International Journal on Advances in Networks and Services

IARIA

2010 vol. 3 nr. 3&4

The *International Journal on Advances in Networks and Services* is published by IARIA.
ISSN: 1942-2644
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Networks and Services
Volume 3, Number 3 & 4, 2010

Internet and Web Services

- Thomas Michael Bohnert, SAP Research, Switzerland
- Serge Chaumette, LaBRI, University Bordeaux 1, France
- Dickson K.W. Chiu, Dickson Computer Systems, Hong Kong
- Matthias Ehmann, University of Bayreuth, Germany
- Christian Emig, University of Karlsruhe, Germany
- Geoffrey Fox, Indiana University, USA
- Mario Freire, University of Beira Interior, Portugal
- Thomas Y Kwok, IBM T.J. Watson Research Center, USA
- Zoubir Mammeri, IRIT – Toulouse, France
- Bertrand Mathieu, Orange-ftgroup, France
- Mihhail Matskin, NTNU, Norway
- Guadalupe Ortiz Bellot, University of Extremadura Spain
- Dumitru Roman, STI, Austria
- Monika Solanki, Imperial College London, UK
- Vladimir Stantchev, Berlin Institute of Technology, Germany
- Pierre F. Tiako, Langston University, USA
- Weiliang Zhao, Macquarie University, Australia

Wireless and Mobile Communications

- Habib M. Ammari, Hofstra University - Hempstead, USA
- Thomas Michael Bohnert, SAP Research, Switzerland
- David Boyle, University of Limerick, Ireland
- Xiang Gui, Massey University-Palmerston North, New Zealand
- Qilian Liang, University of Texas at Arlington, USA
- Yves Louet, SUPELEC, France
- David Lozano, Telefonica Investigacion y Desarrollo (R&D), Spain
- D. Manivannan (Mani), University of Kentucky - Lexington, USA
- Jyrki Penttinen, Nokia Siemens Networks - Madrid, Spain / Helsinki University of Technology, Finland
- Radu Stoleru, Texas A&M University, USA
- Jose Villalon, University of Castilla La Mancha, Spain
- Natalija Vlajic, York University, Canada
- Xinbing Wang, Shanghai Jiaotong University, China
- Qishi Wu, University of Memphis, USA
- Ossama Younis, Telcordia Technologies, USA

Sensors

- Saied Abedi, Fujitsu Laboratories of Europe LTD. (FLE)-Middlesex, UK
- Habib M. Ammari, Hofstra University, USA
- Steven Corroy, University of Aachen, Germany

- Zhen Liu, Nokia Research – Palo Alto, USA
- Winston KG Seah, Institute for Infocomm Research (Member of A*STAR), Singapore
- Peter Soreanu, Braude College of Engineering - Karmiel, Israel
- Masashi Sugano, Osaka Prefecture University, Japan
- Athanasios Vasilakos, University of Western Macedonia, Greece
- You-Chiun Wang, National Chiao-Tung University, Taiwan
- Hongyi Wu, University of Louisiana at Lafayette, USA
- Dongfang Yang, National Research Council Canada – London, Canada

Underwater Technologies

- Miguel Ardid Ramirez, Polytechnic University of Valencia, Spain
- Fernando Boronat, Integrated Management Coastal Research Institute, Spain
- Mari Carmen Domingo, Technical University of Catalonia - Barcelona, Spain
- Jens Martin Hovem, Norwegian University of Science and Technology, Norway

Energy Optimization

- Huei-Wen Ferng, National Taiwan University of Science and Technology - Taipei, Taiwan
- Qilian Liang, University of Texas at Arlington, USA
- Weifa Liang, Australian National University-Canberra, Australia
- Min Song, Old Dominion University, USA

Mesh Networks

- Habib M. Ammari, Hofstra University, USA
- Stefano Avallone, University of Napoli, Italy
- Mathilde Benveniste, Wireless Systems Research/En-aerion, USA
- Andreas J Kassler, Karlstad University, Sweden
- Ilker Korkmaz, Izmir University of Economics, Turkey //editor assistant//

Centric Technologies

- Kong Cheng, Telcordia Research, USA
- Vitaly Klyuev, University of Aizu, Japan
- Arun Kumar, IBM, India
- Juong-Sik Lee, Nokia Research Center, USA
- Josef Noll, ConnectedLife@UNIK / UiO- Kjeller, Norway
- Willy Picard, The Poznan University of Economics, Poland
- Roman Y. Shtykh, Waseda University, Japan
- Weilian Su, Naval Postgraduate School - Monterey, USA

Multimedia

- Laszlo Boszormenyi, Klagenfurt University, Austria
- Dumitru Dan Burdescu, University of Craiova, Romania
- Noel Crespi, Institut TELECOM SudParis-Evry, France

- Mislav Grgic, University of Zagreb, Croatia
- Hermann Hellwagner, Klagenfurt University, Austria
- Polychronis Koutsakis, McMaster University, Canada
- Atsushi Koike, KDDI R&D Labs, Japan
- Chung-Sheng Li, IBM Thomas J. Watson Research Center, USA
- Parag S. Mogre, Technische Universitat Darmstadt, Germany
- Eric Pardede, La Trobe University, Australia
- Justin Zhan, Carnegie Mellon University, USA

International Journal on Advances in Networks and Services
Volume 3, Numbers 3 & 4, 2010

CONTENTS

Live Geography: Interoperable Geo-Sensor Webs Facilitating the Vision of Digital Earth

Bernd Resch [1,2]
Research Scientist
bernd.resch@researchstudio.at

Thomas Blaschke [1]
Head of Department
thomas.blaschke@sbg.ac.at

Manfred Mittlboeck [1]
Key Researcher
manfred.mittlboeck@researchstudio.at

[1] Research Studios Austria
studio iSPACE
Leopoldskronstrasse 30
5020 Salzburg, Austria

[2] MIT
SENSEable City Lab
77 Massachusetts Avenue
building 10, room 400
Cambridge, MA 02139, USA

Abstract – **In the last decade, rapidly declining sensor costs and intense research in sensor technologies lead to the deployment of a number of sensor networks. However, most of these sensor networks are monolithic stovepipe-like systems causing limited interoperability and reusability of both data and workflow components. We present a Live Geography approach which integrates real-time measurement data in a fully standardised infrastructure and couples it with Complex Event Processing (CEP). We demonstrate the interoperability of this geo-sensor web approach and the resulting high degree of flexibility and portability beyond single monitoring applications generally and for five concrete real-world implementations in different application fields. We prove that the Live Geography approach allows for reacting to observed changes through sophisticated embedded processing based on OGC standards such as Sensor Observation Service (SOS) and Sensor Alert Service (SAS). Finally, we discuss how this approach contributes to the vision of Digital Earth as described by Al Gore in 1998 and how it contributes to monitor continuously the status of the environment, of urban infrastructure and the location and health conditions of persons to support an understanding of dynamic processes, to enhance prediction of developments, and to serve Spatial Decision Support Systems.**

Keywords – Live Geography; Standardised Geo-sensors; Embedded Sensor Webs; OGC Sensor Web Enablement; Interoperable Monitoring Systems; Digital Earth.

I. INTRODUCTION

Monitoring single environmental parameters is established in many fields such as measuring water levels, precipitation, air quality, or traffic volume, and plenty more. Especially in urban areas the need for monitoring capabilities is increasing both from a technical side in regard to urban management, infrastructure planning and development capabilities, as well as from a more citizen-centered perspective aiming to support health applications, quality of life, or „geodemographics". The latter term shall be a synonym for analysing the „Where" of people, groups and populations based on tight coupling of absolute locations of individuals, relative movements and – if available – further parameters of the individuals.

The ability to monitor the behaviour of people is still very limited for most city administrations. A mayor or a responsible security manager is usually not able to state how many people are at a certain place at a certain time. One department may know the number of vehicles travelled at an inbound route over the last hour, another information system may report on air quality. Existing demographics may reveal where people sleep or work but do not directly tell how many persons may be present at a certain central place in a city, e.g., at 10am on a Monday morning. Surveillance cameras may provide a first picture at critical locations, but are typically not meant to be quantitatively exploited or in regard to spatial movement patterns.

In other words, integrated monitoring capabilities are critical in cities to ensure public safety including the state of the national infrastructure, to set up continuous information services, and to provide input for spatial decision support systems [1].

However, establishing an overarching monitoring system is not trivial. Up to now, different authorities with heterogeneous interests each implemented their own monolithic sensor systems to achieve specific goals. For instance, regional governments measure water levels for flood water prediction, while local governments monitor air quality to dynamically adapt traffic conditions, and energy providers assess water flow in order to estimate energy potentials. However, these data are mostly not combinable due to different data formats, proprietary protocols or closed-off data access.

This restricts automated workflows and machine-to-machine communication, and prohibits the achievement of the long-term vision of a „digital skin for the Earth" [2], comprised of innumerable heterogeneous sensors, discoverable and accessible over the internet.

In this regard, it is interesting to read how the future of the year 2010 was predicted in 1999: "Ten years from now, there will be trillions of such telemetric systems, each with a microprocessor brain and a radio. Consultant Ernst & Young predicts that by 2010, there will be 10,000 telemetric devices for every human being on the planet. They'll be in constant contact with one another" [2]. This optimistic view may have been inspired by the famous speech of the American Vice-President Al Gore in 1998 [3].

