest this extension that is more in line with the SIP notation than what was previously suggested in [1].

## 4. Proactive handover

In this section we will first give a short summary of the proactive handover scheme as we described in [1]. Then we discuss different ways of triggering the handover before we suggest some improvements to the existing scheme.

### 4.1. The existing scheme

The purpose of the proactive handover scheme is to provide the means to support vertical handover that does not require changes in existing SIP infrastructure and is easily deployable. We assume that the end node user equipment has more than one interface carrying IP traffic, for instance WiFi (IEEE 802.11), Ethernet (IEEE 802.3) and UMTS, and that the user can obtain network access through these simultaneously. We also assume that the user can reach its CN through routes via each of these, i.e. more than one route. The scheme is backwards compatible, meaning that if a MN that is prepared for proactive handover finds that it is communicating with a node that does not support proactive handover, the MN will fall back to the ordinary SIP behaviour.

To promote the success of terminal mobility with SIP, four points are of importance:

1) *Delay:* The handover delay must be short enough not to break an ongoing session or to introduce serious degradation of user experience during the handover. This is especially important in real-time sessions.
2) *Packet loss and jitter:* The packet loss and jitter during the handover should be minimized. In addition to a degraded user experience, too high packet loss or jitter can make it impossible for a streaming session to synchronize and thus interrupt the whole session.
3) *Recovery capabilities:* A good handover scheme should in the case of sudden link loss recover fast enough to prevent sessions from collapsing.
4) *Ease of deployment:* To ease deployment of a handover scheme, the possibility of gradually deployment should be supported.

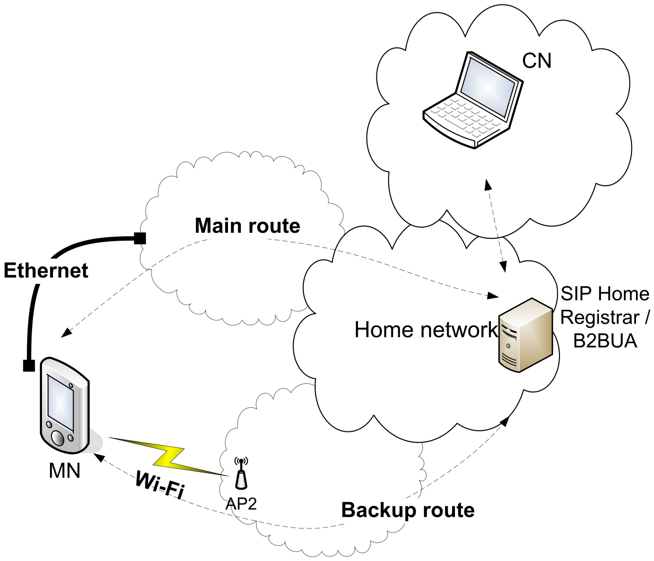


Figure 4. Reaching the CN through two possible routes

As in several of the handover architectures presented earlier, also our scheme suggested in [1] requires the use of a B2BUA. The B2BUA is situated in the mobile node's home network and bridges calls between the MN and its CN. As the B2BUA not only intercepts the signalling messages but also the media transport, the CN does not need to support proactive handover and the two call legs may even use two different codecs. The MN can still switch between different networks as the handover is managed between the MN and the B2BUA. This is depicted in Figure 4.

In the home network there is also a registrar. When the MN registers with the registrar, it registers its main address but also the current address of the backup interface(s). The interfaces on which the MN can be reached are prioritized by adding a parameter *if_q* in the *Contact* header field of the REGISTER request. To make sure that the registering of the backup interface does not overwrite the first register, the parameter *ua_id* is also added to the *Contact* header field. *ua_id* is a random string provided by the user agent and is the same for all main or backup registrations until the user agent re-registers or unregisters. Registering a backup interface can also be done once a session has already been established. When receiving a REGISTER message with the *ua_id* and *if_q* parameters, the registrar will provide a *if_no* parameter in its 200 OK response to the MN. The *if_no* parameter tells the MN how many interfaces the registrar has currently registered. If the registrar does not support proactive handover, it will only ignore the *ua_id* and *if_q* parameters and will consequently not reply with a *if_no* parameter in its Contact header field. This tells the MN that the registrar does not support proactive handover and it will continue its communication as any other SIP node. The message exchange for the register process is shown in

Figure 5.

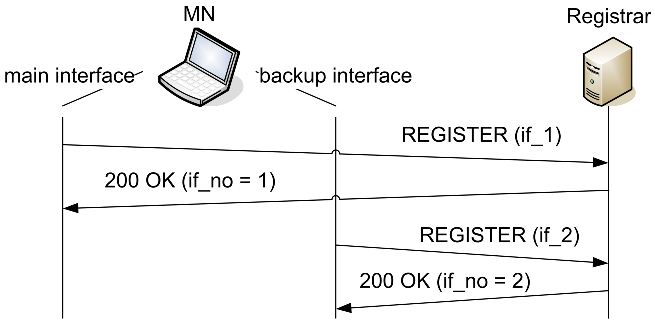


Figure 5. Register two interfaces.

Once a session has been initiated, the MN uses its backup interface to send a new INVITE message addressed to the CN. This message uses the same call-id as the ongoing session, but one of its SDP attributes is set to *sendonly*. The B2BUA will intercept the message and recognize the INVITE message as a backup INVITE. The B2BUA does not initiate a new call leg towards the CN, but replies with a 200OK to the MN. Thus a backup session has been initiated. No media packets will go through this route until a handover is triggered.

When the handover is triggered in the MN it will send a new INVITE request through the backup interface with SDP attribute set to *sendrecv* and the B2BUA will route all packets through this interface. The B2BUA can start sending media over the new interface as soon as it receives the re-INVITE and does not have to wait for the full INVITE - 200 OK - ACK three-way handshake to complete. Figure 6 shows the message exchange for the initialization of the session and the handover. The session is initiated using the main interface. The backup session is initiated, but no media packets go through the backup interface. When the handover is triggered, a new INVITE message is sent over the backup interface and the session is activated. The B2BUA bridges the call legs between the CN and the backup session. When the handover has taken place the register is updated so that the interface that was the backup interface is now set up as main interface.

## 4.2. Triggering the handover

As we have previously defined handover in three steps along a timeline (handover initiation, handover preparation and handover execution), it is important also to mention the handover decision algorithm that triggers the whole operation, and the handover metrics on which the handover decision algorithm bases its outcome.

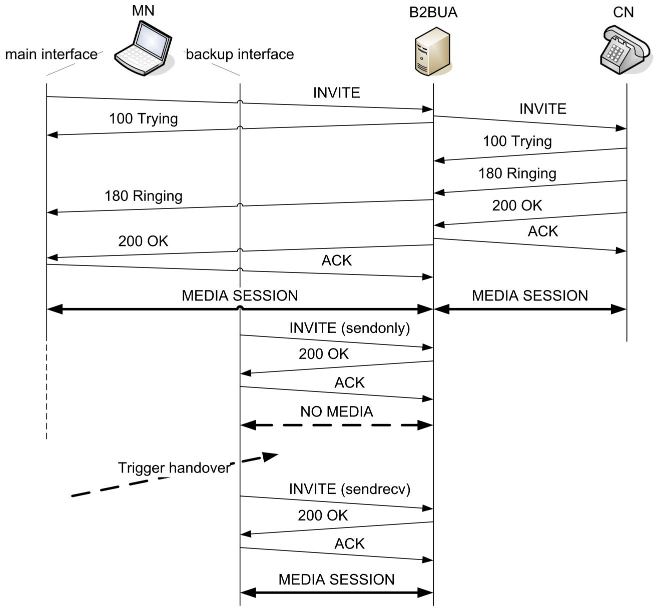


Figure 6. Initiating the session and the backup session.

Proactive handover using SIP [1] only describes how to prepare for and perform the handover. The handover must be triggered by some event that is not specified in the proactive handover scheme. SIP is an application layer protocol and is independent of the type of network that carries the messages. It is important to maintain this independence also when suggesting extensions to SIP. However we wish to use SIP for mobility management by using the handover procedure above. By keeping the handover decision algorithm as a separate procedure, we ensure that Proactive Handover can be used over different network types.

There can be many different reasons why we want to perform a handover. The user can have entered the coverage of a cheaper or otherwise better network to which he also has access. The most obvious reason is of course that the underlying transport link is broken, overloaded or deteriorating. In a heterogeneous network handover metrics should include relative signal strength, link quality, user preference, network conditions, application types and cost.

The link quality can be monitored by measuring frame or bit-error rate, packet jitter, delay and packet loss, as well as signal strength in wireless networks. However, these parameters will have different characteristics depending on the underlying network. A decision to trigger a handover must be made on the basis of which type of network is currently utilized. The cause of a handover in one network type may not be the best for another.

As the proactive handover scheme does not define the handover decision algorithm, this can very well be based on the 802.21 framework mentioned in Section 3.1. However, while we wait for the 802.21 framework to be deployed, any other way of triggering the handover can be used. Other possible solutions include the handover decision algorithm based on location data through GPS and APLP presented in [16]. If the communication takes place in autonomous networks like for instance military networks for tactical communications, one can also envision proprietary solutions based on a mix of public standards and tactical protocols. For instance is the ICMP (Internet Control Message Protocol) Source Quench protocol commonly used in military radios to request the sender to decrease the traffic rate of messages. In practice this means that the operating link has deteriorated. Thus this information can be used as a metric for the handover decision algorithm.

### 4.3. Improved handover scheme

Wu et al. [13] and Fathi et al. [12] point out the problems of long session setup delays, especially when connecting via links like UMTS radio links. Wu et al. show that the data connection setup delay can be in the range of 1500ms. This delay occurs even before the SIP signalling begins, during the data link setup. On the other hand Fathi et al. show that the SIP signalling itself can be very slow. In a normal session setup, the CN can begin sending media packets as soon as it has processed the INVITE message and the corresponding SDP. The MN can start its media session right after it has received and processed the 200 OK from the CN. However, if the conditions are as suggested in [12], that the signalling can have higher delays than the media packets, we risk that MN loses the first packets sent from the CN as it has not yet received the 200 OK from the CN or is still busy processing the 200 OK. If we again consider that the handover delay ideally should be in the range of 50ms - 200 ms, we argue that soft handover techniques must be used so that packets still can flow through the old interface while we wait for the new media route to be set up end-to-end. On the basis of this we propose how the Proactive handover scheme should support soft handover.

As mentioned in [1], one drawback of using a B2BUA to bridge the media streams between the MN and the CN is that it may become a vulnerable hot spot and also become a challenge in terms of scalability. One of the strengths of SIP is indeed that the signalling and the media can take different paths. The main argument for using a B2BUA to bridge both signalling and the media is that the CN does not need to support proactive handover. However, if the CN *can* support proactive handover, it would be better to let the media go directly between the two, making the CN duplicate the packets. While leaving more of the handover duties on the end points, we still want to provide the opportunity for media handling and handover management in the B2BUA for sessions where the CN does not support Proactive handover. To manage this, the B2BUA must know when to bridge the call and when not to.

In the following we suggest some changes to the scheme presented in [1]. We still want to provide a solution that is easy to deploy and that can be managed with only a few changes to the UA requiring the handover and to the B2BUA that assists in it. At the same time we want to improve the previous scheme by utilizing information already available through existing RFCs. We want seamless handover through the use of multiple interfaces that are active concurrently, and a means to make the UAS either in the CN or in the B2BUA start duplicating the media packet. This can be achieved by using a SIP extension, the *Handover* header that we propose here.

The *Handover* header bears resemblance to the *Join* header field defined in [20] and the *Replaces* header defined in [21]. When MN has obtained a new data connection through the backup interface, it sends an INVITE message over the backup interface. This INVITE initiates a new dialog with the CN. The CN replies with the usual 200 OK / ACK handshakes. A Handover header field could look like this:

```
Handover: a84b4c76e66710@pc33.company_a.com
;to-tag=a6c85cf
;from-tag=1928301774
```

In this Header field the MN has included the original *Call-Id*, *to-tag* and *from-tag* so that the CN can identify the right dialog.

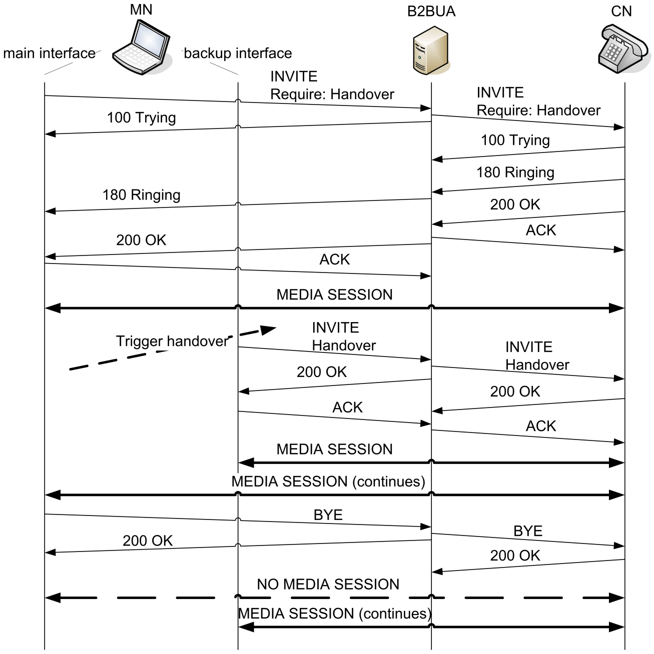


Figure 7. When receiving the INVITE with Handover, the CN sends media to both interfaces.

As described in 7, the CN then start to send the media packets to both interfaces and responds with a 200 OK. As the MN will now receive the same media packets from two directions, duplicate packet detection and filtering is necessary in the mobile node. When the MN sees that the new media stream is of a certain quality, it sends a BYE message over the initial interface and thus stops handling the incoming media packets. The CN will stop duplicating media packets as soon as it has processed the BYE message and then it sends the 200 OK response.

The B2BUA must know when to bridge a media stream and when to leave the handover management to the end points as described in Figure 7. In the initial INVITE the MN uses the *Require* header field listing the option tag *handover*. The B2BUA will forward the INVITE with the *Require* to the UAS in the CN. In the cases where a UAS does not support the extension listed in the *Require* field, the RFC3261 [2] states that the UAS *must* respond with status code 420 Bad Extension and add the *Unsupported* header field where it lists the unsupported extensions required by the UAC. Upon receiving a status code 420 Bad Extension, the B2BUA knows that it will be responsible for bridging the media. This can also occur if the CN realises that it does not have enough resources to handle an eventual handover. The B2BUA sends a new INVITE to the MN without the *Handover* header field. This time the Contact field is changed to indicate the B2BUA's address. When the CN responds with its 200 OK, the media path is set up between the CN and the B2BUA and between the B2BUA and the MN. This is shown in Figure 8. If the CN however does support the *Handover* extension, it will answer with a 200 OK and the media path can be set up directly between the MN and the CN as in Figure 7.

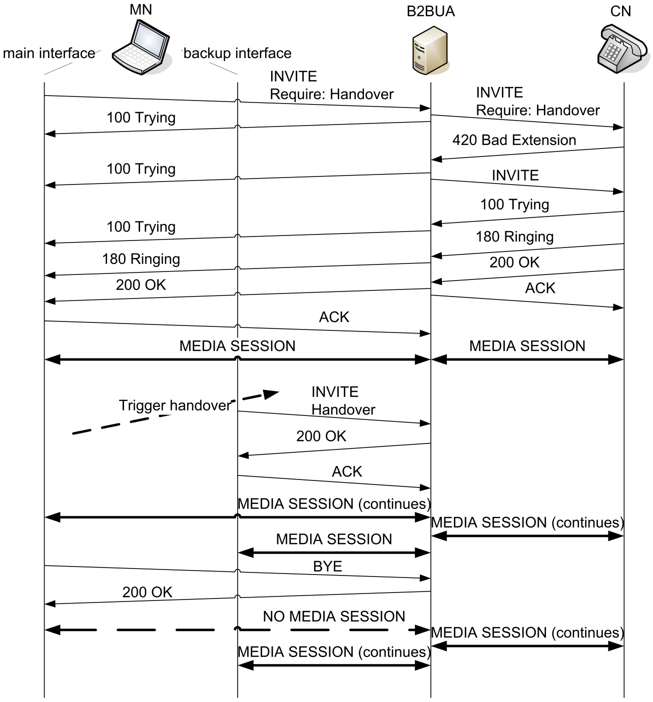


Figure 8. When receiving the INVITE with Handover, the B2BUA sends media to both interfaces. The CN knows nothing about the handover.