"Digital Earth" was and still is a vision of a multi-resolution, three-dimensional representation of the planet that would make it possible to find, visualise, and make sense of vast amounts of geo-referenced information on the physical and social environment. Such a system would allow users to navigate through space and time, access to historical data as well as future predictions based for example on environmental models, and support access and use by scientists, policy-makers, and children alike [3].

At this time, this vision of Digital Earth seemed almost impossible to achieve given the requirements it implied about access to computer processing cycles, broadband internet, interoperability of systems, and above all data organisation, storage, and retrieval [4].

Generally speaking, the integration of inhomogeneous data poses great challenges, e.g., regarding multi source and heterogeneous, multi-disciplinary, multi-temporal, multi-resolution, and multi-media, multi-lingual information. It is more and more believed that interoperability is key to a success of ubiquitous monitoring. This requires data pre-processing following strict and rigid rules in monolithic sensor systems, in order to fit the specific non-recurring interfaces of the analysis system. Such analysis systems mostly analyse data in a closed black-box model, and usually provide data in a singular and application-tailored format preventing open use and re-use of processed data. When these systems are deployed in an isolated and uncoordinated way automatic assembly and analysis of these diverse data streams is impossible. However, making use of all available data sources is a prerequisite for holistic and successful monitoring for broad decision support using pervasive measurement systems. Thus, recent research increasingly addresses standardised interoperable sensor devices enabling the establishment of portable domain-independent sensing infrastructures [5], [6], [7].

This vision of fully integrated and interoperable sensing workflows fosters awareness for the benefits of open measurement systems. This is especially important for critical monitoring tasks such as emergency management, environmental monitoring or real-time traffic planning, which are not only relevant to the sensor network operators, but also for the city management and for the citizens.

This paper presents the Live Geography approach, which proposes a fully standards-based distributed infrastructure combining current sensor data with Complex Event Processing (CEP) mechanisms, alerting and server-based analysis systems for a wide range of monitoring applications [8]. This approach's main contribution is the creation of a generic standardised sensing and analysis infrastructure, which can be applied to a variety of end applications. This paper illustrates how the developed technical infrastructure can be applied in a broad range of application contexts. The architecture itself and its performance are described in more detail in [8] and in [9], respectively.

This paper is structured as follows. After this introduction, Section II presents related work in the field of distributed sensing infrastructures; Sections III and V describe the Live Geography approach and its deployment in various heterogeneous application areas; Section IV illustrates the challenges and our specific implementation of geo-sensor webs, while Section VI contains a short conclusion.

II. Related Work

The Oklahoma City Micronet [10] is a network of 40 automated environmental monitoring stations across the Oklahoma City metropolitan area. The network consists of 4 Oklahoma Mesonet stations and 36 sites mounted on traffic signals. At each traffic signal site, atmospheric conditions are measured and transmitted every minute to a central facility. One major shortcoming of the system is that it is a highly specialised implementation not using open standards or aiming at portability. The same applies to CORIE [11], which is a pilot environmental observation and forecasting system (EOFS) for the Columbia River. It integrates a real-time sensor network, a data management system and advanced numerical models.

Another sensing infrastructure named *CitySense* is described by Murty et al. [12]. The CitySense project uses an urban sensor network to measure environmental parameters and is thus the data source for further data analysis. The project focuses on the development of a city-wide sensing system using an optimised network infrastructure.

King's College London designed an urban sensor network for air quality monitoring. The London Air Quality Network (LAQN) [13] currently consists of about 150 monitoring sites being a very promising approach to real-time monitoring as it also offers on-the-fly creation of statistic graphs, time series diagrams and wind plots. However, the network does not make use of open standards as a whole, meaning that it is built up in a closed system, although sensor data are accessible over the internet and despite the fact that this solution has a great local significance, but limiting trans-regional inter-linkage with other similar approaches.

One more example is the Networked Soil CO2 Sensing Systems developed by UCLA with the objective to examine the spatial and temporal heterogeneity of a soil environment within a forest area in the James Reserve. The soil environmental measurements are collected with ten stations, each of which consists of an array of belowground sensors including soil CO2, soil temperature, soil water content, and aboveground air temperature, relative humidity, and photosynthetic active radiation. Models are used that relate the aboveground microclimate and the soil measurements to belowground measurements made by the project's sensors to „map" the microclimate in a fine-grained resolution, and investigate soil CO2 fluxes depending on the local characteristics of the forest cover story [14].

Volcano activity observation of the volcano Reventador in Ecuador in 2005–2008 is technically interesting with regard to the remoteness and inaccessibility of the area [15]. Two US Universities have collaborated for several sensor network deployments in the remote, inaccessible area at the active volcano. The objective of the sensor network was to test the ability to detect and measure tremor events of the volcano. The geo-sensor was deployed over a linear centrifugal stretch of 3Km network consisting of seismo-acoustic sensors. The sensor nodes used short-range, battery-preserving wireless multi-hop communication to communicate with each other and relay data, and the sensor network was connected via a long-distance radio communication link to a Freewave radio modem powered a solar-panel powered car battery at a make-shift observatory. The goal of the sensor network deployments was to detect and measure tremor events saved batteries lifespan. The nodes were programmed to compare a short-term average with a long-term average based on locally stored samples. If the difference was bigger than a threshold, a node would send a message to the base station. If a sufficient number of nodes reported an event, the base station triggered a data collection request to all nodes in the sensor network.

Among various examples for mobile geo-sensor networks consisting of individual sensor nodes that are mobile or attached to mobile objects one convincing example is the management of ocean buoys [16].

More recently, the Martha's Vineyard Coastal Observatory (MVCO), owned and operated by the Woods Hole Oceanographic Institution (WHOI), provided the test bed for the first part of the Q2O project, returning the GetCapabilities, DescribeSensor and GetObservation responses for real time offerings of waves every twenty minutes. Wave parameters are computed using an acoustic Doppler current meter, deployed at the 12m isobath continuously measuring pressure and horizontal velocity at 2 Hz. SensorML instances and SOS offerings were developed, describing the sensor characteristics, system provenance and lineage, and the computation of the derived wave height parameters. Quality control tests recommended by the Waves Team of QARTOD were implemented and reported through the SWE offerings [17].

Most of these examples – and many others - exhibit pioneering efforts and contributed significantly to the development of Geo-Sensor Webs. However, common shortcomings of the approaches described above and other related efforts are that the system architectures are at best partly based on open (geospatial) standards, and thus limit interoperability of data and services. Such systems are not able to tackle the challenges of numerous sensors which are built in masses today to observe the Earth surface, atmosphere, solid Earth, and the ocean in different dimensions. At a global level efforts to overcome these challenges are increasingly channelled by the Global Earth Observation (GEO) within the developing Global Earth Observation System of Systems (GEOSS) [5], [6].

Sensor derived information is not only been produced at various locations, with various accuracies, different timely and spatial acquisition patterns but also archived at widely distributed locations. The Live Geography approach employs the „Digital Earth" vision for concatenating sensor information and other geospatial information which is also widely collected and archived with the aim to providing information se**r**vices and ultimately solving challenging environmental and societal issues beyond single application domains. The Live Geography approach seeks to fully utilise all available information resources and to apply them „intelligently". In the following section we will lay out how we ensure the information is being gathered, processed and distributed in a fully interoperable way through open, community-consensus standards among current users and how to process information on demand in order to use non-expert users.

III. LIVE GEOGRAPHY APPROACH

Utilisation of real-time data in GIS applications requires a rethinking of existing practices. The authors even believe that the next generation of GIS will be driven by process models in a sense that users' requests trigger algorithms and heuristics to perform specific services. An „on demand" connection to information networks and an „intelligent" harvesting of existing information in combination with real-time or near real-time data will be key in such a service-centred architecture. This may also require further advances in space-time data models ultimately contextualising Hägerstrand's vision of a time geography [18]. *The time-space path*, devised by Hägerstrand, shows the movement of an individual in the spatial-temporal environment with the constraints placed on the individual by these two factors. Only for the last few years, we are able to utilise location information from GPS, mobile phones or indoor navigation systems to make such concepts operational. Even more recently, researchers study the behavior of groups or larger populations of cities or accessibility aspect based on the space-time model [19].

At present, we may diagnose that Geographic Information Systems begin to evolve from „classic" geospatial data analysis to more „on demand" analysis. The GIS workflow established in the 1960ies and 1970ies may be deliberately characterised that analysis is performed in costly specialised software involving a high degree of manual intervention for data gathering, pre-processing and quality assurance. Furthermore, geospatial analysis has in a vast majority of cases been applied to „not up-to-date" data (at the very time of the analysis) with typically long cycles from data generation to analysis output and a real-world impact. While custodial GIS may predominantly still be associated with this style of information processing research in Geographic Information Science has paved the road towards „information harvesting" on demand in spatial data infrastructures (see various publication in the recent issues of „International Journal of Digital Earth" or „International Journal of Spatial Data Infrastructures Research").

Generally speaking, sensor webs have only emerged very recently because of increasingly reliable communication technologies, affordable embedded devices and growing importance of sensor data for (near) real-time decision

support (see discussion in Section V). They monitor phenomena in Geographic space [20].

The criteria for sensor webs are threefold. The first characteristic is *interoperability*, which means that different types of sensors should be able to communicate with each other and produce a common output. The requirement of *scalability* implies that new sensors can be easily added to an existing topology without necessitating aggravating changes in the present hardware and software infrastructure. Finally, *intelligence* means that the sensors are able to „think" autonomously to a certain degree, which could for instance result in a data processing ability in order to only send required data.