The *Handover* extension must require that the sender authenticates himself. The mentioned RFCs [20] and [21] propose mechanisms for this, as all three methods can be a security threat in the form of call hi-jacking. Digest authentication or end-to-end message integrity such as S/MIME are used. The need for authentication was one reason to propose a passive backup session in [1]. The B2BUA could require the backup interface to be authenticated before setting up the backup session. However, as the UA on the mobile node already knows the credentials used in the main session, these can be reused when setting up the backup session.

There may be situations where the original link is so unstable that the MN does not have enough time to initiate the new dialog, perform packet duplication detection, filtering and synchronisation, in addition to closing the dialog using BYE over the old interface. The UAC will then send the BYE message using the new interface to release the resources spent on packet duplication in the UAS that handles the handover management.

Should the main link be broken before we have the time to initiate the handover, a regular INVITE will be sent over the backup interface using the existing dialog id and the *handover* extension will not be used.

## 4.4. Changes from the old to the new Proactive Handover scheme

In comparison with the scheme presented in [1] some changes are more salient than others. In [1] we proposed to register the backup interfaces with the registrar in the B2BUA either before we initiated a session, were they known, or during the session as they became available. Here we suggest that the register process is kept apart form the B2BUA logic as they are indeed defined as different logical entities. The handover shall be completed successfully before the registrar is updated. We do this because a new register message only will be relevant for subsequent sessions. The UAS that handles the handover in the ongoing session will only use the address it has found in the *Contact* header field and is informed of the address change through the new INVITE message. By only contacting the registrar when an address change actually has occurred, this also saves some unnecessary REGISTER transactions.

In this new scheme, one has the opportunity accept the incoming request and to set the media direct on hold with one of the SDP attributes set to *sendonly* as it is done in [1]. However, here, this SDP attribute is not used to determine whether this is a backup session or not, as we use the *Handover* header instead. The reason for suggesting the directly-on-hold solution in the previous solution was that in case of a sudden break of the main link, the re-INVITE would theoretically be quicker than setting up the session from the beginning. If we would have to authenticate the

MN when making the backup session, this would be true. In this new proposal we provide credentials for authentication in the re-INVITE and do not necessarily need the challenge/response mechanism that requires several messages back and forth between the MN and the CN or B2BUA. This means that (given that the authentication is accepted) an INVITE message activating a passive backup session will not be processed faster than the first INVITE request that puts the backup session on hold. Thus, unless a passive session is used to monitor and compare which connection provides the best link properties, a re-INVITE initiating a backup session should only be sent when the actual handover will take place.

## 5. Discussion and future work

With the proposed Handover scheme, we provide a solution for seamless handover in heterogeneous networks. Given that a new session is set up before the old is released, handover delays can be avoided because there will always be at least one functional path between the MN and the CN. Sources of disturbance in the media flow can can occur in the process of handling duplicated RTP packets arriving over two different interfaces and smoothing out any differences in path delays. However, the delays due to differences in the two path delays (the time between CN and MN) is expected to be less than any handover delays occuring due to a break-before-make scheme. In continuation of the proof-of-concept implemented for the proactive handover scheme presented in [1] the new solution is under implementation.

We assume that we can rely on the lower layer mechanisms, instructed by the application, to create a new data link connection based on a set of rules (Examples: ”Set up WLAN connection if I am currently on a UMTS connection as the WLANs are usually faster and cheaper” or ”Whenever I am connected using Ethernet and I discover an accessible WLAN AP, I shall set up a data link connection as long as I am not low on battery”). As we can prepare for the handover by setting up the new data connection while the old is still active, a decision on how much resources are to be used during the handover has to be made by the handover decision algorithm. If we, for instance, wait with the search for and setup of new data links until we actually have an ongoing session, the MN can save unnecessary searches and updates of the backup interface. This saves battery in the MN, but also resources in the network used for backup. This solution is not applicable, though, as it requires a common set of rules for all the applications on the MN that require network access, not just the SIP UA.

We have described a solution that support the use of the CN as the handover assistant when duplication media packets. When the B2BUA tries to find out whether the CN supports the Handover extensions, it will get a 320 Bad Extension in response if the CN does not support Handover. This results in a longer setup delay at the initiation of the session. Further studies will show to what degree this extra delay in the beginning degrades the user experience.

As already mentioned, security will be a very important issue when implementing the Handover extension. This will also be subject to further study, as it is necessary to study whether the security mechanisms suggested in Section 4.3 are good enough to prevent call hi-jacking.

## 6. Conclusion

Mobile users move across different types of network, such as WiFi, WiMAX and UMTS. The mobile equipment is already capable of connecting to different network types simultaneously, but in such a heterogeneous environment, session continuity when changing connection from one network to another is still a challenge. The differences in properties on the physical and link layer promote higher-layer solutions for handover. In this paper we have presented various challenges when handling handover in heterogeneous networks and some of the solutions proposed to overcome them. The application-layer-based Session Initiation Protocol (SIP) supports terminal mobility, but this procedure suffers from long handover delays. Various architectures and procedures have been proposed to manage handover in SIP. However, solutions proposed so far mainly considers reducing the handover delay when disconnecting from one access point and connecting to the new. Some also suggest deploying network elements such as B2BUAs in all the subnets to assist during a handover.

We propose a new SIP extension, the *Handover* header field, which enables seamless handover. The MN will connect to a new access point and set up a data link while the first interface is still connected and a session is active. During the handover period, the MN holds two concurrent sessions to the same B2BUA and receives media packets on both the old and the new data link before the old session is released. If the CN also support the *Handover* extension, the media path can go round the B2BUA and thus reduce the load on the B2BUA. The solution can easily be implemented using a B2BUA in the mobile node’s home domain. Thus, as an example, a VoIP provider can offer his mobile customer support for handover, independent of which domain the he is currently visiting.

## References

[1] E. S. Boysen, H. E. Kjuus, and T. Maseng, “Proactive handover in heterogeneous networks using SIP,” in *Proceedings of the Seventh International Conference on Networking 2008 (ICN 2008)*. IEEE, April 2008, pp. 719–724.

[2] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, “SIP: Session Initiation Protocol,” RFC 3261 (Proposed Standard),

Jun. 2002, updated by RFCs 3265, 3853, 4320, 4916. [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt

[3] H. Lee, S. W. Lee, and D.-H. Cho, "Mobility management based on the integration of mobile IP and session initiation protocol in next generation mobile data networks," in *IEEE 58th Vehicular Technology Conference (VTC), 2003.*, vol. 3. IEEE, October 2003, pp. 2058–2062.

[4] Q. Wang and M. A. Abu-Rgheff, "Mobility management architectures based on joint mobile IP and SIP protocols," *IEEE Wireless Communications*, vol. 13, no. 6, pp. 68–76, December 2006.

[5] C. Perkins, "IP Mobility Support for IPv4," RFC 3344 (Proposed Standard), Aug. 2002, updated by RFC 4721. [Online]. Available: http://www.ietf.org/rfc/rfc3344.txt

[6] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775 (Proposed Standard), Jun. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3775.txt

[7] J. S. Lee, S. J. Koh, and S. H. Kim, "Analysis of Handoff Delay for Mobile IPv6," in *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall*, vol. 4. Los Angeles: IEEE, September 2004, pp. 2967–2969.

[8] ETSI, "TS 101 329 -2 v2.1.3 (2002-01) Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 3; End-to-end Quality of Service in TIPHON systems; Part 2: Definition of speech Quality of Service (QoS) classes," www.etsi.org, 2002. [Online]. Available: www.etsi.org

[9] E. Wedlund and H. Schulzrinne, "Mobility Support using SIP," in *WOWMOM '99: Proceedings of the 2nd ACM international workshop on Wireless mobile multimedia*. New York, NY, USA: ACM Press, 1999, pp. 76–82.

[10] N. Nakajima, A. Dutta, S. Das, and H. Schulzrinne, "Handoff delay analysis and measurement for SIP based mobility in IPv6," in *IEEE International Conference on Communications, 2003. ICC '03.*, vol. 2. Convent Station, NJ, USA: IEEE, May 2003, pp. 1085–1089.

[11] C.-H. Yeh, Q. Wu, and Y.-B. Lin, "SIP Terminal Mobility for both IPv4 and IPv6," in *26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS)*. IEEE, July 2006, pp. 53–53.

[12] H. Fathi, S. S. Chakraborty, and R. Prasad, "Optimization of SIP Session Setup Delay for VoIP in 3G Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1121–1132, September 2006.

[13] W. Wu, N. Banerjee, K. Basu, and S. K. Das, "SIP-based vertical handoff between WWANs and WLANs," *IEEE Wireless Communications*, vol. 12, no. 3, pp. 66–72, June 2005.

[14] F. Chahbour, N. Nouali, and K. Zeraoulia, "Fast Handoff for Hierarchical Mobile SIP Networks," *International Journal of Applied Science, Engineering and Technology*, vol. 5, pp. 34–37, 2005.

[15] P. Bellavista, A. Corradi, and L. Foschini, "SIP-Based Proactive Handoff Management for Session Continuity in the Wireless Internet," in *26th IEEE International Conference on Distributed Computing Systems Workshops 2006, (ICDCSW06)*. IEEE Computer Society, July 2006, pp. 69–69.

[16] S. Tsiakkouris and I. Wassell, "PROFITIS: Architecture for Location-based Vertical Handovers Supporting Real-Time Applications," in *25th IEEE International Performance, Computing, and Communications Conference, 2006 (IPCCC 2006)*. IEEE, April 2006, pp. 629–634.

[17] N. Banerjee, S. K. Das, and A. Acharya, "SIP-based Mobility Architecture for Next Generation Wireless Networks," in *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*. IEEE Computer Society, March 2005, pp. 181–190.

[18] P. Bellavista, A. Corradi, and L. Foschini, "Application-Level Middleware to Proactively Manage Handoff in Wireless Internet Multimedia," in *Management of Multimedia Networks and Services (MMNS)*, vol. 3754. Berlin / Heidelberg: Springer, October 2005, pp. 156–167.

[19] IEEE 802.21, "http://www.ieee802.org/21/index.html," Web page, 2003. [Online]. Available: http://www.ieee802.org/21/index.html

[20] R. Mahy and D. Petrie, "The Session Initiation Protocol (SIP) "Join" Header," RFC 3911 (Proposed Standard), Oct. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3911.txt

[21] R. Mahy, B. Biggs, and R. Dean, "The Session Initiation Protocol (SIP) "Replaces" Header," RFC 3891 (Proposed Standard), Sep. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3891.txt

# 3-CE: A Cooperation Enforcement Technique for Data Forwarding in Vehicular Networks

Yao H. Ho[1], Ai H. Ho[1], Georgiana L. Hamza-Lup[2], and Kien A. Hua[1]

[1]School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816, USA
{yho, aho, and kienhua}@cs.ucf.edu

[2]Department of Computer Science and Engineering, Florida Atlantic University, Port St. Lucie, FL 34986, USA
ghamza@cse.fau.edu

## ABSTRACT

**Operations of vehicular ad hoc networks rely on the collaboration of participating nodes to route data for each other. This standard approach using a fixed set of nodes for each communication link cannot cope with high mobility due to a high frequency of link breaks. A recent approach based on virtual routers has been proposed to address this problem. In this new environment, virtual routers are used for forwarding data. The functionality of each virtual router is provided by the mobile devices currently within its spatial proximity. Since these routers do not move, the communication links are much more robust compared to those of the conventional techniques. In previous work [8], we investigate techniques to enforce collaboration among mobile devices by indentify and punish misbehaving users in supporting the virtual router functionality. The preliminary results showed the proposed 3CE approach is promising. In this paper, we provide a more detail and enhance version of the proposed technique. In addition, we provide more simulation results to indicate that the proposed technique is effective.**

**Keywords:** Cooperation-enforcement; vehicular network; virtual routers; connectionless approach; selfishness.

## 1. INTRODUCTION

Vehicular Network (VNET) has attracted great research interest in recent years. Similar to Mobile Ad hoc NETworks (MANETs), a vehicular network is a self-organizing multi-hop wireless network where all vehicles (often called nodes) participate in the routing and data forwarding process. The deployment of ad hoc vehicular networks does not rely on fixed infrastructures such as router and base station, thereby posing a critical requirement on the nodes to cooperate with each other for successful data transmission. Many works (e.g., [2], [3], and [8]) have pointed out that the impact of malicious and selfish users must be carefully investigated. Existing cooperation enforcement techniques ([2], [3], [8], [10], [11], and [12]) cannot be adapted for recent advance in routing protocols – connectionless oriented approach ([4] and [7]). In particular, we are interested in the *Connectionless Approach for Street* (CLA-S) [6], in this paper. This technique does not maintain a hop-by-hop route for a communication session to minimize the occurrence of broken link. In CLA-S, the streets are divided into non-overlapping grid cells, each serving as a *virtual router*. Any physical router (i.e., mobile host), currently inside a virtual router, can help forward the data packet to the next virtual router along the virtual link. This process is repeated until the packet reaches its final destination. Since a virtual link is based on virtual routers which do not move, it is much more robust than physical link.

The goal of this research is to address the security and cooperation issues for *Connectionless Approach for Street* (CLA-S) in vehicular networks. There can be both selfish and malicious nodes in a vehicular ad hoc network. The selfish nodes are most concerned about their energy consumption and intentionally drop packets to save power. The purpose of malicious node is to attack network using various intrusive techniques. In general, nodes in an ad hoc network can exhibit Byzantine behaviors. That is, they can drop, modify, or misroute data packets. As a result, the availability and robustness of the network are severely compromised. Many works ([2], [3], [8], [10], [11], and [12]) have been published to combat such problem - misbehaving nodes are detected and a routing algorithm is employed to avoid and penalize misbehaving nodes. These techniques, however, cannot be applied to CLA-S since any node in the general direction towards the destination node can potentially help forward the data packets.

The primary contributions of this paper are as follows: 1) We introduce a cooperation enforcement technique, called 3CE (*3-Counter Enforcement*), for the *Connectionless Approach for Street* (CLA-S); 2) We apply the 3CE method to CLA-S; and 3) We present simulation results to show that with the 3CE features, CLA-S can prevent malicious nodes and enforce the cooperation among nodes to maintain the good performance of the network. The remainder of this paper is organized as follows. We review the *Connectionless Approach for Street* (CLA-S) in Section 2. In Section 3, we present our cooperation enforcement technique for CLA-S. We give simulation results in Section 4 to demonstrate the benefits of the proposed techniques. Finally, we draw conclusion on this work in Section 5.

## 2. CONNECTIONLESS APPROACH FOR STREET

To make the paper self contained, we first describe previous work, *Connectionless Approach for Street* (CLA-S), in more detail in this section. In CLA-S, the streets are divided into small "virtual cells." These cells are divided according to intersections and blocks (see Figure 1). Instead of maintain a hop-by-hop route between the source and destination node, the source only needs to maintain the location of the destination. Using this location information, the source dynamically computes and selects a list of grid cells that form a "connecting" path between the source and destination. The location of destination is discovered by the CLA's ***location discovery procedure*** where a simple broadcasting technique [5] is employed. The procedure is as follow. The source will broadcast a Location Discovery (LD) packet that contain source node ID, destination node ID, location (i.e., cell ID) of the source node, and a unique request ID to determine the location of destination. The LD packet will propagate unit it reaches the destination node. When the destination received the LD packet, the destination node will reply with Location Reply (LR) packet. The LR packet includes the location of the source node (i.e., source cell ID) and the location of the destination node (i.e., destination cell ID).

When a node (not the source node) receives a LR packet, it will determine if it is on the grid path using a Reference Line (see Figure 2). Reference Line is the straight line that connects the source (i.e., the center of the source cell) and the destination (i.e., the center of the destination cell). When the source node receives the LR packet, the source node can start communication sessions to the destination node by simply broadcast data packets which contain location information of Source and Destination (i.e., Source Cell ID and Destination Cell ID), Reference Line information, and current cell ID (i.e., cell ID of the node that is about to forward the data packet) in the packet header.

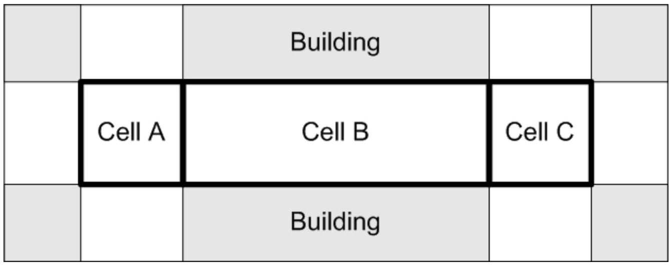


**Figure 1.** Grid path.

Once the Reference Line has been established, we need to determine the reference points. The *reference points* (RP's) on a reference line are the interceptions of the reference line and centerline of either a vertical street or a horizontal street (see Figure 2). Once all reference points of a reference line have been determined, we will use reference points to determine each *Forwarding Zone*. A *Forwarding Zone* is an area that is determined by a reference point or the center of a source cell. A reference point can be on a horizontal block, a vertical block, or an intersection (a block is considered as horizontal if the street it is on has a horizontal orientation; otherwise, it is vertical).

When a node *n* receives a data packet from *m*, the data forwarding procedure is as follows:

1) If *n* is the destination, *n* does not forward the data.
2) If *n* is not in the forwarding zones, *n* does not forward.
3) If *n* or any other node in the cell containing *n* has forwarded, *n* does not forward.
4) If Steps 1, 2, and 3 fail (i.e. *n* might need to forward the data), *n* delays the forwarding.
5) During this delay period, *n* will cancel the forwarding if *n* either hears the same packet from a neighboring node on the same cell or if *n* is in a block cell and *n* hears the same packet from both adjacent intersections.
6) At the end of the delay period, if the forwarding decision has not been cancelled, *n* forwards the data.

When a node receives a packet with a new Forwarding Area (because of a new reference line), it will compute the *Forwarding Zones* and save the result as a list of streets and the ranges of the streets that are encompassed by the *Forwarding Zones*. This allows the node a quick and simple way to determine if it is in a *Forwarding Zone* for subsequent packets with the same Forwarding Area.

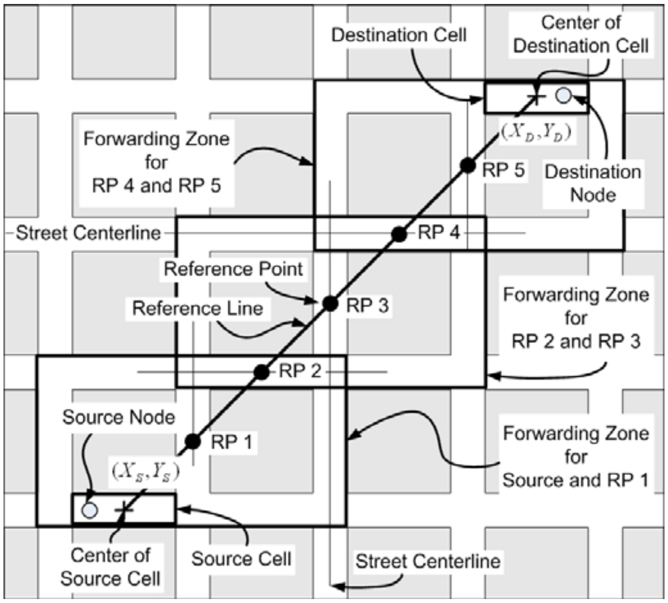


**Figure 2.** Reference Line, Reference Points, and Forwarding Zones

In the above procedure, the delay of a node *n* is computed as follows:

$$DELAY_n = \left| \frac{\alpha}{2 \bullet D\_Dist_n} - \frac{\alpha}{2 \bullet Dist_n} \right| \qquad (1)$$

,where $\alpha$ is a maximum delay constant in $\mu$sec, $D\_Dist_n$ the distance between node *n* and the center of the cell denoted by the *Destination Cell ID* in the packet header, and $Dist_n$ the distance between node *n* and the center of the cell denoted by the *Current Cell ID* (cell of previous relaying node *m*) in the packet header (See Figure 3). The significance of this equation is to select a node farther away from *m* and closer to the destination node to forward the data packet.

If the node *n* is at an intersection of two streets, we will set a shorter delay period. In the simulation, the delay for an intersection node is set to one third of the normal *DELAY*. The reason for this is that, when at an intersection, a node's effective radio range can cover the 2 intersecting streets compared to the single street coverage of another node on a block. The detail information for CLA and CLA-S such as Path Computation, Data Forwarding, Path Update, and Empty Cell/Obstacle Recovery can be found in [6] and [7].

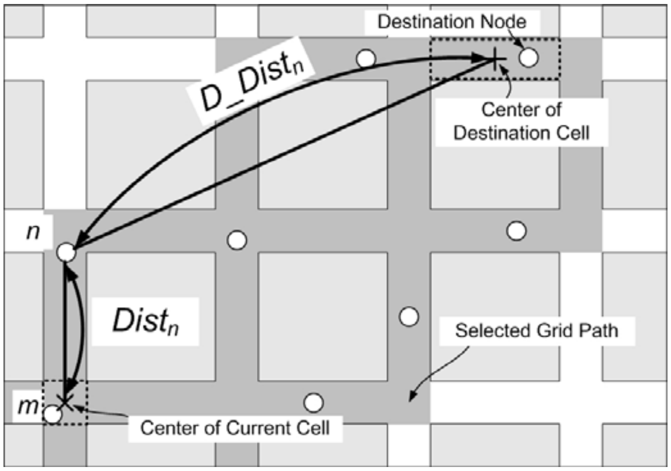


**Figure 3.** Delay for node *n*.

## 3. 3-COUNTER ENFORCEMENT (3CE) FOR COLLABORATION IN FOR CLA-S

In this section, we first briefly describe the configuration of mobile nodes and their Tamper Proof Module. We then present our cooperation enforcement techniques, called 3CE, for CLA-S.

### 3.1 Node Configuration and Tamper Proof Module

The proposed technique is based on nodes with the following configuration. First, nodes are equipped with wireless interface cards that can be switched to detection mode to "detect" data transmission on a "suspicious" node in their proximities. Second, connectionless-oriented routing protocol is employed in the network layer. Without loss of generality, we base our discussion on the more recent techniques developed for routing in VNETs (i.e., Connectionless Approach routing protocol (CLA-S) [6]). Nevertheless, the technique can be incorporated into any location-aid protocols to protect nodes against uncooperative behaviors. Third, reliable communication protocols such as TCP cannot be employed in this type of routing protocols. While other routing protocols need to maintain (proactively or reactively) neighbor nodes location information and establish a connection to the next hop before forwarding a data packet, CLA-S simply forward data packet without first establishing the link to the next node. Any node that happens to be in the general direction towards the destination node can compete for the "right" to forward data packets.

In addition, similar to the techniques presented in [3] and [8], we also equip each node with a tamper resistant module. All other hardware and software components are susceptible to illicit modifications. We notice that a tamper-proof security module remains controversial [13], but it proves to be inevitable in a large scale and high mobility network environment. Our approach guarantees that as long as the tamper resistant module is not compromised, nodes cannot benefit from uncooperative behaviors. Some mission critical data is stored in the tamper resistant module. This information include: 1) a unique *ID* of the node; 2) a pair of public/private keys; 3) a ***Forward Request Counter*** that counts number of packets that are received and need to be forwarded; 4) a ***Forward Counter*** that counts number of packets have been forwarded; 5) a ***Location Discovery Counter*** that counts number of Location Discovery packets initiated by a node; 6) a ***Session Table*** that keeps track ongoing communication sessions; 7) a **Counter Update Procedure** that updates the three counters; 8) a **Misbehavior Detection Procedure** that initiates the detection to identify a malicious node. Since the tamper proof module maintains information of three counters that are used to determine maliciousness of a node and initiate the detection, hereafter we also refer to this module as the 3C Module, and the proposed technique as the 3CE or 3C Enforcement technique.

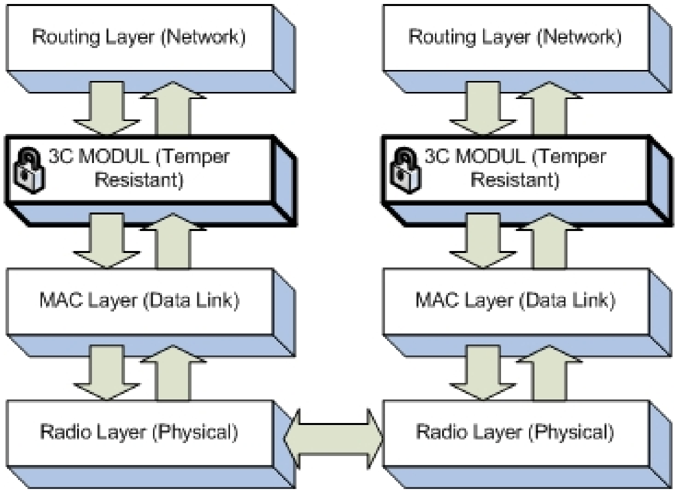


**Figure 4.** Layer Stucture.

The 3C Module inspects Location Discovery packets, Location Reply packets and data packets exchange between the network layer and the MAC layer (see Figure 4); and the module updates the counters as follows:

1. When a new packet arrives at a non-destination node, it updates (i.e., increment by one) its ***Forward Request Counter***;

2. When a node forward a packet, it updates (i.e., increment by one) its ***Forward Counter***; and
3. When a note initiates a Location Discovery packet, it updates (i.e., increment by one) it's ***Location Discovery Counter***.

In addition, the 3C Module constructs and adds 3C's header (i.e., the value of three counters) to the Location Discovery packet as in various layers of the OSI model.

## 3.2 3C Module

In CLA-S, the location of the destination node is needed before a node can start a data transmission session to another node. Thus, a Location Discovery packet is broadcasted to find the destination. Once its location is determined, intermediate nodes can forward data packet according to the general direction towards the destination; and all packets exchanged between nodes are examined by the nodes' 3C Module.

In a 3C Module, three counters (i.e., ***Forward Request Counter***, ***Forward Counter***, and ***Location Discovery Counter***) are updated according to the counter update procedure. These counters are maintained by the node's own 3C Module (see Figure 4). Similar to [3] and [8], we assume the 3C Module is a tamper resistant module that malicious users cannot contaminate it.

When a source node *S* initiates a Location Discovery packet, node *S*'s 3C Module adds the 3C's header to the Location Discovery packet as in various layers of the OSI model. **3C header** contains the value of three counters (i.e., ***Forward Request Counter***, ***Forward Counter***, and ***Location Discovery Counter***) of node *S*. Based on this header, neighboring nodes of *S* can decide to forward or discard the Location Discovery packet. If a node *n* "suspects" the source node *S* is misbehaved, *n* invokes its **Misbehavior Detection Procedure**. A node suspects another node is misbehaving if one of the following is true: a) the *Forward Ratio* (i.e., ratio of *Forward Counter* to *Forward Request Counter*) of *S* falls below the *Forward Ratio* of *n*; or b) the *Request Ratio* (i.e., ratio of the *Location Discovery Counter* to *Forward Counter*) of *S* rises above the *Request Ratio* of *n*. If so, *n* exchanges 3C information (i.e., the value of the three counters) with its neighboring nodes to determine the network condition in the local area (i.e., *n*'s neighboring nodes). If the source node *S* is identified (by **Misbehavior Detection Procedure**) as misbehaving, its neighboring nodes will penalize this node by not forwarding *S*'s Location Discovery packets.

In order for malicious nodes to rejoin the network, non-malicious nodes still allow malicious nodes to participate in forwarding data. Unlike many techniques that avoid the malicious nodes during the routing procedure, our approach allows malicious nodes to rejoin the network by contributing its share (i.e., forwarding data for others) of network workload. This way, nodes are given more incentive to act collaboratively. By forwarding data packets for other nodes, a malicious node can increase its ***Forward Counter***. When its ratio of ***Forward Request Counter*** to ***Forward Counter*** rises above threshold $\alpha$ and its ratio of ***Location Discovery Counter*** to ***Forward Counter*** fells below threshold $\beta$, the malicious node will again be allowed to join the network, i.e., its neighboring nodes again help forward its Location Discovery packets. We elaborate the above processes in the following sections.

## 3.3 Counters Update during the Location Discovery Phase

As mentioned earlier, a node needs to find the location of the destination before it can start to send data packets in connectionless-oriented protocols such as CLA-S. A node can initiate a Location Discovery procedure, receive a Location Discovery packet, or forward/reply a Location Discovery packet. To initiate a Location Discovery procedure, a source node broadcasts a Location Discovery packet.

**Location Discovery packet:** Location Discovery packet contains the following information: source node ID (*source_ID*), source node's location (*S_cell_ID*), destination node ID (*destination_ID*), destination node's location (*D_cell-_ID*), forward node ID (*forward_ID*), and forward node's location (*F_cell_ID*).

When a node receives a Location Discovery packet, it checks if it is the destination node. If so, it returns a Location Reply packet that contains its location (*D_cell_ID*); otherwise, if the node did not see this Location Discovery packet before, it adds its ID and its cell ID (i.e., forward node ID - *forward_ID* and the currently location - *F_cell_ID*) and broadcasts the Location Discovery packet to other nodes. In Figure 5's Routing Layer (i.e., Network Layer), we show the data forwarding procedure for CLA-S.

***Session Table:*** Each node maintains a *Session Table* in its 3C Module to track all the ongoing communication session. An ongoing communication session is identified by a *session_ID* which is a pair of *source_ID* and *destination_ID* of the communication session. This table contains the following information for each entry (i.e., communication session): *session_ID* (i.e., a pair of *source_ID* and *destination_ID*) and a *time to live* (*TTL*) *timer*. An entry is deleted from the *Session Table* when one of the following information is true: (i) A communication session ended; (ii) Entry's *TTL* (time to live) *timer* expired; (iii) Entry belongs to an identified malicious node. An entry's *TTL timer* is reset when a packet received such that: *a*) the packet corresponds to this entry (i.e., *source_ID* and *destination_ID* = *session_ID*) and; *b*) it is not from a malicious node.