In a recent overview on Geo-Sensor Webs [21] three major trends are identified: the first trend is the currently more readily available technology of seemingly ubiquitous wireless communication networks, including access in remote and inaccessible areas without a wired communication infrastructure and often without even power lines. Furthermore, significant progress has been made in the development of low-power, short-range radio-based communication networks, which augment existing long-distance wireless communication networks. Second, the miniaturisation of computing and storage platforms has led to low power consumption and has enabled novel computational platforms that can run on battery power for extended periods of time (e.g., several months with today's technology). The third major trend is the development of novel sensors and sensor materials; this includes improved and size-reduced traditional sensors as well as the development of novel micro-scale sensors and sensor materials. For example, novel bio-chemical sensors may be used in the marine sciences or air pollution monitoring, or highly sensitive vibration and sound sensors have been applied for volcano monitoring.

Operational real-world sensor network applications are still rare and the majority still serves a single purpose, which limits broader usage of measurement data. This section presents the Live Geography approach. It proposes a flexible and portable measurement infrastructure enabling a wide variety of real-time and near real-time monitoring applications. The system makes extensive use of open (geospatial) standards throughout the entire process chain – from sensor data integration to analysis, Complex Event Processing (CEP), alerting, and finally information visualisation. The basic architecture for such applications is illustrated in Fig. 1.

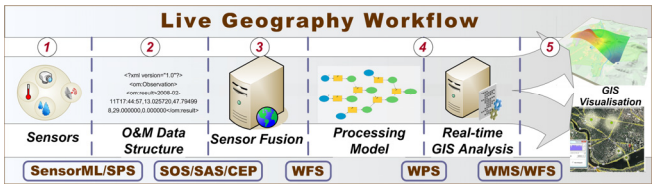


Figure 1. Basic Architecture Components and Standardised Interfaces of the Live Geography Infrastructure.

Generally speaking, the infrastructure shown in Fig. 1 can be sub-divided in five components, i.e., stand-alone parts, which have to be conflated. Component **1** is the geo-sensor network itself including measurement devices Global Navigation Satellite System (GNSS) connectivity and basic processing capabilities. Component **2** covers the communication of the sensor network with a data centre via a variety of wireless and wired transmission technologies. Component **3** deals with sensor fusion, i.e., the harmonisation of measurements stemming from different heterogeneous sensor networks. Component **4** comprises the application-specific analysis of the sensor data on the server side. This operation does not only include pure sensor data processing, but also the integration of static and legacy geospatial data. Component **5** finally treats the presentation of analysed data depending on the particular requirements of end users or user groups.

As the Live Geography approach accounts for the entire workflow, it builds the architectural bridge between domain-independent sensor network development and use case specific requirements for end user sensitive information output.

The implementation of the Live Geography approach only became feasible through the sharp decline of sensor costs over the last decade and intense research in sensor technologies (for an overview see [21]). This fact, together with miniaturisation efforts, increasing monitoring demands due to increasing pressure on resources, environmental regulations, security regulations and – at least partially - rising awareness of the benefits of automated real-time sensor applications, resulted in the deployment of a number of geo-sensor networks.

This in turn will result in the emergence of vast amounts of sensor data during the next years. A main challenge will be to harmonise these data and integrate them in real-time into geospatial analysis systems.

IV. LIVE GEOGRAPHY: IMPLEMENTATION OF AN INTEROPERABLE EMBEDDED GEO-SENSOR WEB

A. Standardisation Enabling Open Measurement Infrastructures

The increasing amount of data measured triggers a more extensive use of open standards and geospatial web services for structuring and managing heterogeneous data. The Open Geospatial Consortium (OGC) has achieved remarkable progress in setting up necessary standards (see Sub-section IV.C). One of the remaining challenges is the distributed processing of large amounts of sensor data in real-time, as the widespread availability of sensor data with high spatial and temporal resolution will increase dramatically with rapidly decreasing prices [8], [21], particularly if costs are driven down by mass utilisation.

From a political and legal standpoint, national and international legislative bodies are called upon to foster the introduction of open standards in public institutions. Strong early efforts in this direction have been made by the European Commission through targeted, including the INSPIRE (*INfrastructure for SPatial InfoRmation in*

Europe), which aims at Europe-wide harmonisation of discovery and usage of geographical data for analysing and solving environmental problems [22].

These regulations both trigger and support the development of ubiquitous and generically applicable real-time data integration mechanisms. Shifting development away from proprietary single-purpose implementations towards interoperable analysis systems will not only enable live assessment our environment, but can also lead to a new perception of our surroundings in general, e.g., expressed by the vision of a "digital skin" [1]. Consequently, this trend may in turn foster the creation of innovative applications that treat the city as an interactive sensing platform, as the *WikiCity* concept [23], involving the people themselves into re-shaping their own socio-technical context. This way, we may enable a manifestation of the vision of "citizens as sensors" [24].

B. Embedded Device Hardware

The measurement device for the concrete implementation presented in this paper has been particularly designed for pervasive GIS applications using ubiquitous embedded sensing technologies. The system has been conceived in such a modular way that the base platform can be used within a variety of sensor web applications such as environmental monitoring, biometric parameter surveillance, critical infrastructure protection or energy network observation by simply changing the interfaced sensors.

The sensor pod itself consists of a COTS embedded device, ISEE IGEPv2 platform including an ARM7-based Cortex A8 600MHz processor with 512MB RAM and 32MB flash memory. Generally speaking, ISEE offers a highly modular and easily expandable system. The computer-on-module (the actual embedded device including CPU, memory and some interfaces) offers two I/O ports, which allows for extensibility of the basic system by specific modules such as GPS, Bluetooth, WiFi, LAN, interface breakouts or a console board for programming the device.

In the configuration for the specific implementation presented within this paper, different sensors (GPS module, LM92 for ambient temperature, SHT15 for air temperature and humidity, NONIN 8000SM oxygen saturation and pulse, or SSM1 radiation sensors) have been attached via standardised interfaces like UART, I^2C, USB, etc. The technical specifics of the sensor pod are described by Resch et al. [9].

The size of the complete sensor pod is approximately 93x65x10mm, i.e., about the size of a chewing gum package. In full load, the device features an energy consumption of <2.2W including a running data query, the GPS module and data transmission via UMTS, which is known to be comparatively energy intensive way of broadcasting data. This configuration yields an operation time of 9.1 hours given a battery capacity of 4000mAh, which is held by a reasonably-sized rechargeable Lithium-ion Polymer (LiPo) battery (140x40x10mm) – whereby capacity and required sizes depend on the specific use case.

C. Embedded Software Infrastructure

The sensing device runs a customised version of the *Ångström* Linux distribution (kernel version 2.6.33) with an overall footprint of about 2MB. The software infrastructure comprises an embedded secure web server (Lighttpd), an SQLite database and several daemons, which convert sensor readings before they are served to the web. The database serves for short-term storage of historic measurements to allow for different error detection procedures and plausibility checks, as well as for non-sophisticated trend analysis.

The hardware drivers for interfacing sensors and reading their measurements make up the low-level part of the embedded software infrastructure. As the geographical position is an essential must-parameter in geo-sensor networks, the sensor pod interfaces a location sensor (e.g., a GPS/Galileo module, a ZigBee/WiFi-based positioning component, etc.).

These measurements are then read by a special sensor daemon that essentially builds the bridge between the sensors and the internal software components. These data are then stored into an embedded database (SQLite), which is held at a maximum data set volume, currently 12500 readings.

The sensor data, which is stored in the database, is then accessed from two different web servers (HTTP/HTTPS and XMPP [Extensible Messaging and Presence Protocol]), which make the measurements accessible from the internet. HTTPS is considered a high enough security level for this implementation providing a secure channel between server and client using the Secure Socket Layer (SSL) protocol. Web Service Security (WSS) would be a viable alternative providing message-based security. However, as WSS is using the SOAP protocol, it is characterised by large overhead, which is not suitable for embedded sensor unit implementations.

Communication of the sensing device with other components in the workflow is based on open standards of the Sensor Web Enablement (SWE) family [25]. This requires a SensorML-conformal description of the measurement platform, Observations and Measurements (O&M) compliant encapsulation of measurement values, as well as an SAS-compliant alerting module. In addition, an embedded database has to be implemented directly on the sensor device to provide for the possibility of short-term data storage, which enables trend analysis and quality assurance, and reduces communication overhead with the central archive database. Thus, the device also implements the following essential standards of the SWE family:

- *Observations & Measurements (O&M)* – O&M allows for the formalised description of sensor measurements in a structured XML-based encoding schema. Thus, O&M can map sensor parameters and their relations. Measurements are organised by quantities, categories as well as their spatial and temporal characteristics.
- *Sensor Model Language (SensorML)* – The Sensor Model Language (SensorML) is a general schema for describing functional models of the sensor.

Information provided by SensorML includes observation and geometry characteristics as well as a description and a documentation of the sensor, and a history of the component's creation, modification, inspection or deployment.

- *Sensor Observation Service (SOS)* – SOS allows for standardised access to sensor measurements (return type O&M) and their platform descriptions (return type SensorML) via a web service interface [26].
- *Sensor Alert Service (SAS)* – SAS is a service for the surveillance of pre-defined rules and trigger specified actions in a particular workflow in case of violation of these rules.

D. In Detail: Embedded Sensor Observation Service (SOS) and Sensor Alert Service (SAS)

The embedded SOS implements the three mandatory methods, *DescribeSensor*, *GetCapabilites* and *GetObservation*. Basically, the service, which is implemented in Common Gateway Interface (CGI), parses the request and creates the according response using appropriate XML templates.

The SOS harmonises raw sensor measurements by encapsulating them into pre-defined XML-based OGC O&M format. This allows for the provision of sensor measurements (numerical values, raster images, binary states, complex or combined measurement data, etc.) in a structured and standardised format.

For generating alerts, the OGC Sensor Alert Service (SAS) standard has been implemented for mobile sensor devices. SAS, which is part of the SWE initiative, specifies interfaces (not a service in the traditional sense) enabling sensors to advertise and publish alerts including according metadata. Alerts are defined as "data" sent from the SAS to the client, which may as well comprise alerts/notifications (e.g., OGC Web notification service [WNS]) as observational data (measurements matching pre-defined criteria) or a Complex Event Processing Engine (CEP). As SAS is based on the standardised XMPP protocol, alerts can be broadcasted very efficiently over the internet to subscribed consumers.

The service implementation presented in this paper supports the mandatory operations as specified in the standard, namely *DescribeSensor*, *DescribeAlert*, *GetCapabilities*, *Subscribe*, *RenewSubscription* and *CancelSubscription* [27].

In this case, SAS is an asynchronous service connecting a sensor in a network to an observation client. In order to receive alerts, a client subscribes to the SAS. If the defined rules apply, a pre-defined alert is sent to the client via XMPP. It shall be stated that the whole communication between the embedded XMPP server (jabberd2) and the client is XML-based for simplifying M2M messaging.

V. Live Geography Portability – Implemented End Applications

This section describes five concrete real-world implementations in different application fields in order to demonstrate that the approach is highly portable, interoperable and flexible in terms of trans-domain usage and integration of heterogeneous data sources. This again builds the basis for the deployment of an overarching monitoring infrastructure for solving real-time analysis questions across a variety of research and service areas.

A. Live Pollutant Monitoring for Public Health

The Common Scents project focuses on real-time pollutant monitoring for public health. As Zardini [28] states, "we have renounced the utopian idea of a socially, politically, and economically perfect city, but not the promise of a perfectly clean and sanitised environment with pure air for breathing."

Thus, the goal of the project, which is a concerted effort of the MIT SENSEable City Lab, the Research Studio iSPACE, the Harvard University Sensor Networks Lab, the City of Cambridge's Public Health Department, and BBN Technologies, is to provide fine-grained air quality information layers in near real-time. To achieve this vision, the CitySense sensor testbed [12] is utilised, measuring CO_2 concentrations along with environmental parameters like wind speed, air temperature, and relative humidity.

The empirical project goal is to provide citizens with up-to-date information to support short-term decisions in real-time. Here, the term "real-time" is not defined by a pre-set numerical time constant, but more by qualitative expressions such as "immediately" or "ad-hoc", i.e., information layers are created in a timely manner to serve application-specific purposes. Detailed results are presented by Resch et al. [29].

The actual implementation shown in Fig. 2 allows for correlating temporal measurement data fluctuation to traffic density, and other day-time related differences. The lower left part of Fig. 2 shows the temporal development of the sensor values, which have been integrated in the standardised O&M format. Running the time series dynamically changes symbologies in the map on the right side accordingly.

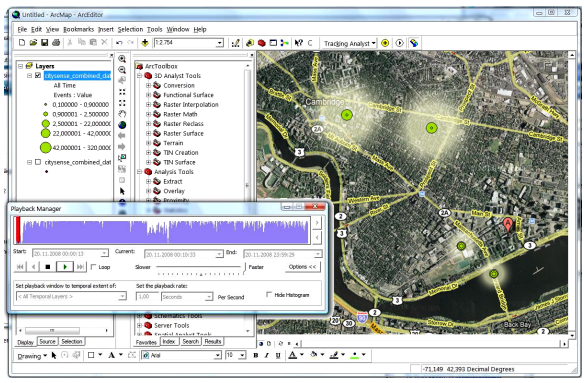


Figure 2. Time Series Visualisation of Pollutant Measurements in an ESRI ArcGIS software environment.

B. *Fine-grained Air Temperature Variations*

Another implementation of the Live Geography has been done in the course of the Real-time Geo-awareness project in a cooperative effort of the Research Studio iSPACE and SYNERGIS Informationssysteme GmbH. Apart from the establishment of the technical components (sensor devices, data integration and analysis), the project's aim was to create a sensor network for fine-grained temperature variation assessment.

The pervasive deployment of temperature sensors can lead to a detection of urban heat islands with a fine spatial resolution. Furthermore, the temperature measurements can be used for correlation with other environmental parameters such as air pollution, ozone or emissions caused by increased traffic emergence. Thus, an essential part of this particular implementation is the alerting functionality, which is achieved by the use of an OGC Sensor Alert Service (SAS), generating alerts according to pre-defined events, i.e., exceedance of pre-defined thresholds. These events are detected by a Complex Event Processing (CEP) engine that also serves for data quality control by identifying measurement outliers and performing other spatio-temporal plausibility controls.

Fig. 3 shows the three-dimensional Inverse Distance Weighting (IDW) interpolation result of air temperature values provided by various OGC Sensor Observation Services (SOS). More implementation details are described by Resch et al. [8].

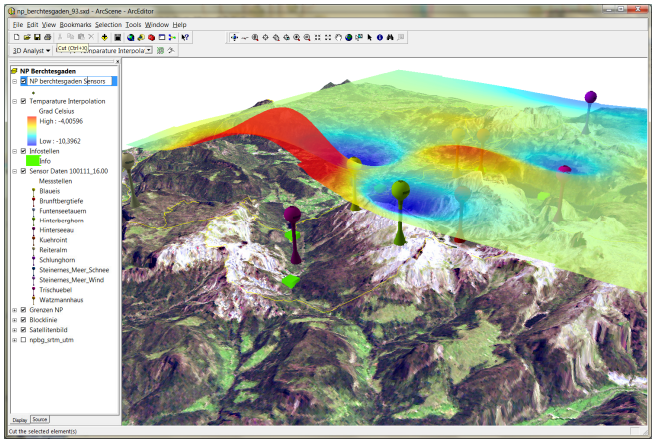


Figure 3. Real-Time Interpolation of Ambient Temperature Values for Monitoring Optimal Environmental Parameters for the Local Fauna and Flora in the National Park Berchtesgaden, Germany.

C. *Ubiquitous Biometric Parameter Surveillance*

The geoHealth Monitor instance of the Live Geography approach responds to the needs of pervasive medical care. The system uses biometric sensors measuring a person's pulse and oxygen saturation in the blood. The project itself has been carried out in cooperation between the Research Studio iSPACE and Salzburg University of Applied Sciences.

The web interface shown in Fig. 4 comprises three sections. Firstly, a configuration panel to select a particular sensing device including different measurement parameters such as the update frequency or the number of measurements stored in the history. The middle section presents the temporal history of OGC Sensor Web Enablement conformal sensor data, which allows for intuitive visual assessment of the measured parameters. Finally, the map on the right side of the interface shows the last few positions of the measurement device to keep track of its spatial trace.

It shall be stated the geoHealth Monitor application cannot only be used for patient surveillance, but may also be employed for equipment tracking, control of the food supply chain including the goods' measured quality condition, or for keeping track of a stolen car.

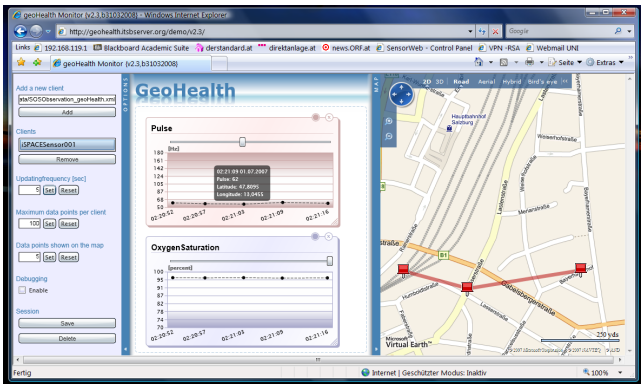


Figure 4. Illustration of a "GeoHealth" application: Biometric Parameter History with Geographical Location Illustration demonstrated in a MS Silverlight environment.

D. *Real-time Air Quality Assessment*

GENESIS (GENeric European Sustainable Information Space for environment), an FP7-funded collaborative research project, has two basic aims: 1.) to establish an open and standards-based infrastructure for managing, analysing and providing environmental information, and 2.) to demonstrate the efficiency of the solution through thematic pilots in different areas within environmental pilot deployments for air quality, water quality and associated health impacts.

The Live Geography approach supports the GENESIS project as it builds the technological foundation for the thematic pilots by providing mechanisms for measurement data provision (Sensor Observation Service), sensor fusion (GeoServer data store), alerting (SAS and CEP) and server-based data analysis (ArcGIS Server application). Fig. 5 illustrates the web interface for live geo-data analysis of environmental information implemented in a kriging process. A special focus in GENESIS is on the coupling of SAS and CEP including the evaluation of the OGC Sensor Event Service (SES), which is widely seen as the successor of SAS. In the project, CEP serves for detecting complex patterns in sensor data related to spatial and temporal parameters as well as measurement values. Another emphasis is on integrating

legacy GIS systems (ArcGIS Server, GRASS GIS, etc.) with the standardised OGC Web Processing Service (WPS) interface to achieve a wholly standardised workflow coupled by a Business Process Execution Language (BPEL) engine.