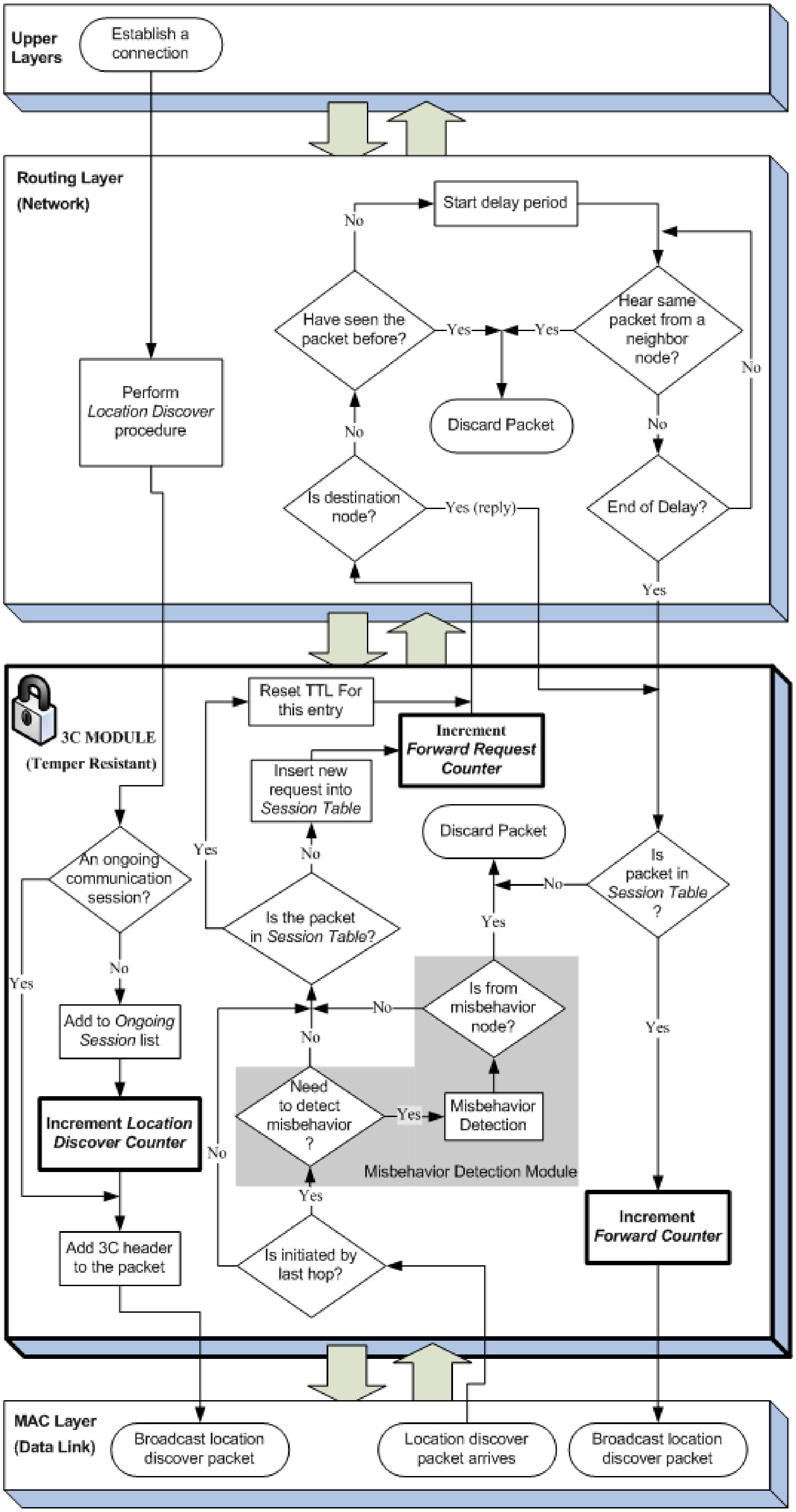

Upper Layers
Establish a connection
Routing Layer (Network)
Start delay period
No
Have seen the packet before?
Yes
Yes
Hear same packet from a neighbor node?
No
Perform Location Discover procedure
Discard Packet
No
No
Is destination node?
Yes (reply)
End of Delay?
Yes
3C MODULE (Temper Resistant)
Reset TTL For this entry
Increment Forward Request Counter
Insert new request into Session Table
Discard Packet
Yes
No
No
Is packet in Session Table ?
An ongoing communication session?
Is the packet in Session Table?
Yes
No
Yes
Is from misbehavior node?
No
No
Add to Ongoing Session list
Yes
Need to detect misbehavior ?
Yes
Misbehavior Detection
Increment Location Discover Counter
No
Misbehavior Detection Module
Yes
Increment Forward Counter
Add 3C header to the packet
Is initiated by last hop?
MAC Layer (Data Link)
Broadcast location discover packet
Location discover packet arrives
Broadcast location discover packet

**Figure 5.** Update the counter during the Location Discovery phase.

### *3.3.1 Initiate Location Discovery*

When a **Location Discovery procedure** in the routing layer passes an initiated Location Discovery packet to the 3C Module, it processes the packet and updates the ***Location Discover Counter*** as follows (see Figure 5):

1. The 3C Module determines if this Location Discovery packet belongs to one of the initiator's (i.e., the source node's) ongoing communication session in the *Session Table*. If it does not belong to an ongoing session, go to Step 2; otherwise, go to Step 3.
2. The 3C Module increments the ***Location Discovery Counter*** by one and adds it to the *Session Table* (and go to Step 3).
3. The 3C Module adds a 3C header containing the values of the three counters (i.e., ***Forward Request Counter***, ***Forward Counter***, and ***Location Discovery Counter***) to this Location Discovery packet before passing it to the MAC Layer for broadcast to other nodes.

In the connectionless-oriented approach, the destination of a communication session is periodically updated according to the mobility of the destination node. The location of the source node is updated by piggybacking the location information in the data packets. However, a source node sometime needs to re-discover the location of a destination node due to packet losses caused by congestion, mobility, or channel errors. Thus, we differentiate between the initial location discovery and the location discovery that is re-establishing an ongoing communication session.

### *3.3.2 Receive Location Discovery Packet*

When a Location Discovery packet broadcast from a node *m* to any of its one-hop neighbor node *n*, *n*'s MAC Layer passes the packet to its 3C Module for processing the Location Discovery packet and updating the ***Forward Request Counter*** as follow (see Figure 5):

1. The 3C Module determines if *m* is the source node that initiated this Location Discovery packet (i.e., packet's *source_ID = forward_ID*). If so, go to Step 2; otherwise, go to Step 3.
2. If *m* is the source node of this Location Discovery packet, the 3C Module in *n* uses the information in the packet's 3C header to determine if there is a need to start the detection procedure to examine *m*'s behavior. We will discuss when to initiate the misbehavior detection and the procedure for misbehavior detection in Section 3.5 and 3.6, respectively. If node *m* is confirmed to be misbehaving, the 3C Module of node *n* discards the packet (as punishment); otherwise, go to Step 3.
3. Node *n* keeps records of ongoing communication session in its *Session Table*. If the arriving Location Discovery packet's *source_ID* and *destination_ID* match an entry in node *n*'s *Session Table* (e.g., packet's *source_ID + destination_ID = session_ID*), its 3C Module resets the *time to live* (TTL) timer of the corresponding entry. Next, the Location Discovery packet is then passed on to the routing layer (Step 5).
4. If the Location Discovery packet is not belonged to any ongoing session in the *Session Table* (e.g., packet's *source_ID* and *destination_ID ≠ session_ID*), the 3C Module updates the *Session Table* and increases the ***Forward Request Counter*** by one. The 3C Module then passes the Location Discovery packet to the routing layer for further processing (Step 5).
5. Depending on CLA-S routing protocols, node *n* can discard the packet, continue to forward (i.e., pass back down to lower layers), or initiate a reply procedure (i.e., reach the destination). In Figure 5, we show the routing protocol for CLA in the Routing Layer.

### *3.3.3 Forward or Reply Location Discovery Packet*

Depending on the role of a node in a communication session (e.g., forwarding node or destination node), a node can forward the Location Discovery packet, reply to the Location Discovery packet with a Location Reply packet, or discard the Location Discovery packet according to its routing protocol. A Location Reply packet is generated by a node's Routing Layer when a Location Discovery packet arrived at a destination. This destination node needs to reply the source node of the Location Discovery packet. If a node is the destination, its Routing Layer generates a Location Reply packet and passes this reply packet to 3C Module.

When Routing Layer submits a Location Discovery packet or a Location Reply packet to 3C Module, 3C Module processes the packet and updates the ***Forward Counter*** as follows:

1. 3C Module determines if the Location Discovery packet or the Location Reply packet matches an entry in the *Session Table*. To determine if the Location Reply packet matches an entry in the *Session Table*, 3C Module simply reverses the order of *source_ID* and *destination_ID* of this packet. If the packet matches an entry in the *Session Table*, go to Step 2. Else, the packet is discarded because a malicious node can generate dummy packets to increase its ***Forward Counter*** to avoid detection.
2. 3C Module increases the ***Forward Counter*** by one. Then, the Location Discovery packet or the Location Reply packet is passed to MAC Layer.

## 3.4 Counters Update during the Data Forwarding Phase

Once the location of the destination node is determined, the source node can start a communication session. In CLA-S, nodes simply forward data packets without first establishing the link to the next node. Any node that happens to be within the forwarding zone and in the general direction towards the destination node can compete for the "right" to forward data packets. When a source node *s* starts to send the data packet from routing layer to 3C Module, *s*'s 3C Module simply passes the data packet to the MAC layer without updating any counter.

### 3.4.1 Receive Data Packet

When a node *n* receives a data packet, its MAC Layer passes the data packet to its 3C Module. Then, *n*'s 3C Module updates the ***Forward Request Counter*** as follows:

1. 3C Module determines if the data packet corresponds to a communication session in *n*'s *Session Table*. If so, go to Step 2. Else, go to Step 3.
2. *n*'s 3C Module resets the *time to live* (*TTL*) *timer* of the corresponding entry in the *Session Table* and passes the data packet to the routing layer. Depend on different routing protocols, the data packet is either discarded or forwarded.
3. If the data packet is not belonged to any ongoing session in the *Session Table*, the 3C Module updates the *Session Table* and increases the ***Forward Request Counter*** by one. The 3C Module passes the Location Discovery packet to the routing layer for further processing (e.g., discard or forward data packet).

### 3.4.2 Forward Data Packet

Depend on the routing protocol, the data packet is either discarded or forward (see the "Routing Layer" in Figure 5). In connectionless-oriented approach, every node has equal probability of participate in the data forward procedure. If the routing layer decides to forward data packet, it returns a data packet to 3C Module. The 3C Module processes the data packet and updates the ***Forward Counter*** as follows:

1. 3C Module determines if the data packet matches any entry in the *Session Table*. If so, it increases the ***Forward Counter*** by one and passes the data packet to the MAC layer.
2. Else, the data packet is discarded. We discard any packets that are not in the *Session Table* due to the same reason as discussed in Section 3.3.3. A malicious node can generate dummy packets to avoid evoking the Misbehavior Detection procedure.

## 3.5 Initiate Misbehavior Detection

By modifying its own routing protocol, a malicious node can intentionally drop (i.e., discard) packets to save its power. However, in the connectionless-oriented approach, every node has an equal chance to participate in a forwarding process. Thus, 3C Module needs to determine to whether to "**invoke**" the **Misbehavior Detection procedure**. In order to determine if there is a need to invoke the Misbehavior Detection procedure, 3C Module exams the 3C header in the Location Discovery packet and calculates two ratios, ***Forward Ratio*** (***FR***) and ***Request Ratio*** (***RR***) as follow:

- ***Forward Ratio**$_i$* (***FR**$_i$*) = $\dfrac{\textit{Forward Counter}_i}{\textit{Forward Request Counter}_i}$
- ***Request Ratio**$_i$* (***RR**$_i$*) = $\dfrac{\textit{Location Discovery Counter}_i}{\textit{Forward Counter}_i}$

  , where *i* is the node that initiated this Location Discovery packet (i.e., the source node).

When a node *n* receives a Location Discovery packet from a node *m*, *n*'s 3C Module checks if *m* is the initiator (i.e., source node) of this Location Discovery packet using the information included in the packet (see Section 3.3). If *m* is not the initiator, *n*'s 3C Module does not invoke the detection procedure. Then, this Location Discovery packet passes to the Counter Update procedure for further process (see Figure 5). If *m* is the initiator of this Location Discovery packet, *n*'s 3C Module checks the 3C header included in this Location Discovery packet for the following conditions:

1. $\boldsymbol{FR_m} < \boldsymbol{FR_n}$
2. $\boldsymbol{RR_m} > \boldsymbol{RR_n}$ * *Initiate Detection Threshold*

If one of the above condition is true, *n*'s 3C Module broadcasts a 3C packet (including *n*'s 3C information) to its one-hop neighbor nodes. When a node receives *n*'s 3C packet, it replies with its own 3C information. When *n* receives its neighboring nodes' replies, *n* calculates the ***Local Average Forward Ratio*** (***LAFR***). This ratio is calculated as follow:

$$\boldsymbol{LAFR_n} = \frac{\sum_{i=1}^{k}(FR_i) + FR_n}{k+1}$$

, where *k* is number of neighboring nodes for *n* (excluding *m*).

In Vehicular Network, network conditions, such as density and congestion, can change dynamically. Thus, the ***Local Average Forward Ratio**$_n$* (***LAFR**$_n$*) is merely the local network condition around *n*. If $\boldsymbol{FR_m} \geq \boldsymbol{LAFR_n}$, it means that network condition at area of *m* might be congested which causes *m* not forward packets. Thus, we do not need to invoke the Misbehavior Detection procedure. On the other hand, if $\boldsymbol{FR_m} < \boldsymbol{LAFR_n}$, then *m* might be misbehaving by not forwarding packets. In this case, *n* activates its **Detection Mode**. Notice that all the neighboring nodes of *m* and *n* can activate its **Detection Mode** (but not at same time) because their ***Forward Ratios*** are similar. When a node activates its **Detection Mode**, it continues to forward for other nodes except for the suspicious node.

To avoid evoking the Misbehavior Detection procedure, malicious nodes can initiate dummy packets to increase their own ***Forwarding Counter***. Although, by doing so, malicious nodes defeat the purpose of saving power. Nevertheless, 3C Module can prevent this misbehavior act by compare the outgoing packets against the *Session Table*. If the packet does not match any entry in the *Session Table*, 3C Module discards this dummy packet.

## 3.6 Detection Mode

The **Detection Mode** has two states: *Listening*-State and *Detecting*-State. Initially, a node in the Detection Mode is set to *Listening*-State. In the *Listening*-State, a node *n* waits for a random period of time. During this delay period of time, *n* does the following:

1. If *n* hears a Detection packet from another node to test node *m* (i.e., the suspect node), *n* resets the delay time. A Detection packet is generated by **Misbehavior Detection procedure** to test a suspicious node.
2. If *n* hears a Detection packet been forwarded by *m*, *n* exits the **Detection Mode**. By exiting the **Detection Mode**, *n* forwards *m*'s Location Discovery packet. Similarly, all other nodes that are in their Detection Mode (*Listen*-State) hear *m* forwarded the Detection packet will exist their Detection Mode.

At the end of delay period, node *n* enters the *Detecting*-State. In the *Detecting*-State, *n* invokes the **Misbehavior Detection procedure** to determine if *m* is a malicious node.

## 3.7 Misbehavior Detection Procedure

The detection mechanism can be implemented as a software application as proposed in [3] for lower cost. Alternatively, it can also be implemented as a build-in component of the temper resistant module for better security. Without loss the generality, we base our discussion on the latter option.

The purpose of the **Misbehavior Detection procedure** is to detect uncooperative behaviors that result in disruption or degradation of data transmission. We focus on network layer attacks and do not address lower level threats such as physical layer jamming and MAC layer disruptions. The attacks contained by the Misbehavior Detection Module are as follows. First, if there is a suspicion of dropping packets was detected during the location discovery phase, the Misbehavior Detection procedure is invoked. Second, the **Misbehavior Detection procedure** captures malicious users who deliberately discard packets that they are obligated to forward either for selfish purposes or to mount denial of service attacks.

When a node *n* invokes its **Misbehavior Detection procedure** to detect a suspect node *m*, the procedure is as follows:

1. *n* calculates a *virtual link* (see Figure 6) using the location information (i.e., cell ID) contained in *m*'s Location Discovery packet.
2. Based on this *virtual link*, *n* generates a Detection packet (i.e., similar to regular data packet). The source location and destination location of this Detection packet are as follow:
   - Source node's location (*S_cell_ID*) of this Detection packet is the cell behind of *n*, relative to *m*.
   - Destination node's location (*D_cell_ID*) of this Detection packet is the cell behind of *m*, relative to *n*.
3. Next, *n* broadcasts this Detection packet. All the neighboring nodes of *m* are in Detection Mode and will not forward this Detection packet.
4. *n* waits for a *t* period of time (*t* = maximum delay time in the routing layer).
5. At the end of the delay, if *n* does not receive the Detection packet forwarded by *m* (i.e., *forward_ID* = *m*), *n* repeats the process again for two times (total of 3 times).

If *n* receives the detection packet which is forwarded by *m*, *n* (and all the neighboring nodes of *m*) exits the Detection Mode. *n* forwards *m*'s Location Discovery packet because *m* has passed *n*'s Misbehavior Detection procedure. If *n* does not receive the detection packet from *m*, *n* punishes *m* by discard *m*'s Location Discovery packet for period of $t_{punish} = C \times (\boldsymbol{LAFR_n} - \boldsymbol{FR_m})$. Thus, the punishment period is proportion to individual (misbehaving) node's misbehaved level.

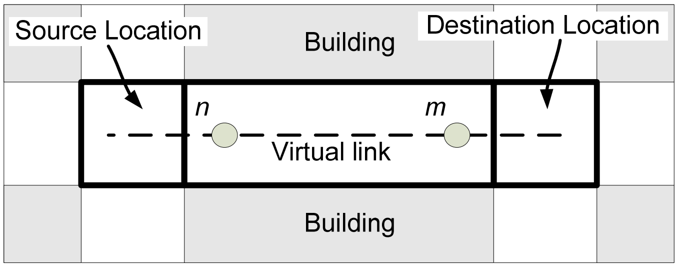


**Figure 6.** Virtual link for a Detection packet.

# 4. EXPERIMENTAL RESULTS

We conducted various experiments to verify the effectiveness of the proposed 3CE (*3-Counter Enforcement*) scheme in enhancing performance of vehicular network. In this section, we first introduce the implemented schemes, simulation setup and parameters. We then study the proposed technique based on various performance metrics.

## 4.1 Schemes Implemented

We implemented three schemes, namely the **reference** scheme, the **defenseless** scheme and the proposed **3CE** scheme, for performance evaluation. In the **reference** scheme, all the nodes act collaboratively and relay data for each other. In the **defenseless** scheme, a certain fraction of nodes are misbehaving as they failed to participate in forwarding procedure. In other wards, these nodes discard any packets not destined at them. No detection or prevention mechanism is implemented so that the network is totally "defenseless". Finally, in the proposed **3CE** scheme, misbehaving nodes are detected and punished. A malicious node can recognize itself is been punished when Location Discovery packets of the node has been dropped four consecutively times. Once malicious nodes recognized themselves been punished, they participate in forwarding data to rejoin the network.

## 4.2 Simulation Setup

All the experiments were conducted using GlomoSim [14]. This simulator, developed at UCLA, is a packet-level simulator specifically designed for ad-hoc networks. It follows the OSI 7-layer network communication model. Although, popular simulators such as NS-2, OPNET Modeler, and GloMoSim provide advanced simulation environments to test and debug network protocols, we prefer GloMoSim due to its ability to handle high mobility of nodes and its scalability of handle large number of nodes and size of network area. Unlike other simulators, GloMoSim uses the parallel discrete-event simulation capability provide by Parsec [1].

Experiments were based on a mobile ad hoc network with 200 nodes. The field configuration is a 2000 by 2000 meters field with a street width of 10 meters and building block size of 100 by 100 meters. All nodes employ 802.11 at the MAC layer. Each node has a radio range of about 375 meters. The nodes move in the directions permitted in the streets. Upon arriving at an intersection, a node probabilistically changes its directions of movement (e.g., turn left, turn right, or continue in the same direction). Traffic applications are constant-bit-rate sessions involving 1/10 of all nodes. Each data packet is 512 bytes. Multiple simulation runs (100 runs per setup on average) with different seed numbers were conducted for each scenario and collected data were averaged over those runs. The total simulation duration for each run was 20 minutes (1200 seconds). We varied the number of misbehaving nodes (i.e., 5%, 10%, 20%, and 30% of total number of nodes) and node mobility (i.e., 10 *m/s* to 25 *m/s* or 22 *mile/hr* to 56 *mile/hr*). The *Initiate Detection Threshold* (*IDT*) is set to 1.2. This threshold determine percentage of a node require to participated in forward procedure in order not to initiate the 3C's detection procedure. For example, when the threshold set to 1.2, a node is allow of 20% of packet drop due to either network condition or mobility. Initially, misbehaving nodes drop all the received packets. Once misbehaving nodes been identified (i.e., all their Location Discovery packets are drop by other neighboring nodes), they behave normally until they are no longer identified as misbehaving nodes (i.e., their Location Discovery packets are forwarded by others).

## 4.3 Metric

In the experiments, we evaluated the proposed scheme based on the following metrics:

1. **Packet delivered ratio (*P*)**: The ratio of the data packets delivered to the destinations and the data packets generated by the CBR source. This measures the rate at which effective data transmission is performed. It is also a good indicator of the degree of collaboration among the nodes.
2. **Misbehaving node detection ratio (*D*)**: The ratio of the number of misbehaving nodes that were correctly identified to the total number of misbehaving node that have actually acted uncooperatively during the simulation.
3. **False accusation ratio (*F*)**: The ratio of the number of 3C Modules that incorrectly accused benign hosts to the overall number of misbehaving nodes that 3C Module identified.
4. **Control overhead ratio (*C*)**: The ratio of the number of routing packets transmitted per distinct data packet delivered to a destination.
5. **End-to-end delay (*D*)**: The number measured in *milliseconds*, includes detecting and processing malicious nodes delay, route discover latency, queuing delays, retransmission delay at the MAC, and propagation and transmission times. This measures the total delay time from a sender to a destination (without communication sessions that belong to misbehaving nodes).
6. **Active detection ratio (*A*)**: The ratio of the number of nodes activated their Detection Mode per misbehaving node's location discovery packet.

## 4.4 Experimental Results

We present the simulation results in this section.

### *4.4.1 Packet Delivered Ratio*

By employing the proposed scheme, significantly more data can be successfully delivered to the destinations since nodes are now required to participating in data forwarding. Figure 7 depicts the practical scenarios where the number of malicious node is 10% and 20% of the total nodes. We observe in the case of fewer malicious nodes (less then 10%), the CLA-S with 3CE (*CLA-S-3C*) has very close throughput to the references CLA (*CLA-S-Reference*). The proposed technique improves the deliver ratio by more than 25% compare to the defenseless scheme.

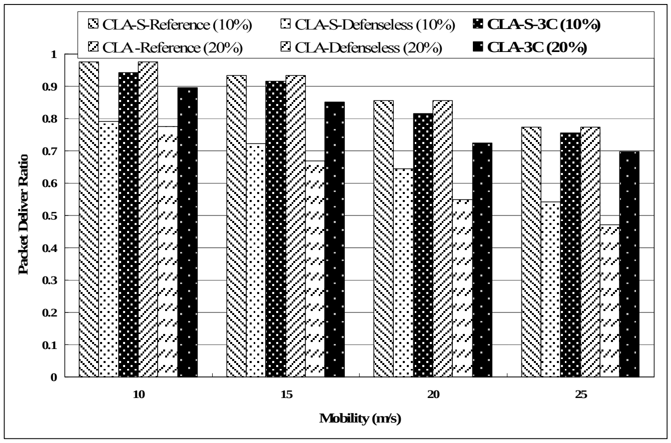


**Figure 7.** Packet Deliver Ratio (*P*) with 10% and 20% Malicious Nodes.

Another important factor to the performance of packet deliver ratio is the speed of mobility. Due to mobility of mobile hosts, addressing frequent and unpredictable topology changes is fundamental to MANET research. As the mobility of node (e.g., speed) increase, the performance of all three schemes (i.e., 3CE, reference, and defenseless) are decreased. Similarly when we increased mobility and number of malicious nodes (see Figure 7), the packet deliver ratio is also decreased as the result. However, consider of mobility increased from 10 *m/s* (or 22 *miles/hour*) to 25 *m/s* (or 56 *miles/hour*), the deliver ratio decreased only 20% in average. Thus, the protocol is still suited for many applications (e.g., video and audio) with error correction code.

### 4.4.2 Misbehaving Node Detection Ratio

We list the results of misbehaving node detection ratio for various simulation scenarios in Table 1. They indicate that the proposed misbehaving node detection mechanism is very effective. In most cases, the 3CE's detection ratio is about 87%. The results demonstrate that on-demand misbehaving node detection is applicable. Since the proposed 3CE technique can adapt by the CLA-S, it is ideal for highly dynamic MANETs such as vehicle-to-vehicle networks.

**Table 1. Detection ratio and False Accusation ration of CLA-S with 3CE.**

| | Detection Ration (D) | | | | False Accusation Ratio (F) | | | |
|---|---|---|---|---|---|---|---|---|
| **Speed (m/s)** | **10** | **15** | **20** | **25** | **10** | **15** | **20** | **25** |
| **5% misbehaving nodes** | 89% | 88% | 83% | 81% | 0% | 2% | 3% | 2% |
| **10% misbehaving nodes** | 93% | 91% | 86% | 88% | 1% | 2% | 2% | 3% |
| **20% misbehaving nodes** | 91% | 85% | 89% | 87% | 1% | 1% | 2% | 2% |
| **30% misbehaving nodes** | 91% | 87% | 84% | 85% | 2% | 2% | 4% | 5% |

### 4.4.3 False Accusation Ratio

We report the false accusation ratios of the proposed 3CE scheme under various scenarios in Table 1. We conclude that in all node mobility scenarios the false accusation ratio is very low. We observe that this ratio is higher when the speed of nodes is increased. This is due to the fact that some of the suspect nodes moved out of the detection node's radio range and were thus incorrectly classified by 3CE's Misbehaving Detection procedure as misbehaving nodes, thereby lifting the false accusation ratio. Nevertheless, further investigation of simulation log files shows that under all simulation configurations, on average less four nodes were incorrectly accused. Both results indicate that the proposed detection mechanism is able to detect most of the in-cooperative nodes with very low false accusation ratio.

### 4.4.4 Control Overhead Ratio

With 20% of malicious nodes, we observe that the Control Overhead Ratio is higher when the speed of nodes is increased (see Figure 8). Similar to False Accusation Ratio, this is due to the fact that some of the suspect nodes moved out of the detection node's radio range and were thus cause some nodes to invoke 3CE's Misbehaving Detection procedure, thereby lifting the Control Overhead Ratio. However, this is inevitable in most on-demand misbehaving node detection approaches.

### 4.4.5 End-to-End Delay

We report the increasing of end-to-end delay in Figure 9. With 20% of malicious nodes, we observe that the proposed scheme incurs minimum end-to-end delay. In most of cases, the length of delay increases approximately six *milliseconds* compared the reference schemes. This can due to the fact that other nodes can continue to forward data packet while one node is detecting a malicious node. Also, malicious nodes are unable to utilize the network resource once they are identified. Since we punish the misbehaving nodes by not forwarding their Location Discovery packet for a period of time, we did not include the communication sessions which the source nodes are misbehaving nodes.

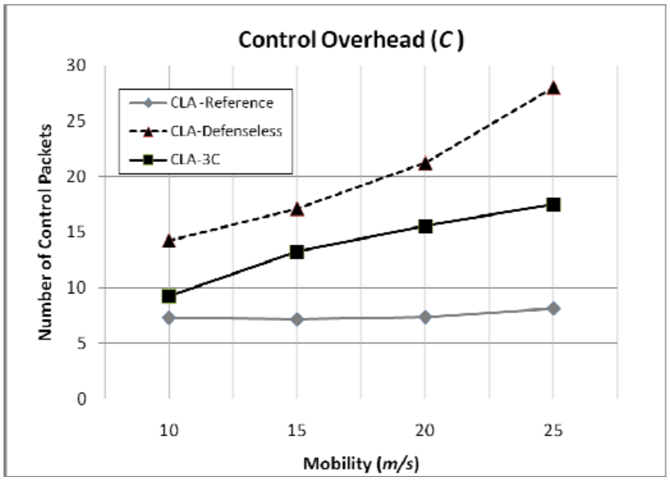


**Figure 8.** Control Overhead (*C*) with 20% Malicious Nodes.

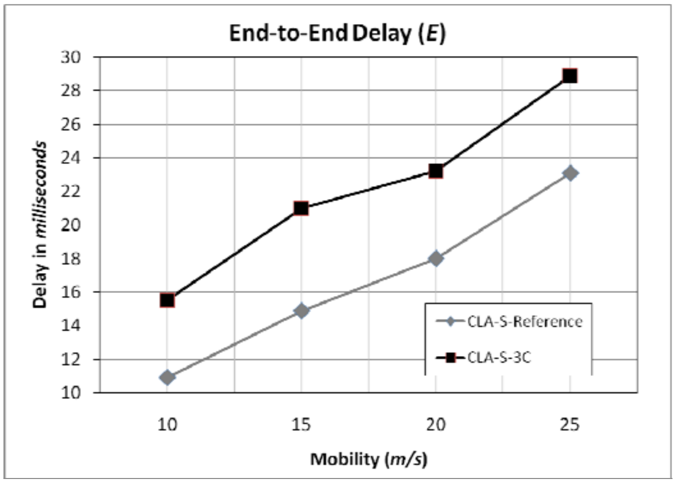


**Figure 9.** End-to-End Delay (*E*) with 20% Malicious Nodes.

### 4.4.6 Active Detection Ratio

With speed of 20 *m/s* and 20% of malicious nodes, we observe that the number of nodes activated Detection Mode per malicious node's location discover packet (that attempt to establish a connection) becomes fixed even the number of nodes in the network increased from 200 nodes to 1400 nodes (see Figure 10). In fact, if we assume a malicious node is stationary and no obstacle (e.g., buildings) within the network, the maximum number of neighboring nodes that are in the Detection Mode (i.e., *Detecting*-State) is six (see Figure 11). If a malicious node is moving at speed of 20 *m/s*, then the moving rang (i.e., a circle with radius of *r*) within the maximum delay time ($t$ = 2 *seconds*) of the Detection Mode is as follow:

$$r = speed * time = 20(m/s)*2(s) = 40(m)$$

With radio range of a node is 375 meters; the radius of circular area of the maximum area of neighboring nodes that can activate Detection Mode is as follow:

$$r_{Detection} = r + \text{radio range} = 40(m) + 375(m) = 415(m)$$

Thus, the maximum number of neighboring nodes that are in the Detection Mode is seven nodes (see Figure 12). In order for a malicious node to move out of area where its neighboring nodes have activated the Detection Mode, the malicious node needs to travel of 790 *meters* (i.e., 415 *m* + 375 *m*). With maximum moving speed of 20 m/s, the time a malicious node to move out of this area is 39.5 seconds (i.e., 790(*m*) / 20 (*m/s*)). Thus, the upper bond of Active Detection Ratio (*A*) is 7 nodes per 39.5 seconds (or 0.18 nodes per second). This confirms with our simulation study. In fact, the result in Figure 10 shows that our approach is able to adapt under high mobility (i.e., variety of applications – vehicular networks) and high density networks (i.e., scalable).

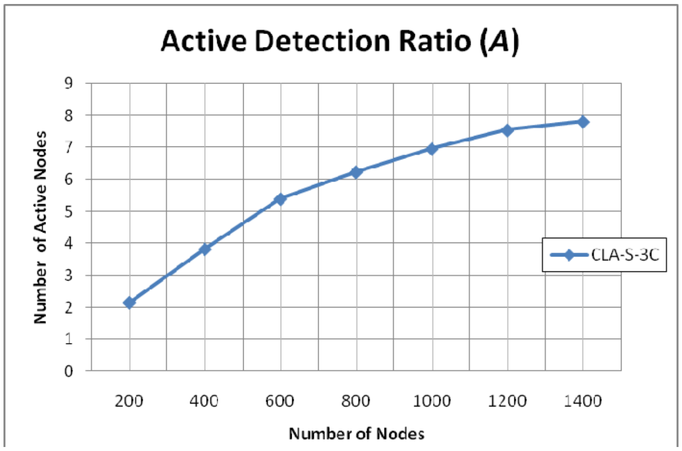


**Figure 10.** Active Detection Ratio (*A*) with 20% Malicious Nodes.

## 5. Conclusion

In this paper, we proposed an efficient 3CE (*3-Counter Enforcement*) scheme to enforce collaboration for CLA-S in vehicular network. Our contributions are as follows. 1) We introduce an on-demand approach to misbehaving-node detection for the CLA-S approach. Since the CLA-S addresses highly dynamic networks (i.e., vehicle-to-vehicle networks), the existing misbehaving-node detection techniques are not suitable. Our approach supports this type of routing protocol under high mobility environments. 2) Each node maintains three counters to represent its own status (i.e., reputation). Since nodes only determine their neighboring nodes' counters information when a location discovery phase, no additional information is needed under a normal operation (i.e., nodes behave normally).

We conducted various experiments to study the effectiveness and efficiency of the proposed 3CE technique. The simulation results indicated that the proposed technique is very effective in enforcing collaboration. The degree of collaboration is significantly strengthened as the network throughput is greatly improved compare to a defenseless network. Such improvement is accomplished with almost no false accusation of cooperative nodes. As of efficiency, the proposed scheme incurs minimum overhead.