The Live Geography solution is likely to be integrated into the overall GENESIS infrastructure, which finally aims at Europe-wide Spatial Data Infrastructure (SDI) harmonisation and provision of a complete infrastructure for standardised data access and analysis. For more information on the architecture and outcomes, see [30].

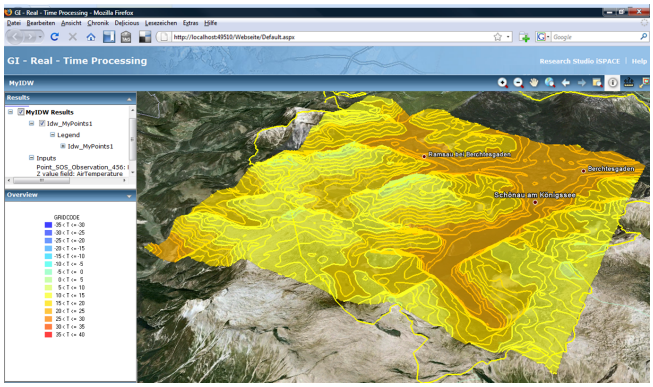


Figure 5. Web-based Live Geo-processing for Fine-grained Real-time Assessment of Urban Air Quality.

E. *Real-time Decision Support for Radiation Safety*

Finally, the Live Geography workflow has been applied and evaluated in the course of the FP6 ERA Star *G2real* project exercise 'Shining Garden' in Seibersdorf, Austria. The field trial setup consisted of modules for live in-situ sensing of gamma radiation (using the SSM-1 radiation detection unit developed by Seibersdorf Laboratories), live geo-processing of radiation measurements, and rapid mapping of up-to-date multi-dimensional sensor information.

The purple dots in Fig. 6 represent the trace of the radiation safety expert carrying the sensor device. Location data and radiation measurements were collected every second. These sensor data were spatially interpolated (in this case using the Inverse Distance Weighting – IDW algorithm). Partitions 1-6 in the figure below show the growing interpolation result in chronological order.

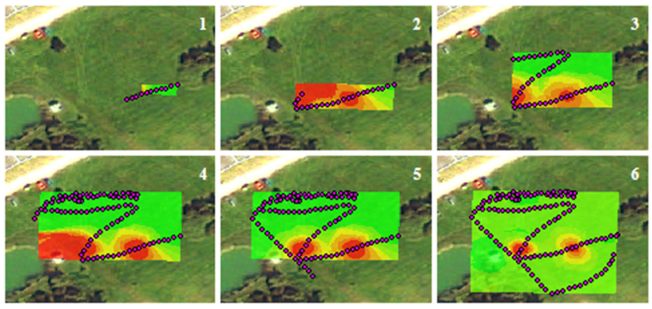


Figure 6. Growing Interpolation Result for Radiation Source Identification.

Results of this experiment confirm that the Live Geography workflow significantly enhances both spatial and situational awareness of people in charge. This in turn enhances time-critical spatial decision support. Detailed results of the field test can be found in [31]. General design challenges are discussed in [9].

VI. Discussion and Conclusion

With the prerequisites and challenges of real-time monitoring in mind, we developed the Live Geography approach. It provides an interoperable, modular and flexible distributed sensing and data analysis infrastructure – as opposed to previous monolithic sensor networks. Thus, it stands for the integration of real-time measurement data in a fully standardised infrastructure for real-time monitoring applications including web-based data processing.

The main benefit of the Live Geography architecture presented in this paper is its composition in loosely-coupled and service-oriented building blocks. This allows for decoupling sensor fusion from CEP, data analysis and visualisation components, enabling flexible and dynamic service chaining. Consequently, the whole infrastructure can be ported easily to various application domains by changing the sensors (what shall be measured) and the process models (how shall the measurements be analysed).

To demonstrate the Live Geography approach's portability, five concrete real-world implementations in different application fields have been presented in this paper. This is to show that the approach is highly portable and flexible in terms of trans-domain usage and integration of heterogeneous data sources. This again builds the basis for the deployment of an overarching monitoring infrastructure for solving real-time analysis questions across a variety of research and service areas. In the future, platforms may generally get more lightweight and portable, which opens up a plethora of new application areas for which platforms have been too expensive or too difficult to deploy before. Another important aspect is real-time data delivery of information on demand.

Consequently, it can be stated that a substantial benefit of the approach is that the developed infrastructure is applicable to a wide variety of cross-domain use cases due to its high degree of interoperability, modularity and flexibility.

With the dramatic decrease of sensor costs as occurring recently, it can be assumed that even larger and even more heterogeneous amounts of measurement data will be available in the near future. A major future research task will be the standardisation and combination of these heterogeneous data sources using internationally standardised interfaces. Recently, several „testbed", „experiment", or „pilot" activities are carried out worldwide such as the Web Services-OWS7 Testbed initiative the OGC which demonstrated the advantages of interoperable measurements infrastructures generally, the OGC Ocean Science Interoperability Experiment (Oceans IE) or the GEOSS Architecture Implementation Pilot. To summarise, although SWE is still evolving and new services will be developed to satisfy emerging requirements of sensor web development OGC and ISO standards for „localised

information“ discovery, retrieval and communication are increasingly providing a baseline needed for interoperability and portability. Furthermore, SensorML can be used outside the scope of SWE enabling long-term archive of sensor data to be reprocessed and refined in the future allowing software systems to process, analyse and perform a visual fusion of multiple sensors [32].

Further research will elucidate the applicability of the Live Geography approach for situation awareness. Although preliminary work has demonstrated the potential of using combination of prevailing sensor data with real-time processing mechanisms to achieve situational awareness for an instantaneous assessment of environmental conditions [33] the advantage of the complete system described in this paper is that hitherto GIS approaches, which offered GIS functionality only in resource-consuming desktop applications, can be replaced by web-based analysis tools. GIS operations are performed on server-side whereas the results are sent to the client, which can for instance be an internet-connected personal computer in the mission control centre or also a tablet PC used by action forces on-site. This will allow for a real-time situational awareness making emergency and rescue actions much more efficient. Situational awareness is the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future [28]. In the context of sensor data about the real world, this may be translated to: (1) detecting and recognising objects and events, (2) determining how they are interrelated, and (3) predicting how things are going to change over a period of time going forward [34]. These developments may change the GIS community significantly and they do so already. Geo-processing featured prominently in the early origins of online GIS where server-based GIS delegated much of the work that a desktop-client would perform to the background, hidden from the user [35].

The Live Geography approach has been made possible through the availability of both the body of standards described herein and the GIS-based investigation tools: we can build the means to facilitate analyses and appropriate characterisations of various processes involved many application areas proofed for five different applications. In turn, we might hope to achieve the further problem formalisation steps thanks to the result of our better understanding of complex systems. This is a prevalent topic in Geographic Information Science [36]. We are making progress beyond a reconstruction of the current state of the world from sensor data to reasoning from observed effects to possible causes. Predicting the future states or course of action is akin to deduction in logic, that is, reasoning from causes to effects [34]. In the future, situation awareness applications may benefit from symbolic representation of the state of the world by adding spatial reasoning capacities to our interoperable framework and by incorporating schemes from time geography into the Live Geography approach allowing structured queries to be performed on data’s temporal and spatial attributes simultaneously.

The ability to obtain all kinds of sensor information at decreasing costs with higher measuring accuracies unlocks research potential and potentially end user applications to creating information at ever higher abstraction levels. This may cause now types of problems. The ability to determine and view locations and associated sensing information with high accuracy is increasingly in the hands of millions of people. Commonly available high-resolution digital terrain and aerial imagery, coupled with GPS-enabled handheld devices, powerful computers, and Web technology, is ultimately changing the quality, utility, and expectations of GIS to serve society. Analytic methods for non-expert users will need to be provided as millions of internet users learn to use data with greater detail and intensity, especially in terms of temporal resolution and the resulting amount of data and level of detail but they may not be aware of basic statistical and cartographic principles. GIS is more and more to be viewed as a media that helps data producers to communicate Geographic information in various forms to receivers, just as newspapers and television communicate more general forms of information [37].

Concluding, it shall be constituted that the main challenge in geo-sensor web research for monitoring applications in the coming years will be to harmonise existing networks with upcoming initiatives in order to guarantee optimal data availability for assessing environmental dynamics. As laid out, this requires a shift from monolithic single-purpose sensor systems to interoperable measurement infrastructures, which necessitates adequate public awareness and policy frameworks. This in turn allows for the straight-forward use of live sensor data in existing spatial decision support systems.

ACKNOWLEDGMENT

Our approach requires expertise in a wide variety of research areas such as sensor networks, data integration, GIS data and analysis, visualisation techniques, etc. We would like to thank all contributing groups at the Research Studio iSPACE, at MIT, at Salzburg University of Applied Sciences for their valuable inputs and contributions in different stages of the development process.

Parts of the developments presented in this paper have been funded by the European Commission (FP7 project *GENESIS*, ref. no. 223996 and ERA STAR Regions project *G2real*, ref. no. 819747) and the Austrian Federal Ministry for Science and Research.

REFERENCES

[1] Resch, B. and Mittlboeck, M. (2010) Live Geography - Interoperable Geo-Sensor Webs Enabling Portability in Monitoring Applications. In: Proceedings of the 2nd IEEE International Conference on Advanced Geographic Information Systems, Applications, and Services - GEOProcessing 2010, St. Maarten, Netherlands Antilles, 10-15 February 2010, pp. 74-79.