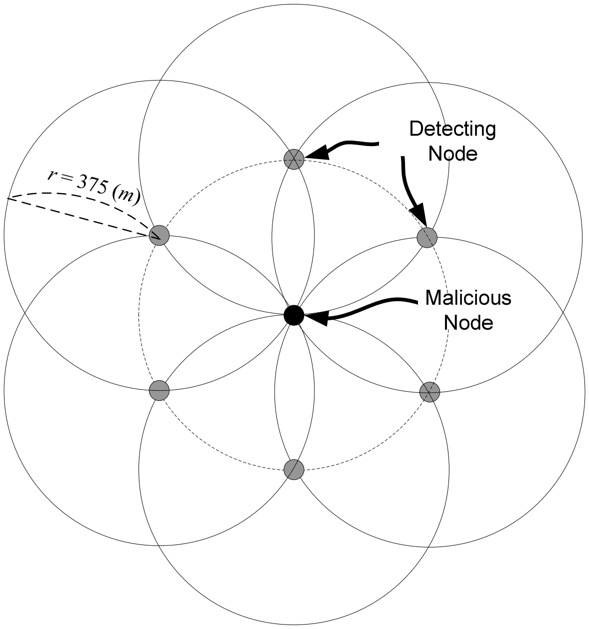


**Figure 11.** Number of detecting nodes needed per malicious node at 0 (*m/s*).

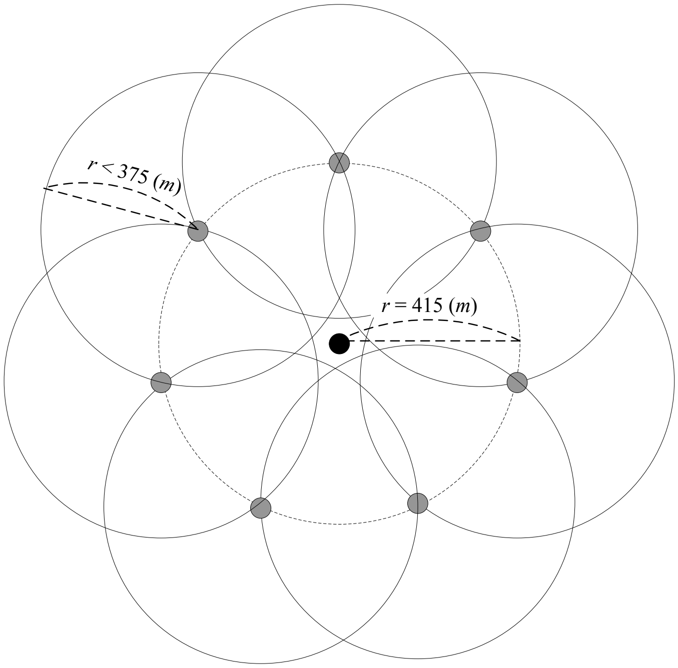


**Figure 12.** Number of detecting nodes needed per malicious node at 20 (*m/s*).

# 6. REFERENCES

[1] R. Bagrodia, R. Meyer, M. Takai, Y. A. Chen, X. Zeng, J. Martin H. Y. Song. Parsec: A Parallel Simulation Environment for Complex Systems. IEEE Computer. Vol 31, Issue 10, pp. 77 – 85. Oct 1998.

[2] S. Buchegger and J. L. Boudec. Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes – Fairness in Dynamic Ad Hoc Networks. In Proceedings of IEEE/ACM MobiHoc, Lausanne, CH, June 2002.

[3] L. Buttyan and J. Hubaux. Enforcing Service Availability in Mobile Ad Hoc WANs. In Proceedings of IEEE/ACM MobiHoc, Boston, MA, USA, August 2000.

[4] H. Fubler, J. Widmer, M. Kasemann, M. Mauve, and H. Hartenstein. Contention-Based Forwarding for Mobile Ad-Hoc Networks, Elsevier's Ad-Hoc Networks, Vol 1, no 4, pp. 351-369, 2003.

[5] A. H. Ho, A. Aved, and K. A. Hua. "A Novel Broadcast Technique for High-Density Ad Hoc Networks," Proceedings of International Wireless Communications and Mobile Computing Conference (IWCMC 2006), Vancouver, Canada. July 3-6, 2006. pp. 425 – 430.

[6] A. H. Ho, Yao H. Ho, and Kien A. Hua. "A Connectionless Approach to Mobile Ad Hoc Network in Street Environments." Proceedings of IEEE Intelligent Vehicles Symposium (IV 2005). Nevada, USA. June 2005.

[7] Y.H. Ho, A.H. Ho, K.A. Hua, and G.L. Hamza-Lup. A Connectionless Approach to Mobile Ad hoc Networks. Proc. of Ninth International Symposium on Computers and Communications (ISCC), Vol 1, pp. 188-195, Alexandria, Egypt, 2004.

[8] Y. H. Ho, Ai Hua Ho, Georgiana L. Hamza-Lup and Kien A. Hua, "Cooperation Enforcement in Vehicular Networks," the IEEE International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ) 2008 (as one of the best papers).

[9] N. Jiang, S. Sheu, K. A. Hua, and O. Ozyer. A Finite-State Model Scheme for Efficient Cooperation Enforcement in Mobile Ad Hoc Netowkrs. In Proceedings 11th International Conference on Parallel and Distributed System, Fukuoka, Japan, 2005.

[10] B. Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In Proc. of MOBICOM '00, page 243-254, Boston, MA, U.S.A., August 2000.

[11] S. Marti, T.J. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In Proceedings of MOBICOM 2000, page 255-265, 2000.

[12] P. Michiardi and R. Molva. CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Network. 6th IFIP Conference on Security Communications and Multimedia (CMS 2002), Portoroz, Slovenia, 2002.

[13] A. Pfitzmann, B. Pfitzmann, M. Schunter, and M. Waidner. Trusting Mobile User Device and Security Modules. In Computer, pp. 61-68. IEEE, February 1997.

[14] X. Zeng, R. Bagrodia, and M. Gerla "GloMoSim: a library for parallel simulation of large-scale wireless network," Proc. of the 12th workshop on Parallel and distributed simulation, pp. 154-161, May 1998, Banff, Alberta, Canada.

# Stabilizing cluster structures in mobile networks for OLSR and WCPD as Basis for Service Discovery

Tom Leclerc*, Adrian Andronache†, Laurent Ciarletta*, Steffen Rothkugel‡

* MADYNES - INRIA Lorraine, Nancy,France. Email: {tom.leclerc, laurent.ciarletta}@loria.fr
† CRP Henri Tudor, Luxembourg. Email: adrian.andronache@tudor.lu
‡University of Luxembourg, Luxembourg. Email: steffen.rothkugel@uni.lu

***Abstract*—Service discovery is one of the most fundamental building blocks of self-organization. While mature approaches exist in the realm of fixed networks, they are not directly applicable in the context of MANETs. We investigate and compare two different protocols as basis for service discovery, namely OLSR and WCPD. OLSR is a proactive routing protocol while WCPD is a path discovery protocol integrating node and link stability criteria. Two conflicting objectives of service discovery are the coverage of service queries together with the required bandwidth. Simulations are performed based on a setting in a city center with human mobility. We show that OLSR outperforms WCPD in terms of coverage. Due to its proactive nature, however, bandwidth consumption is high. WCPD on the other hand is much more bandwidth efficient, but at the cost of lower coverage. Finally, we motivate employing OLSR on top of an overlay topology maintained by WCPD. This fosters stability while reducing overhead and keeping coverage high. As a first step towards a hybrid protocol, we aim at increasing the stability of the communication paths. To do so, an adaptive approach is used, which increases the robustness of the network topology structures.**

***Index Terms*—Mobile Networks, Clustering, Topology Stabilization, Service Discovery**

## I. Introduction

In this paper we consider large Mobile Ad hoc NETworks (MANET) where the wirelessly connected devices communicate without any infrastructure with each other. In order to provide ad hoc networks with useful, user friendly and interesting features service discovery should be provided. Service discovery facilitates resource/data/multimedia sharing or for example ad hoc/situated games, furthermore it permits to take full advantage of the dynamic networks specificities.

The goal of service discovery is mainly to find services provided by other nodes in the network in an automated way and use them by knowing a basic set of information. Initially, service discovery protocols were designed for wired networks and most services were simple services, like for instance printing services. Not every node can or wants to achieve a given service. For example to print, a node doesn't need to be connected directly to the printer. Hence just by using the service provided by the node that is actually connected to the printer is enough to be able to print. In the last years a wide range of services became popular, like music sharing, game services or gateway services providing Internet access. Without infrastructure, as in ad hoc networks, the need to automatically, hence not manually which would be to complicated, discover services, that the network offers is even more crucial than in classical wired networks as no central information is available. Service discovery is even more indispensable for nodes with limited capabilities, which want to use a service without having the capability to host or run it by themselves. In ad hoc networks nodes, and the services they provide, can come and go so that topology changes all the time. These topological changes have to be reflected on the service discovery architecture.

In wired network a service failure is mostly due to a service inherent problem while in ad hoc networks topology causes most of the service failures.

In mobile ad hoc networks, just finding a service that suits best the user's and application's requirements is merely sufficient. In today's service-rich and growing networks, what matters is finding the best service that also optimizes part or all of the following requirements: the hop distance, stability, availability, effectiveness, etc. To enable these requirements an additional requirement which is a topological structure seems imperative.

We consider topology oriented protocols where some nodes have higher responsibilities like for instance relaying, grouping or disseminating messages from other nodes. Taking the topology building techniques from these protocols for service discovery protocols, allows us to have an efficient dissemination of service information and enables us to take advantage of the higher responsibility nodes. The higher responsibility nodes, also called directories in service discovery, store, forward or query service information for other nodes.

This paper is based on the work published at UBICOMM 2008 [1]. We investigate and compare the performances of the two topology conscious protocols OLSR [2] and WCPD [3], in regards to their topology architecture, for service discovery achievements. As the capabilities of the devices in ad hoc networks are always growing but still heterogeneous, from low capacities to very high, we consider a full range of services from simple classical printing services to advanced multimedia services. We present a hybrid approach using OLSR on top of WCPD and, as first step towards it, analyze a mechanism for stabilizing the cluster topology.

## II. Related Work

As stated before, most of the service discovery protocols designed for wired network, like SLP [4], JINI [5], or UPnP [6] do not take into account any topology information.

Several discovery mechanisms can be implemented and mixed in service discovery protocols: active/passive discovery, directory or directory-less discovery. Active discovery means nodes broadcast a request for a service in the network and receive one or more answer from the service provider matching the request. Passive discovery means service providers periodically announce their services to all the nodes in the network. To reduce broadcasting in the network from many nodes, eventually resulting in massive flooding of the network, directory nodes are used. These nodes are elected by the surrounding nodes and are responsible for the electing set of nodes. Once elected, they store service announcements and corresponding service information, handle queries of their ”slaves”, hence reducing considerably the load of the network and the non-directory nodes.

Allia [7] is a peer-to-peer caching based and policy-driven service discovery framework. It removes the leader election problem by enabling every node to be self-sufficient. Every node creates alliances with other nodes and uses local policies for forward and caching decisions. A node knows which nodes are in his alliance, but it does not know in which alliances it is included from other nodes. As Allia does not take into account the network topology it does not fit our previously stated requirements.

Others propose to take partial aspects of the topology into account like in [8] and [9], where both use a multicast topology for the service discovery which is given by the network layer. Unfortunately the use of multicast induces a large number of control messages, which also does not suit our requirements.

The most interesting approaches for our work are the ones that take advantage of network topology to disseminate service information efficiently.

OLSR (Optimized Link State Routing) is well known as an ad hoc routing protocol but it is also a popular choice for service discovery architectures, mainly as an underlying connectivity provider. In [10] and [11] the OLSR protocol is used to encapsulate the service discovery messages. Furthermore in [12] the bordercasting, which is the ”Multipoint Relay (MPR)” mechanism of OLSR, is used to efficiently flood the network.

Another interesting architecture is the Hierarchical OLSR [13] (HOLSR) which actually is not a service discovery protocol, but does address our problem of disseminating information through ad hoc networks efficiently.

The other type of topology we are taking into consideration is the cluster topology. Although in service discovery the cluster topology can be referred as service discovery with directory. The service discovery directories correspond to the clusterheads of the cluster architecture. Directories are elected on various criteria, like for instance node coverage.

A good example is Scalable Service Discovery for MANET [12] which is a distributed central directories discovery architecture. Directories are responsible for caching the service descriptions, advertising their presence to nodes within their vicinity and handling their service requests by checking the local cache or forwarding the query to other directories. The election of the directories is done on the fly and the main election criterion is the node coverage. To exchange the directory profiles they use bloom filters and ”bordercast” (using MPRs) it in the two-hop neighborhood. However since the selection of the directory nodes relies on the node coverage can lead to problems. For example, superfluous elections occur when a nearby coming node traverses the network and obtains a high node coverage at one particular moment, but disconnects because of his mobility shortly after being elected, thus inducing a new election.

## III. Topology protocols

This section briefly describes the protocols, OLSR and WCPD used in our experiments to find a good suited topology for service discovery. We choose to compare OLSR and WCPD because both build well known topology architectures. On one hand OLSR builds a tree topology and on the other hand WCPD builds a star topology.

### A. OLSR

The Optimized Link State Routing Protocol (OLSR) is a well known routing protocol designed for ad hoc networks. It is a proactive protocol; hence it periodically exchanges topology information with other nodes of the network. One-hop neighborhood and two-hop neighborhood are discovered using Hello Messages (similar to the beacon message). The multipoint relay (MPR) nodes are calculated by selecting the smallest one-hop neighborhood set needed to reach every two-hop neighbor node. The topology control information is only forwarded by the nodes which are selected as MPR. Every node possesses then a routing table containing the shortest path to every node of the network. OLSR enables efficient flooding of the network by building a Tree like topology for every node from a source (Figure 1).

### B. WCPD

The Weighted Cluster-based Path Discovery protocol (WCPD) is designed to take advantage of the cluster topology build by Node and Link Weighted Clustering Algorithm (NLWCA) [14] in order to provide reliable path discovery and broadcast mechanisms in mobile ad hoc networks (Figure 2).

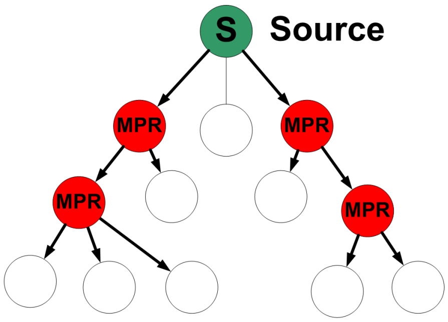


Fig. 1: OLSR topology for one source node in particular.

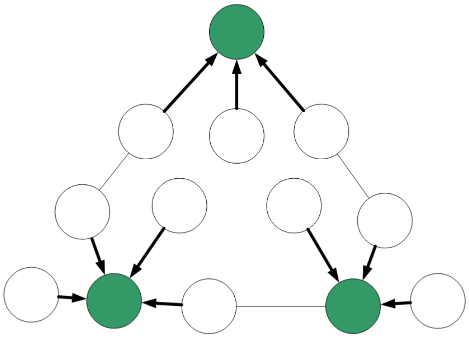


Fig. 2: WCPD cluster topology. The clusterheads are connected by multi-hop paths, which are used for inter-cluster information exchange.

NLWCA organizes ad hoc networks in one hop clusters by using only information available locally. Each device elects exactly one device as its clusterhead, i.e. the neighbor with the highest weight.

The main goal of the algorithm is to avoid superfluous reorganization of the clusters, particularly when clusters cross each other. To achieve this, NLWCA assigns weights to the links between the own node and the network neighbor nodes. This weight is used to keep track of the connection stability to the one-hop network neighbors. When a link weight reaches a given stability threshold the link is considered stable and the device is called stable neighbor device. The clusterhead is elected only from the set of stable neighbors which avoids the re-organization of the topology when two clusters are crossing for a short period of time.

WCPD discovers nearby stable-connected clusters in a pro-active fashion. For the nearby clusterheads discovery algorithm, WCPD uses the beacon, which is a periodically broadcasted message used in ad-hoc networks to detect devices in communication range.

WCPD runs on each network node and requires solely information available locally in the one hop neighborhood. The algorithm uses information provided by NLWCA: the set of stable connected network neighbor nodes and the ID of the own clusterhead. NLWCA also propagates by beacon the own weight and the ID of the current clusterhead. Besides the information provided by NLWCA, the WCPD protocol uses the beacon to disseminate the list of locally discovered nearby connected clusterheads.

By doing so, every node has the following information about each stable one hop neighbor: its clusterhead ID and the ID set of discovered clusterheads and the respective path length. After the data of all stable one hop neighbors is checked, the set of discovered nearby clusterheads and the path length is inserted into the beacon in order to propagate it to the one hop neighborhood.

The WCPD broadcasting algorithm is simple and easy to deploy: the broadcast source node sends the message to the clusterhead, which stores the ID of the message and broadcasts it to the one hop neighborhood. After that, it sends it to all nearby clusterheads by multi-hop unicast and to the own subheads by unicast. The inter-cluster destination nodes repeat the procedure except that the message source clusters are omitted from further forwarding. Additionally the information about the ID of the broadcast messages and their sources is stored for a given period of time to avoid superfluous re-sending of the message.