[2] Gross, N. (1999) 14: The Earth Will Don an Electronic Skin. http://www.businessweek.com, BusinessWeek Online, 30 August 1999. (2 January 2011)

[3] Gore, A. (2010) The Digital Earth: Understanding our planet in the 21st Century. Speech by Vice President Al Gore, Given at the California Science Center, Los Angeles, California, on January 31, 1998. http://www.isde5.org/al_gore_speech.htm. (4 January 2011)

[4] Craglia, M., Goodchild, M.F., Annoni, A., Camara, G., Gould, M., Kuhn, W., Mark, D., Masser, I., Maguire, D., Liang, S., and Parsons, E. (2008) Next-Generation Digital Earth: A Position Paper from the Vespucci Initiative for the Advancement of Geographic Information Science. International Journal of Spatial Data Infrastructures Research 1(3), pp. 146-167.

[5] Yang, C., Li, W., Xie, J., and Zhou B. (2008) Distributed Geospatial Information Processing: Sharing Distributed Geospatial Resources to Support Digital Earth. International Journal of Digital Earth, 1(3), pp. 259–278.

[6] Nativi, S. (2010) The Implementation of International Geospatial Standards for Earth and Space Sciences. International Journal of Digital Earth, 3(S1), pp. 2–13.

[7] Percivall, G. (2010) The Application of Open Standards to Enhance the Interoperability of Geoscience Information. International Journal of Digital Earth, 3(S1), pp. 14–30.

[8] Resch, B., Mittlboeck, M., Girardin, F., Britter, R., and Ratti, C. (2009) Live Geography – Embedded Sensing for Standardised Urban Environmental Monitoring. International Journal on Advances in Systems and Measurements, 2(2&3), ISSN 1942-261x, pp. 156-167.

[9] Resch, B., Lippautz, M., and Mittlboeck, M. (2010) Pervasive Monitoring - A Standardised Sensor Web Approach for Intelligent Sensing Infrastructures. Sensors - Special Issue "Intelligent Sensors 2010", 10(12), 2010, pp. 11440-11467.

[10] University of Oklahoma (2009) OKCnet. http://okc.mesonet.org, March 2009. (12 January 2011)

[11] Center for Coastal and Land-Margin Research (2009) CORIE. http://www.ccalmr.ogi.edu/CORIE, June 2009 (14 July 2009)

[12] Murty, R., Mainland, G., Rose, I., Chowdhury, A., Gosain, A., Bers, J., and Welsh, M. (2008) CitySense: A Vision for an Urban-Scale Wireless Networking Testbed. Proceedings of the 2008 IEEE International Conference on Technologies for Homeland Security, Waltham, MA, May 2008.

[13] King's College London (2009) The London Air Quality Network. http://www.londonair.org.uk, August 2009. (3 February 2011)

[14] Vargas, R., Allen, M., Swenson, W., and Hamilton, M. (2005) Soil Embedded Networked Systems for Studying Soil Carbon Dynamics: the A-MARSS Project. In Proceedings of Third USDA Symposium on Greenhouse Gases and Carbon Sequestration in Agriculture and Forestry, Baltimore, MA, USA, March 21–24, 2005.

[15] Werner-Allen, G., Lorincz, K., Ruiz, M., Marcillo, O., Johnson, J., Lees, J., and Welsh, M. (2006) Deploying a Wireless Sensor Network on an Active Volcano. IEEE Internet Computing 2006, 10, pp. 18–25.

[16] Nittel, S., Trigoni, N., Ferentinos, K., Neville, F., Nural, A., and Pettigrew, N. (2007) A Drift-tolerant Model for Data Management in Ocean Sensor Networks. Proceedings ACM MobiDE'07, Beijing.

[17] Fredericks, J.J., Botts, M., Cook, T., and Bosch, J. (2009). Integrating Standards in Data QA/QC Into OpenGeospatial Consortium Sensor Observation Services. IEEE Xplore Oceans, 2009.

[18] Hägerstrand, T. (1953) Innovationsförloppet ur korologisk synpunkt, C.W.K Gleerup, Lund, Sweden. Translated as Innovation Diffusion As a Spatial Process, Chicago, University of Chicago Press, 1967.

[19] Pulselli, R.M., Romano, P., Ratti, C., and Tiezzi, E. (2008) Computing Urban Mobile Landscapes Through Monitoring Population Density Based on Cell-phone Chatting. International Journal of Design & Nature and Ecodynamics, 3(2), pp. 121-134.

[20] Nittel, S., Labrinidis, A., and Stefanidis, A. (2006) Introduction in Geosensor Networks. In: Geosensor Networks Nittel, S., Labrinidis, A., Stefanidis, A. (eds.), Springer Lecture Notes in Computer Science 4540, Heidelberg, pp. 1-6.

[21] Nittel, S., (2009) A Survey of Geosensor Networks: Advances in Dynamic Environmental Monitoring. Sensors 9(7), pp. 5664-5678.

[22] European Commission (2009) INSPIRE Directive. http://inspire.jrc.ec.europa.eu, August 2009. (7 February 2011)

[23] Resch, B., Calabrese, F., Ratti, C., and Biderman, A. (2008) An Approach Towards a Real-time Data Exchange Platform System Architecture. In: Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications, Hong Kong, 17-21 March 2008.

[24] Goodchild M.F. (2007) Citizens as Sensors: Web 2.0 and the Volunteering of Geographic Information. *Geofocus*, 7, pp. 8-10.

[25] Botts, M., Percivall, G., Reed, C., and Davidson, J. (Eds.) (2007) OGC® Sensor Web Enablement: Overview and High Level Architecture. http://www.opengeospatial.org, OpenGIS White Paper OGC 07-165, Version 3, 28 December 2007. (17 January 2011)

[26] Na, A. and Priest, M. (Eds.) (2007) Sensor Observation Service. http://www.opengeospatial.org, OpenGIS Implementation Standard OGC 06-009r6, Version 1.0, 26 October 2007. (12 January 2011)

[27] Simonis, I. (Ed.) (2007) Sensor Alert Service. http://www.opengeospatial.org, Candidate OpenGIS Interface Standard OGC 06-028r5, Version 0.9.0, 14 May 2007. (19 January 2011)

[28] Zardini, M. (Ed.) (2006) Sense of the City: An Alternate Approach to Urbanism. 352 pp., ISBN 3-03778-060-6, Lars Müller Publishers, Baden, Switzerland, 2006.

[29] Resch, B., Britter, R., and Ratti, C. (2011) Live Urbanism - Towards the Senseable City and Beyond. In: Pardalos, P. and Rassia, S. (Eds.) (2010) Sustainable Architectural Design: Impacts on Health.

[30] Resch, B., Mittlboeck, M., Lipson, S., Welsh, M., Bers, J., Britter, R., and Ratti, C. (2009) Urban Sensing Revisited – Common Scents: Towards Standardised Geo-sensor Networks for Public Health Monitoring in the City. In: Proceedings of the 11th International Conference on Computers in Urban Planning and Urban Management - CUPUM2009, Hong Kong, 16-18 June 2009.

[31] Sagl, G., Lippautz, M., Resch, B., and Mittlboeck, M. (under review) Near Real-Time Geo-Analyses for Emergency Support: An Exercise for Radiation Safety. In: Proceedings of the 14th AGILE Conference on Geographic Information Science, Utrecht, The Netherlands, 18-21 April 2011.

[32] Chu, X. and Buyya, R. (2007) Service Oriented Sensor Web. In: Mahalik, N. (ed.) Sensor Networks and Configuration Fundamentals, Standards, Platforms, and Applications, Springer, Berling, Heidelberg, pp. 51-74.

[33] Endsley, M. R. (1995) Toward a Theory of Situation Awareness in Dynamic Systems. Human Factors 37(1), pp. 32-64.

[34] K. Thirunarayan, Henson, C., and Sheth, A. (2009) Situation Awareness via Abductive Reasoning from Semantic Sensor Data: A Preliminary Report. International Symposium on Collaborative Technologies and Systems (CTS2009), Workshop on Collaborative Trusted Sensing, Baltimore, Maryland, 2009.

[35] Torrens, P. (2009) Process Models and next-Generation Geographic Information Technology. ArcNews 31(2), pp. 1-5.

[36] Blaschke, T. and Strobl, J. (2010) Geographic Information Science Developments. GIS.Science. Zeitschrift für Geoinformatik 23(1), pp. 9-15.

[37] Sui, D. (1999) GIS as Media? Or How Media Theories Can Help Us Understand GIS and Society. In: Sheppard, E. and McMaster R. (eds.), GIS and Society: An International Perspective.

Call Admission Control Dimensioning for VoIP Traffic over Wireless Access Networks: From Network to Application-specific Perspective

Kiril Kassev, Yakim Mihov, Adriana Kalaydzhieva, Boris Tsankov
Department of Telecommunication Networks
Technical University of Sofia
Sofia, Bulgaria
e-mail: kmk@tu-sofia.bg; yakim_mihov@abv.bg; akalaidjieva@abv.bg; bpt@tu-sofia.bg

Abstract—**Admission control is a key issue for quality of service (QoS) provisioning in both wired and wireless communication networks. Providing QoS for voice traffic transmission over packet-based network is a crucial task, which requires development of accurate resource estimation models. In order to perform bandwidth saving and improve the naturalness of the voice service contemporary voice coding schemes are equipped with the functionality of voice activity detection and background noise transmission during inactive speech periods. The adoption of traditional ON-OFF traffic model may cause significant errors in estimating the bandwidth required to meet the performance bounds of aggregated traffic flow. The aim of this article is to present a "new paradigm" of call admission control (CAC) dimensioning applicable to wireless access transmission media. The admission decision policy is codec-dependent and relies on the concept of user-perceived voice quality, since it could provide a tight connection with the QoS metrics that shall be guaranteed by the network. The well-known bufferless fluid-flow method is applied and new simple exact formulas for CAC performance evaluation are derived. The proposed methods are illustrated with numerical examples and some comparisons are made.**

Keywords-CAC; comfort noise generation; MOS; packet loss; perceived voice quality; VoIP over wireless access networks

I. INTRODUCTION

As next generation networks architecture is moving towards packet, also known as "all-IP" architecture, emerging wireless access technologies rely on fully shared radio resource allocation schemes, which allow scarce resource usage in an efficient way. Interactive services as well as real-time constraining services, such as voice or streaming applications, are supported over shared radio resource. Thus, it becomes critical that the emerging wireless access systems should be capable of employing efficient radio resource management schemes for packet data, in order to ensure that real-time services can be supported according to their stringent QoS requirements.

Providing QoS for real-time traffic flows in modern telecommunication networks is still a challenging task, which can be split up as follows: QoS guarantee in access network and QoS provisioning in the backbone. The former needs more difficult and more costly solutions, compared to the latter. It is due to the cheap available resource (e.g., fiber optic) and easy way of providing additional bandwidth in the core network, in comparison with the scarce and expensive resources in the wireless access networks (WAcN) as well as the specific features of the wireless medium. As a consequence, the QoS requirements may not be satisfied, even though a large amount of resources (i.e. bandwidth) is allocated to a certain connection.

The possibilities of contemporay WAcN, such as Wi-Fi (a set of standards IEEE 802.11), WiMAX (a set of standards IEEE 802.16), and LTE (3GPP standard of Long Term Evolution), to serve voice traffic are subject of particular attention. In spite of the increasing popularity of pure data services, the voice service demands still remain the biggest revenue contributor of telecommunication network operators [2]. VoIP traffic flows over WAcN encounter different problems and particularities like: a) stringent norms of admissible delay; b) the traffic flow is formed by relative short packets with high arrival rate; c) the scarce radio resource is wasted due to the relative long packet header, which can considerably exceed the packet payload, carrying voice frames; d) time-varying channel conditions.

Transmission of voice over packet-based networks is possible by applying different coding techniques. In order to prevent wasting available resources in WAcN voice codecs employing silence suppression techniques are preferred to be used. This is a reasonable solution in contrast to the implementation of constant bit rate (CBR) codecs, which are well accepted in systems where the network resources are not problematic. An additional option of the modern voice codecs is their ability of improving speech quality of parties participating in communication. This is done by generating a special frame, called Silence Insert Descriptor (SID), which describes the talker's background noise. As a result, the traffic profile of aggregated voice traffic should be addressed, by developing accurate models in which the generation of SID frames is included.

Since VoIP services continue to be commercially attractive for network operators, their adoption could be influenced by the users' satisfaction. The major challenge of the packet-based networks (either wireless access or core domain) serving streaming traffic is to provide QoS guarantee, such that consumer satisfaction is the same or similar to that of conventional fixed or cellular telephone services. This should imply methodologies and models for perceived voice quality prediction to be incorporated in

algorithms for network resource management, with respect to QoS provisioning.

On the other hand, an important aspect of radio resource management and QoS provisioning in wireless access networks is towards the CAC mechanism implementation. The necessity of CAC arises with the wide deployment of connection-oriented packet switching technologies. Based on a source's traffic characteristics and required performance metrics, it encompass a set of tools, which has to take a decision whether or not a new connection can be accepted by the system, in addition of those connections in progress. If a new connection is admitted, it must not deteriorate the bandwidth usage and the performance of the connections already established. Since CAC is a fundamental mechanism for congestion control and QoS provisioning, it has been extensively studied in both wired [3] and wireless [4] network domains.

The CAC design and performance analysis became an inseparable part of ATM- and IP-based networks planning process [5], including different wireless networks as well. It can be classified based on various objectives and design options, such as QoS parameters (call-level and packet-level congestion probabilities, packet delay, bandwidth guarantee); throughput optimization, power allocation, fairness; controlling handover dropping probability, etc. Our research work is focused on both call blocking and packet dropping probabilities. Call admission decision is based on a simple threshold rule – an offered call is accepted if all the calls in progress are less than a pre-calculated threshold value.

Taking into consideration the decision time, CAC schemes can be classified as proactive (parameter-based) and reactive (measurement-based). In the former scheme the decision rule is based on a predictive analytical evaluation of the QoS constrains, while in the latter scheme, the CAC decision is based on certain QoS measurement. In order to get more efficient congestion control, a combination of both approaches can be realized.

This article is an extended version of the research work carried out in [1]. We investigate admission control schemes for streaming traffic and propose a new paradigm of a proactive CAC mechanism and VoIP dimensioning framework, based on perceived voice quality evaluation models. We aim at developing a solution that is applicable for a broad class of contemporary VoIP coding schemes.

II. RELATED WORK

The packet form of voice transmission differs considerably from other data transmissions. The VoIP traffic is streaming with stringent delay requirements. The activity periods (talk-spurts) are relatively long and the transmission rate of active voice frames (ACT) is not very high. The packets are relatively short with constant length and more often the packet header size is larger than the payload size. Due to the great interest in using IEEE 802.11 and 802.16 standards as well as emerging 3GPP LTE with E-UTRAN as access networks for packetized voice transmission, there exist numerous publications on modeling the VoIP call performance. References [6]-[8] (Wi-Fi), [9][10] (WiMAX) and [11]-[13] (LTE) are just a few among the latest ones.

In voice communications, speech is usually represented as a sequence of talk-spurts, interleaved with silence periods. This is a consequence of the widespread employment of voice codecs schemes with silence suppression or Voice Activity Detection (VAD) feature. In teletraffic point of view, this led to modeling the VoIP traffic pattern as an ON-OFF source. The analytical modeling of the ON-OFF voice traffic started with voice over ATM and continues with voice over IP applications [5]. Among analytical methods for performance evaluation, Markov-Modulated Poisson Process (MMPP) [14] and Fluid-flow model [15] are well-distinguished. Both models are described and compared in more details in [41][42]. The MMPP approach is more accurate, but at the cost of more computational efforts, when compared with the fluid-flow model, which is usually preferred as a less complex solution [16].

Although quite low bandwidth requirements, the packetized voice features lead to poor traffic performance in the high-speed WAcN [7][8][17]. In [6], the theoretical capacity for VoIP traffic of IEEE 802.11b WAcN is computed to be 15 calls. A comparative study shows the benefit of implementing silence suppression technique, which enhances the theoretical capacity to 38 calls. This is still a poor performance compared with the IEEE 802.11b radio channel PHY rate of 11 Mbps. Thus, it is beneficial to improve the performance and properly dimension the WAcN for VoIP traffic.

Along with the advantages of VAD feature for bandwidth reduction, its application may lead to sudden drop of the signal level during voice inactivity periods (OFF state), which is perceived unpleasant by the other dialogue party. Hence, to fill up this inactive period of time, a description of the background noise characteristics shall be sent from the inactive voice encoder. This is done by SID frames generated by the codec's algorithm. The corresponding output signal is referred to as comfort noise [18]. Hence, we should note that the presence of SID packets in VoIP traffic pattern can affect the traditional ON-OFF traffic evaluation [19]. Since the ON-OFF model is not valid for such a case, in [14] authors even went so far as to suggest the notation ON-SID instead of ON-OFF.

CAC techniques in modern wireless networks have been a subject of intensive study [4]. The design of CAC schemes for voice traffic has often been based on QoS performance metrics, which are treated separately (e.g., a design objective could be to achieve minimum packet loss rate for a specified delay constrains). On the other hand, in contrast to the TDM systems in which G.711 CBR codec can be only used, voice communications over packet networks can be realized by a number of low-bit-rate codecs for the purposes of bandwidth saving. Due to the complicated signal processing algorithms, user satisfaction strongly depends on the robustness of the particular type of codec to the instantaneous network performance (e.g., packet loss rate). In order to provide tight connection between both voice quality perceived by users and QoS parameters that shall be guaranteed by the network, either subjective or objective methods, developed through the years, should be incorporated. The Mean Opinion Score (MOS) is the most widely used subjective measure of voice

quality expression and the ITU-T E-model is a computational model for predicting voice quality from network parameters [20]-[22].

MOS-based rate adaptation for VoIP sources has been incorporated in [23]. Architecture for adaptive control of a VoIP source coding rate, based on the state of the network, is proposed. The goal is to maximize the voice quality perceived by the receiver. Similar approach is presented in [24], where dynamic joint source and channel coding adaptation algorithm for the AMR speech codec is proposed. It is aimed at finding the optimum solution between packet loss recovery and end-to-end delay in either wired or wireless networks, in order to maximize the perceived voice quality. The ITU-T E-model is successfully incorporated in [25], which proposes an optimization algorithm that selects coding scheme, packet loss bound and maximum link utilization for a VoIP connection.

There are a large number of researches concerning the voice quality prediction models and their main application in playout buffer optimization algorithms. This is due to the fact that in the past, the choice of a buffer algorithm was entirely based on buffer delay and loss performance, treated separately (e.g., minimum end-to-end delay for a given packet loss).

In [26] a method for enhancing perceived quality of streaming applications and adaptive playout buffer control is proposed. The method is based on the assumption that the relationship between MOS and packet loss for codecs is linear, which is not correct, especially for newer codecs.