The protocol sends the broadcast message to nearby clusters connected by stable links in order to disseminate it to the network partition. Nevertheless the message also reaches crossing clusters since the broadcasts are received by all nodes in the one-hop neighborhood of local leaders. This increases the chance that the message reaches a high number of nodes in the mobile network partition.

### C. *Disseminating Messages*

As our comparison relies on the information dissemination of both OLSR and WCPD, we furthermore compare both message dissemination mechanisms. When following the flow of a disseminating message, the topological structures, tree and star, of both protocols are highlighted.

The tree topology of OLSR is pointed out in Figure 3. A message sent from a source traverses the network by being forwarded only from the MPRs calculated by OLSR. As OLSR assures the full coverage of the network with the MPR selection, the messages reaches every node in the network.

The star topology is revealed in WCPD on Figure 4. Here a message from a source S (in this case a slave node) is first sent to its clusterhead B. Clusterhead B then one-hop broadcasts the messages to its slaves and multi-hop unicasts it to the nearby clusterheads A and C. Upon receiving the messages clusterheads A and C start the same procedure; broadcast to the slaves and unicast to nearby clusters (omitting source cluster). Thus every node (clusterheads and slaves) will receive the message. However nodes that are not considered as stable (e.g. fast moving nodes) might not receive the message

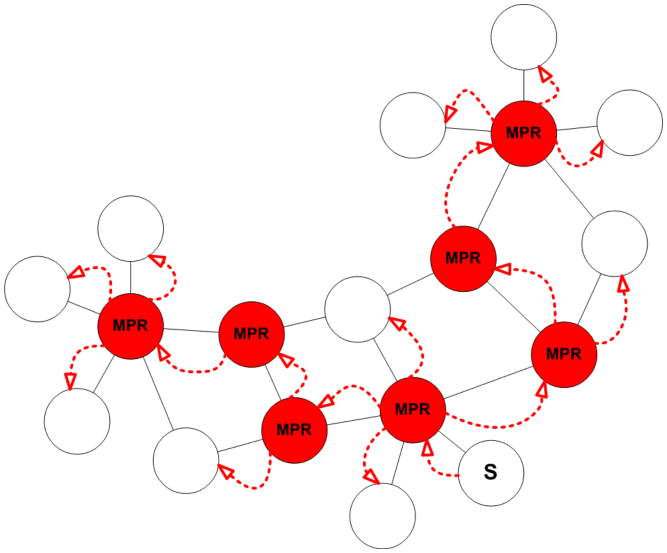


Fig. 3: OLSR message dissemination through a network.

unless they are in the direct neighborhood of a clusterhead that is broadcasting the message (i.e. intended to its slaves).

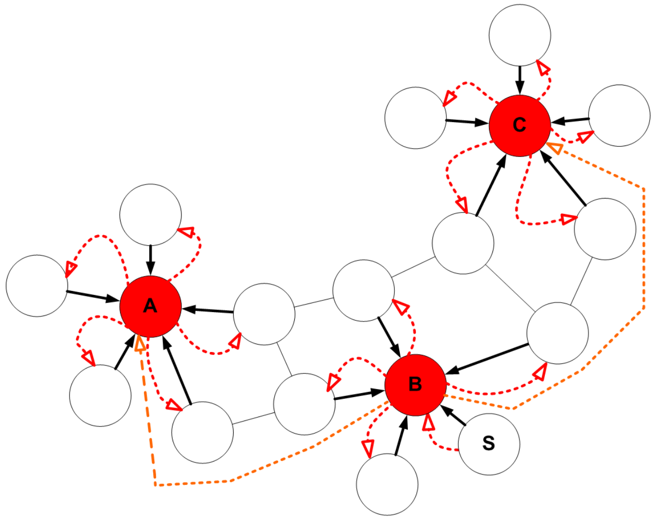


Fig. 4: WCPD message dissemination through a network.

## IV. Experiments

In order to determine the best suited topology for our service discovery protocol, we implemented both protocols on the top of the JANE simulator [15] and performed several experiments.

### A. Simulation settings

For the conducted experiments we used the Restricted Random Way Point mobility model [16], whereby the devices move along defined streets on the map of Luxembourg City for 5 minutes (Figure 5). For each device the speed was randomly varied between [0.5;1.5] units/s. At simulation startup, the devices are positioned at random selected crossroads and the movement to other crossroads is determined by the given random distribution seed. For the experiments a number of 10 different random distribution seeds were used in order to feature results from different topologies and movement setups.

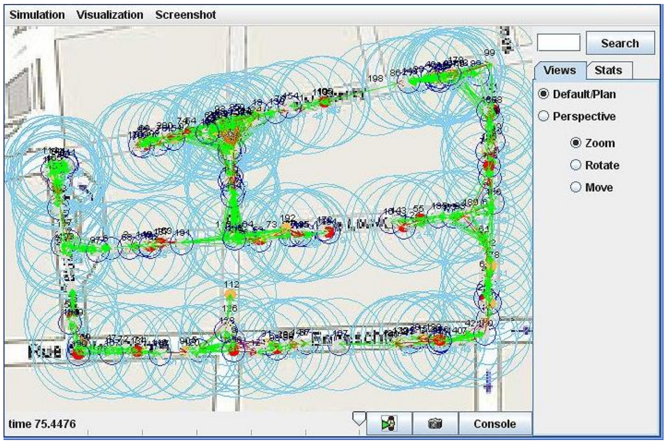


Fig. 5: JANE simulating the protocols on 100 devices. The mobile devices move on the streets of the Luxembourg City map. The devices move with a speed of 0.5 - 1.5 m/s.

For the used mobile environment where nodes move with low speeds between 1.8 and 5.4 km/h the NLWCA link-stability threshold is set on 2. Simulations were done to determine both the used bandwidth in order to build the topologies and the information dissemination performance of broadcasting on top of the two different topologies.

To build the MPR topology, OLSR exchanges the sets of one-hop neighbor nodes with every node in the communication range. Similar to OLSR, WCPD use the beacon to exchange the list of the discovered nearby-clusterheads with the one-hop neighbor nodes. To find out the network load produced during this phase, the size of both the one-hop neighbor sets and the size of discovered clusterheads were tracked every second of the simulation. In order to monitor the information dissemination performance and network load of the broadcasting mechanisms, a node was chosen to broadcast a message every 10 seconds during different simulation runs using different distribution seeds. The number of sent messages (i.e. broadcasts and unicasts) during the dissemination and the number of reached network nodes were tracked.

### B. Results

The results in figures 6, 7 and 8 are illustrating the size of the exchanged node-ID lists at the respective point in the timeline.

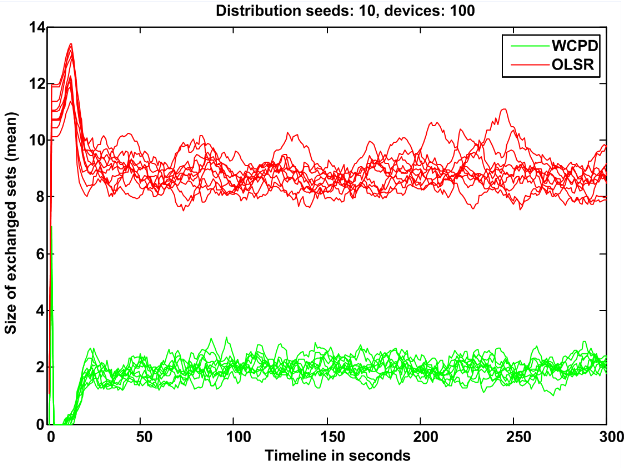


Fig. 6: Size of the sets exchanged per second in order to build the topology.

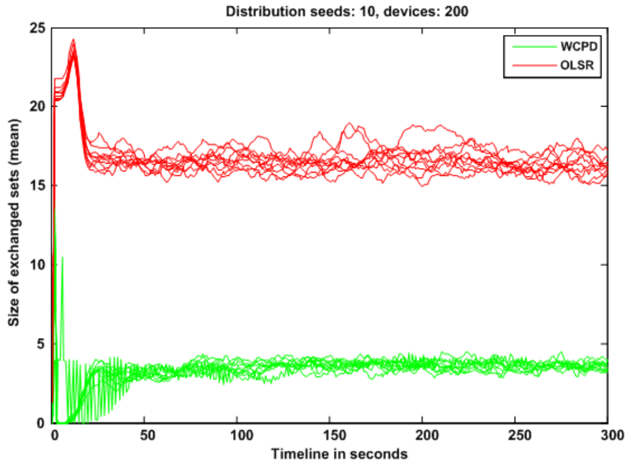


Fig. 7: Size of the sets exchanged per second in order to build the topology.

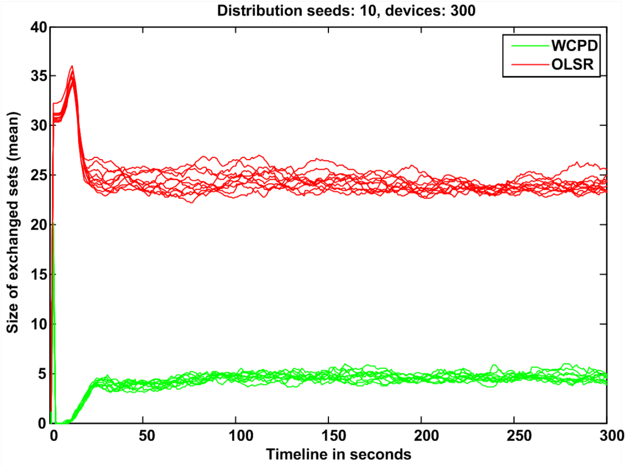


Fig. 8: Size of the sets exchanged per second in order to build the topology.

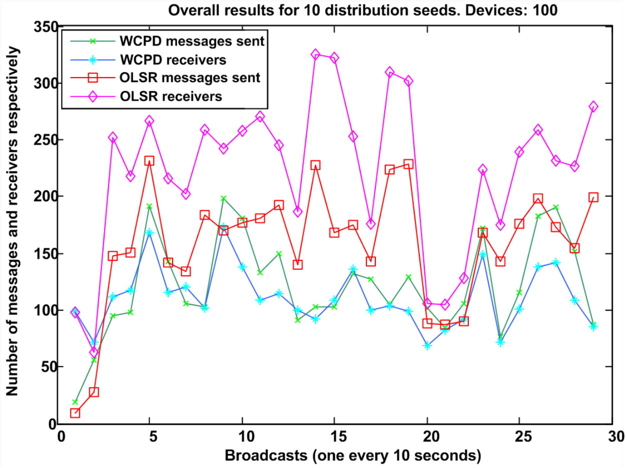


Fig. 9: Overall number of sent messages and node receivers respectively for 100 nodes.

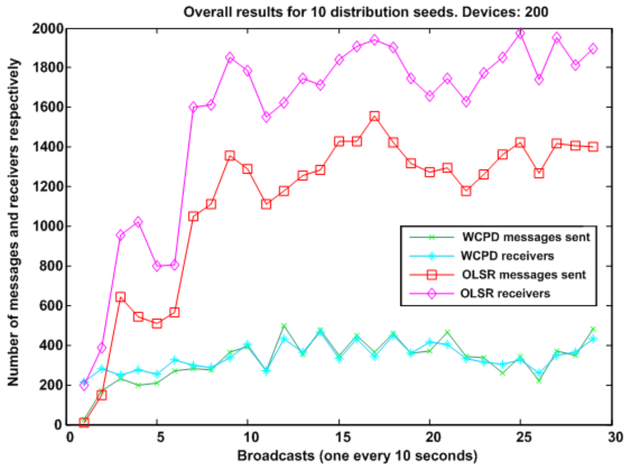


Fig. 10: Overall number of sent messages and node receivers respectively for 200 nodes.

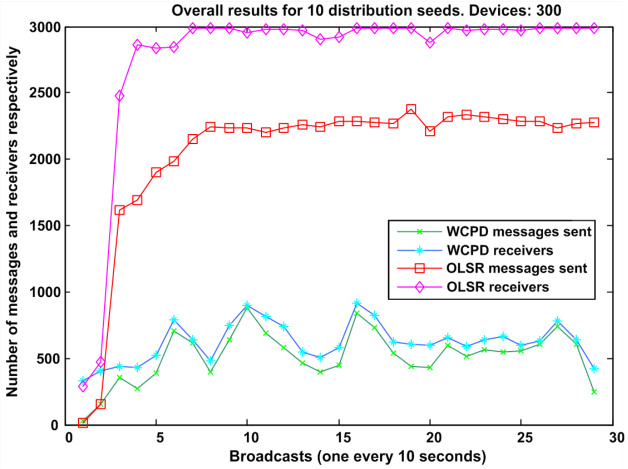


Fig. 11: Overall number of sent messages and node receivers respectively for 300 nodes.

To calculate the bandwidth used by the protocol, one needs to take into consideration the time interval used to periodically send the exchange messages (i.e. hello messages or beacons) and the size of the used node IDs (e.g. 32 bits for IPv4 addresses). This leads to formula 1 for a mean bandwidth B used in an IPv4 network where $|S|$ is the mean number of exchanged addresses and t is the time between the periodically exchanges:

$$B = \frac{|S| \times 32}{t} \text{bits/sec} \qquad (1)$$

The results illustrated in figures 6, 7 and 8 show that OLSR uses a higher bandwidth in both sparser and denser networks. This situation was expected since OLSR is exchanging the set of one-hop neighbors needed for the MPR nodes election.

In contrast to OLSR, WCPD only exchange the set of local discovered nearby clusterhead and sub-heads in order to discover stable paths between clusters in the network vicinity. The NLWCA protocol elects one clusterhead/sub-head in each one-hop neighborhood, which means that the number of clusterheads is a fractional amount of the number of nodes in the network.

The tracking results regarding the message dissemination performance and network load of the broadcasting protocols are presented in figures 9, 10 and 11. The overall results show that the broadcasting on top of the OLSR topology performs much better in terms of message dissemination than on top of the WCPD topology. The denser the network, the higher is the difference between both the number of sent messages and the number of receiver nodes.

## V. A HYBRID APPROACH SOLUTION

OLSR broadcasting is based on flooding the network in an efficient way via the MPRs in such a way that messages reach all nodes already captured. Even in the presence of mobility, the broadcast will arrive at a high number of nodes. In contrast to that, the WCPD approach aims at spreading the messages between topology structures that are considered to be connected in a stable way. Especially in the presence of mobility, the stability threshold might not be reached by all nodes, which might result in a smaller number of broadcast receivers.

We propose to overlay both topologies—in this context for service discovery—by employing the OLSR MPR algorithm on top of the WCPD cluster topology.

In this hybrid approach clusterheads are used as service discovery directories. The discovery of nearby directories in turn is facilitated and maintained by the WCPD protocol. The communication paths between the directories used to exchange service discovery information are maintained by OLSR. Thus, the OLSR protocol has to establish the MPR topology only between clusterheads, which dramatically reduces the required communication load. Additionally, the OLSR topology on top of the cluster topology will result in optimized inter-cluster communication paths.

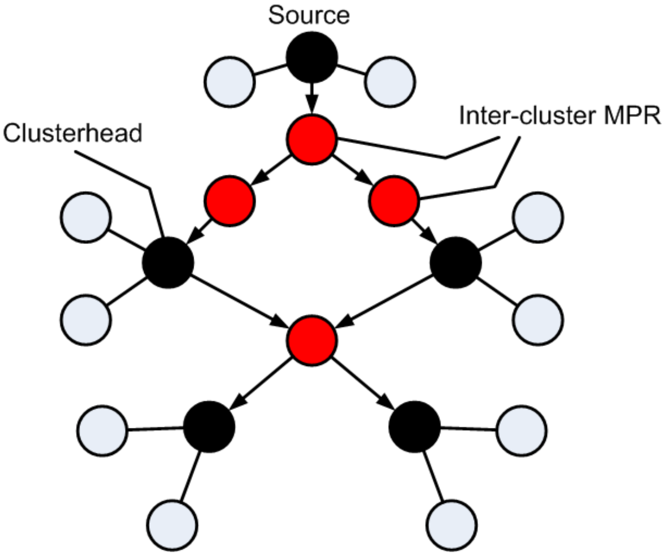


Fig. 12: A hybrid architecture where the OLSR MPR protocol is used to connect nearby clusters discovered by the WCPD protocol.

### A. Stabilizing the cluster topology

The performance of WCPD depends on the stability of the underlying NLWCA topology. In order to increase the reliability of WCPD, the robustness of the cluster topology has to be increased. This section presents the adaptive NLWCA approach, which is a first step towards the hybrid OLSR-WCPD communication protocol.

The NLWCA protocol uses a link stability threshold (LST) in order to decide if a communication link to a neighbor mobile node is stable or not. Simulation results showed that low thresholds are best suited for networks with low mobility. For instance, such networks can be formed by device of users that are in school rooms, cinemas, restaurants, pubs and so on. In such settings a low LST enables the NLWCA to organize fast the local devices, thus reducing the number of elected clusterheads.

In settings where the network nodes are moving around more often and faster, a higher LST is better suited. This allows stable connected clusters to cross each other without to be re-organized by NLWCA. Such networks are created by devices of users for instance at train and subway stations, on the streets of big cities, in shopping malls and so on. The higher LST avoids the organization of crossing nodes but as consequence it increases the number of clusterheads in the mobile network.

The value of the LST has a critical impact on the NLWCA topology. If the LST is to low then the topology is unstable, which means that nodes re-affiliate to new clusterheads very often. This triggers additional network communication and also decreases the robustness of the stable inter-cluster paths. On the other side, a LST that is to high for the given mobility setting leads to election of superfluous clusterheads in the mobile network.

In real mobile environments the network nodes often change their position and the mobility setting. Besides this, scenarios with mixed mobility are common in reality. For instance the nodes in a restaurant have a low mobility and a low LST is best suited. But some nodes in the restaurants that are near to the street might be in communication range of nodes passing by on the street. Thus, these nodes are in a mixed mobility area. Such nodes require a higher LST than the nodes positioned more back in the restaurant in order to avoid superfluous re-affiliations with nodes on the street. This example shows that a constant LST is not the best suited approach for network models with different mobility settings.

To avoid the drawbacks brought by a constant LST, the NLWCA protocol is augmented by a mechanism that allows the change of the LST during runtime. Thus, the protocol is able to adapt the threshold to the given network mobility in order to increase the topology stability.

### B. The adaptive NLWCA protocol

The augmented NLWCA protocol enables each network node to maintain an own link stability threshold. The LST is inserted into the network beacon in order to make it known to the neighbor nodes. When two nodes enter the communication range, the higher of the two LSTs is chosen to be used by both nodes for the link between them. For instance, a node from a high mobility area with a high LST might pass a low mobility area where the nodes have low LSTs. In this case NLWCA uses the high LST of the passing node for the links between it and the nodes in the low mobility area (Figure 13). Thus, a cluster affiliation of the passing node is avoided.

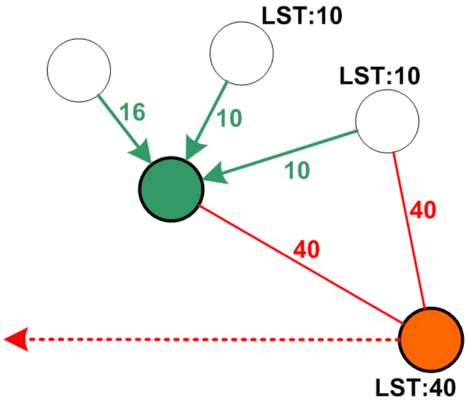


Fig. 13: The higher LST is used by the nodes for the communication links between them. This avoids the cluster affiliation of the node passing by.

The implemented adaptive NLWCA protocol adapts the LST of the nodes by following the listed rules:

1) The LST takes values between 3 and 600 seconds.
2) Monitor the one-hop network neighborhood for a time span of 10 seconds.
3) If stable links are disconnected during the time span then increase the LST by 3.
4) If no link (stable or unstable) is disconnected during the time span then decrease the LST by 1.
5) Updates of a node LST triggers an update of the LSTs of all of its links.
6) Already stable links remain stable even if the LST is increasing. This protects already stable structures from re-organization.
7) Go to rule 2.

Note: All values used might be changed in future work in order to increase the performance of the adaptive mechanism.

## VI. Simulation experiments and results

The goal of the first simulation experiments was to keep track of how NLWCA performs by adapting the LST under mixed mobility settings. In order to do so, a mobility scenario with three mobility settings was created (Figure 14). The first area is a 400 meters long stripe that represents a street. The half of the network nodes used in the simulation randomly move along the street area from one end to the other end with a random speed between 0.5 and 2.5 meters/second. These nodes create the high mobility area of the simulation scenario. Along the street, five areas with low mobility are created. These areas represent restaurants, pubs or other places where people might spend some time. Half of the network nodes are randomly distributed on these areas and they are not leaving the areas for 15 minutes of simulation, thus creating low mobility network settings. The nodes near to the street area are in communication range of the nodes passing by. Thus, these nodes are in a mixed mobility area.

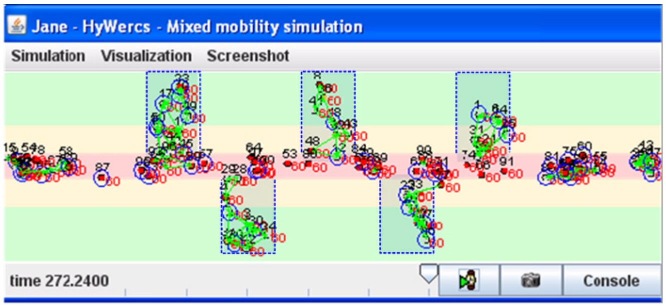


Fig. 14: Simulation scenario with one high mobility area and five low mobility areas.

The sending radius of the nodes was set to 20 meters. In order to compare the adaptive approach with the previous static protocol, the simulation runs were repeated with LST values of 3, 30, 60, 120 and 180 seconds. Each simulation setting was conducted with 10 different distribution seeds. The first sets of experiments were done with a number of 100 mobile nodes.

Figure 15 shows the mean number of elected local leaders during 15 minutes simulation time. A local leader is a clusterhead or a sub-head since both are used by WCPD for inter-cluster path discovery. A low number of local leaders is advantageous since it reduces for instance the backbone communication and the inter-cluster information exchange overhead.

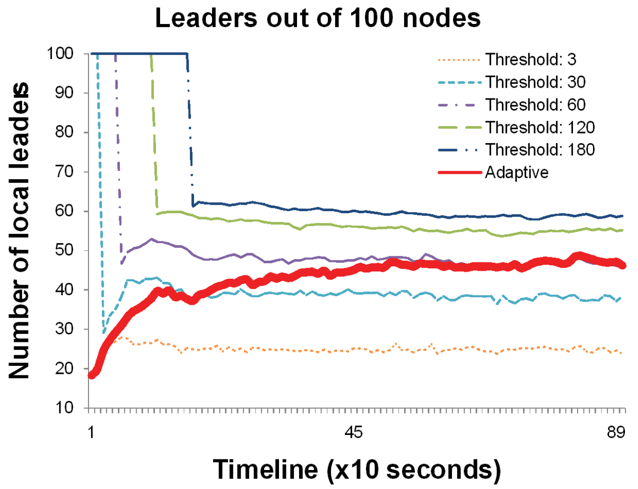


Fig. 15: Number of elected local leaders out of 100 nodes during 15 minutes of simulation.

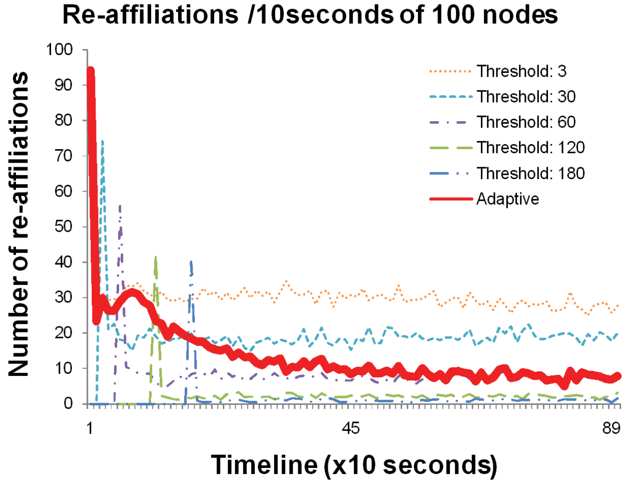


Fig. 16: Number of cluster re-affiliations per 10 seconds during 15 minutes of simulation.

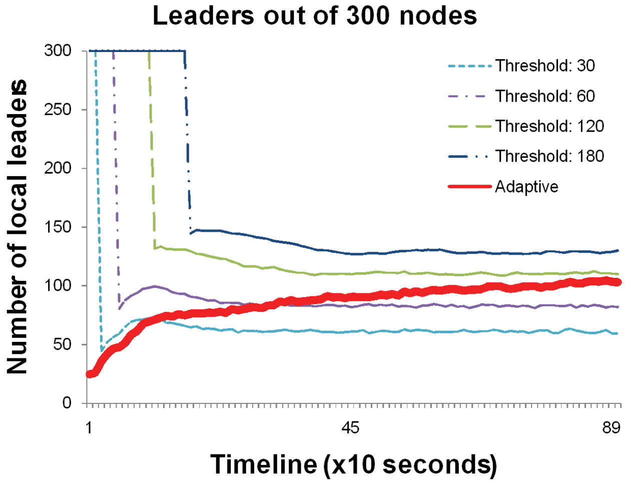


Fig. 17: Number of elected local leaders out of 300 nodes during 15 minutes of simulation.

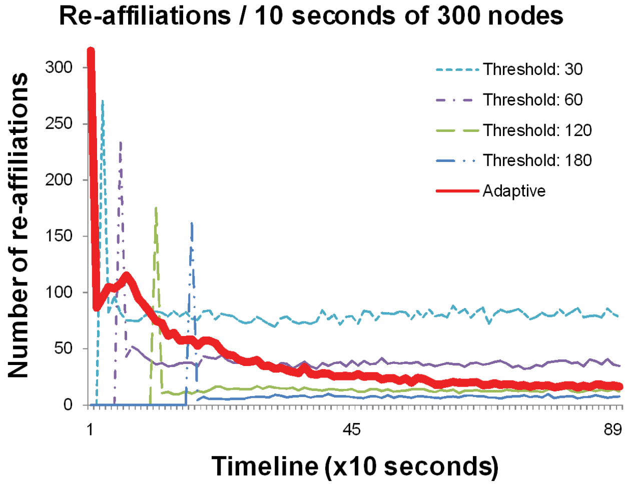


Fig. 18: Number of cluster re-affiliations per 10 seconds during 15 minutes of simulation.

The results in figure 15 show that the lower thresholds lead to a low number of elected local leaders. This is due to the fast organization of the mobile nodes when using a low LST. The drawback of this setting is that crossing clusters are not protected from re-organization. This can be observed in Figure 16, which shows the number of cluster re-affiliations tracked every 10 seconds. A re-affiliation means that a node changed its clusterhead, thus affiliating to another cluster. This induces network communication overhead as well as inter-cluster paths losses or re-configurations.

The lowest LST of 3 seconds triggers a mean value of 30 re-affiliations per 10 seconds compared with 1 re-affiliation per 10 seconds triggered by the 180 seconds LST. This means that the high LST leads to more robust cluster structures. The drawback of the high LST is that it produces a high number of local leaders like Figure 15 shows. Besides this, in low mobility areas such high LSTs are not necessary and lead to a slow cluster organization.

The adaptive NLWCA protocol acts as expected during the simulations. In the beginning, it starts with a low LST, which triggers a high number of re-affiliations by organizing the network in a small number of clusters. Since NLWCA aims to increase the stability of the cluster structures it increases the LST on the nodes in high mobility areas. This leads to a higher number of local leaders (Figure 15) but it highly reduces the number of re-affiliations (Figure 16), thus increasing the robustness of the topology structures.

The same experiments settings were used in simulations with denser networks of 300 mobile nodes. The results are illustrated in Figures 17 and 18.

The behavior of the adaptive protocol in networks with 300 nodes is similar to the one observed in networks with 100 nodes. By increasing the LST of the nodes in high mobility areas, the adaptive NLWCA protocol increases the number of local leaders, thus decreasing the number of re-affiliations.

The results of the conducted adaptive NLWCA simulation experiments are very promising. Nevertheless, experiments with a higher number of network environment scenarios are planned as future work in order to optimize the parameters of the adaptive protocol.

## VII. Conclusion & Future Work

The simulation results show that between the two analyzed approaches, the one based on OLSR is the better choice in order to reach as many nodes as possible by broadcasting for instance service discovery queries. This protocol highly outperforms in terms of broadcast receivers the WCPD approach that fosters the communication between nearby clusters considered to be stable-connected. On the other side, the network load produced by OLSR to build the topology is much higher compared to the one of the WCPD protocol. Besides that, services discovered on nodes in the network vicinity are more valuable than the ones on nodes topologically far away. The multi-hop path to a service host can be easily lost in mobile environments due to the movement of the nodes or network partitioning. In conclusion the OLSR broadcasting approach has the advantage of reaching a much higher number of nodes than WCPD but at the cost of high network overload for the topology maintenance.

In future work we aim to combine the two protocols in a synergetic way by building clusters of stable connected nodes and using the OLSR topology on top of the cluster topology. Thus, a better inter-cluster path discovery and loop-free broadcasting mechanism may be provided at a low network load used for topology maintenance. This will enable the service discovery protocol to take advantage of stable paths to service hosts in the vicinity and at the same time to reach a high number of network nodes by broadcast.

In mobile network environments devices might experience various mobility settings. To increase the stability of the cluster structures NLWCA was augmented to adapt the link stability threshold to the given network mobility. Experiment results show that the middleware successfully reduces the cluster re-affiliations of the mobile devices, thus increasing the robustness of the network structures.

As next step, the hybrid protocol will be deployed and analyzed on top of the robust cluster topology.

## References

[1] T. Leclerc, L. Ciarletta, A. Andronache, and S. Rothkugel, "Olsr and wcpd as basis for service discovery in manets," in *UBICOMM '08: Proceedings of the 2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 184–190.

[2] "Optimized link state routing protocol (olsr), rfc3626," United States, 2003.

[3] A. Andronache and S. Rothkugel, "Hytrace backbone-assisted path discovery in hybrid networks," in *CTRQ '08: Proceedings of the 2008 International Conference on Communication Theory, Reliability, and Quality of Service*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 34–40.

[4] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan, "Service location protocol," 1997.

[5] Jini technology home page. http://www.sun.com/software/jini/.

[6] Universal Plug And Play Forum. http://www.upnp.org/.

[7] e. a. Ratsimor O., "Allia: Alliance-based Service Discovery for Ad-Hoc Environments," in *ACM Mobile Commerce Workshop*, 2002.

[8] S. e. a. Helal, "Konark a service discovery and delivery protocol for ad-hoc networks," 2003.

[9] U. C. Kozat and L. Tassiulas, "Service discovery in mobile ad hoc networks: a field theoretic approach," 2005.

[10] L. Li and L. Lamont, "A lightweight service discovery mechanism for mobile ad hoc pervasive environment using cross-layer design," *Pervasive Computing and Communications Workshops, IEEE International Conference on*, 2005.

[11] e. a. Jose Luis Jodra, "Service discovery mechanism over olsr for mobile ad-hoc networks," *Advanced Information Networking and Applications, International Conference on*, 2006.

[12] F. Sailhan and V. Issarny, "Scalable service discovery for manet," in *PERCOM '05*, Washington, DC, USA, 2005.

[13] L. Villasenor-Gonzalez, Y. Ge, and L. Lament, "Holsr: a hierarchical proactive routing mechanism for mobile ad hoc networks," *Communications Magazine, IEEE*, vol. 43, no. 7, pp. 118–125, July 2005.

[14] A. Andronache and S. Rothkugel, "Nlwca node and link weighted clustering algorithm for backbone-assisted mobile ad hoc networks," in *ICN '08: Proceedings of the Seventh International Conference on Networking*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 460–467.

[15] D. Gorgen, H. Frey, and C. Hiedels, "Jane-the java ad hoc network development environment," *Simulation Symposium, Annual*, 2007.

[16] L. Blažević, S. Giordano, and J.-Y. Le Boudec, "Self organized terminode routing," *Cluster Computing*, vol. 5, no. 2, pp. 205–218, 2002.