In [27] the assessment of how buffer algorithms affect perceived voice quality and how to choose the best algorithm and its parameters to obtain the optimal perceived quality has been carried out. The results are based on Internet trace data measurement and a new methodology for predicting speech quality, which combines the ITU-T Perceptual Evaluation of Speech Quality (PESQ) and ITU-T E-model.

Taking into consideration the widespread use of the E-model for voice quality prediction, Sun [28] points out that it is suitable for a limited number of codecs and network conditions. This is expressed by the necessity of performing time-consuming subjective test in order to derive model parameters. As a result, new accurate models for objective, nonintrusive voice quality prediction in packet networks are proposed. Based on a new methodology, authors of [28] propose efficient regression models, which can be applied for a number of modern codecs.

The packet loss probability evaluation and accurate VoIP bandwidth estimation for aggregate traffic, with respect to the perceived voice quality metrics is a crucial task in a CAC dimensioning. Comparative study shows that a little of research activities concerning this topic area have been carried out so far. A new approach to solving this problem is a subject of the present article, taking into account VAD and Comfort Noise Generation (CNG) features of the contemporary voice coding schemes (codecs).

III. System Model

The current section deals with the development of analytical models for call- and talk-spurt level performance evaluation of the proposed CAC mechanisms for VoIP traffic transmission.

Almost all known publications on packetized voice traffic performance over WAcN have a common feature – the investigations are carried out under overloaded (or nearly overloaded) system conditions. It is explained merely by the fact that the investigations are fulfilled by means of simulations. Good references are [29] and [30], where extensive research work and interesting characteristics of VoIP traffic service through a IEEE 802.16 system are obtained by means of simulations. Telecommunication service providers are mainly interested in system behavior under real traffic load, where the QoS measures such as call blocking, packet loss, etc. are rare events. The reason is that the QoS norms applied for commercial public networks restrict the traffic volume before an overload can occur. Direct simulation of rare events is time and resource consuming process. Thus, simulation acceleration methods are necessary to reduce the computing time. These methods may face certain problems. The proper setting of a particular method's parameters is critical for its performance and accurate results evaluation. An effective alternative is to employ analytical methods for system performance evaluation

Another reason to have a preference on analytical methods is the necessity of cross-layer design application due to the radio channel characteristics. Multi-layer simulations are often hard to be performed because they involve widely different time scales [31] (e.g., call request arrival, packet arrivals, packet processing at MAC level, etc.). Performance analysis by means of multi-layer simulation will take large amount of time and may be unpractical for a broad class of problems analysis.

The subject of interest is the traffic flow generated by multiple homogeneous sources and the bursty traffic in particular. Our considerations are restricted to streaming (real-time) traffic, generated by VoIP sources. We aim at developing an approach for Generalized VoIP (GVoIP) traffic source characterization and parameters estimation of a CAC for an access network.

A. Modeling of VoIP traffic with VAD and SID over wireless systems

A WLAN, WMAN or E-UTRAN, being an access network, is just one hop of a multi-hop end-to-end connection. The total one-way delay for good voice quality is fixed by ITU-T recommendations [32] to be less than 150 ms. Both observations and simulations show that the MAC functional delay is about 15 – 20 ms [9]. The same delay sustains at the other end. An additional delay of 50–60 ms can occur due to the necessity of jitter buffer [19]. Therefore, after extraction of possible queues delays of the backbone routers the remaining delay budget for a wireless access network is quite limited.

The stringent requirements on delays limit the ways in which the voice traffic losses (packet-scale and talk-spurt-scale) can be handled. The packet-scale losses are easy to be prevented by means of a short buffer. However, it is not practical to prevent talk-spurt-scale losses by means of a

buffer. The buffer in this case would have to be large enough, and this would introduce an unacceptable delay. The talk-spurt-scale losses can be reduced to an admissible amount by selecting an appropriate medium transmission rate C during the wireless access network traffic planning process.

In this article, the bufferless fluid-flow approach is used for CAC parameters determination. The assumption that there is not a buffer at talk-spurt level leads to conservative estimates for packet losses and therefore to the safety side of CAC parameter determination.

The bufferless fluid-flow model is used quite a while ago [33]–[35] and [5, Chapter 12] due to its effectiveness and simplicity.

In the majority of investigations on VoIP traffic performance analysis, it is widely adopted the speech generation process to be modeled as a consequence of talk spurts and inactive periods, whose duration is usually assumed to be exponentially distributed. The so defined ON-OFF model does not take into consideration the existence of SID frames in the voice traffic pattern, generated by the voice encoder. When the source is in the ON state, it produces voice frames with a constant bit rate, which are encapsulated in voice packets. During the OFF state, the source sends no packets.

In order to determine the maximum number of calls (sessions) N admitted to the system by the CAC, meeting certain objectives, the bufferless fluid-flow or burst-scale loss approach is well-accepted.

An ON-OFF traffic source is usually characterized by means of the following parameters: the bit rate during a talk-spurt R; the mean bit rate r; the talk-spurt duration T_{ON} and the inactive period duration T_{OFF}. The mean bit-rate during ACT frames (packets) generation process is:

$$r = \frac{T_{ON}}{T_{ON} + T_{OFF}} R = \alpha R. \quad (1)$$

According to (1), an ON-OFF source could be characterized by means of any three out of four parameters. It should be noted that $1/\alpha$ measures the peak to mean ratio of the rate produced by a call and α denotes the voice activity factor.

By employing voice encoders that randomly generate SID frames (packets) during voice inactive periods, the OFF state bit-rate is $R_{OFF} > 0$. The VoIP traffic source with VAD and background noise transmission is called a *Generalized VoIP* (*GVoIP*) source [36]. A GVoIP traffic source is characterized by means of the above mentioned parameters plus the bit rate during the inactive period R_{OFF}. Hence, the overall mean bit rate r^G of a GVoIP source is derived as:

$$r^G = \alpha R + (1-\alpha) R_{OFF}. \quad (2)$$

Since SID frames generation has an influence on the overall packet flow, it could be quantitatively expressed as the ratio R_{OFF} / R, denoted by γ [36]. Thus, Equation (2) can be rewritten as follows:

$$r^G = \alpha R + (1-\alpha) R_{OFF} = \alpha R \left[1 + \frac{1-\alpha}{\alpha} \gamma \right]. \quad (3)$$

Comparison of (1) and (3) gives an increase of the voice mean bit rate due to the SID frames generation, i.e.

$$r^G = r \left[1 + \frac{1-\alpha}{\alpha} \gamma \right]. \quad (4)$$

The actual value of γ can vary, but usually the typical values are limited to 0.1 [37]. On the other hand, the typical values of α range from 0.35 to 0.45 [6]. Therefore, it could be expected an approximate increase of a GVoIP source mean bit rate up to 20 %.

For an aggregated model of multiple homogeneous VoIP sources, which generate SID frames, the arriving flow rate (AFR) of the multiplexed voice calls is:

$$AFR = jR + (i-j) R_{OFF}, \quad (5)$$

where, in the general case, i denotes the number of calls admitted to the system and j – the number of calls in talk-spurt.

B. System model and packet loss paradigm

Let us formulate the problem of aggregating a number of voice traffic sources over the radio interface of an access network. We suppose C represents the total bit rate link capacity allocated by a base station, of a particular wireless access technology, for the purposes of streaming traffic transmission. It is convenient to use the notation R (transmission rate during talk-spurt period) as a unit transmission resource. Therefore,

$$C = nR, \quad (6)$$

where

$$n = C / R \quad (7)$$

denotes the number of network resources (referred to as transmission resource units as well). In case of a classical ON-OFF model (without SID) n represents the maximum number of active calls that can be simultaneously served without any packet losses.

However, if n^G denotes the maximum number of active GVoIP traffic sources, including CNG feature that can be served simultaneously without packet losses, then:

$$C = n^G R + (i - n^G) R_{OFF}. \quad (8)$$

As a consequence:

$$n^G = \frac{\frac{C}{R} - i\gamma}{1-\gamma} = \frac{n - i\gamma}{1-\gamma}. \quad (9)$$

Hence, the maximum number of talk-spurts (active calls) that can be simultaneously served without packet losses is not constant any more, but depends on the number of the calls *i* admitted to the system. Recall that in the classical ON-OFF model, this value is constant and depends on *C* only (7).

The main parameter of a proactive CAC scheme is the maximum number of calls (sessions) admitted to the system by the CAC, denoted by *N*. In order to gain from the statistical multiplexing, it is obvious that $N > n^G$. On the other hand, the following expression $N < C / r^G$ shall be fulfilled in order to prevent a buffer from permanently overflowing. Hence, the following condition can be derived:

$$n^G < N < \frac{C}{r^G}. \quad (10)$$

The research efforts will be directed towards determining both the CAC threshold value *N* of maximum admissible calls (to keep the call blocking below the norm) and the radio-link resources *n* (necessary to keep the packet loss value below the norm).

The analytical evaluation is based on the following parameters: the offered traffic *A*, in terms of Erlangs; the GVoIP source parameters (such as, T_{ON}, T_{OFF}, *R*, and R_{OFF}); the QoS parameters – call blocking probability *B* and packet loss probability P_{PL} .

Our attention is paid to the packet loss evaluation, as a more intricate task. The analytical model derived can be used for evaluation of the necessary medium transmission rate *C*, with respect to the packet loss probability requirements.

The ratio of the packets loss rate to the arriving packets rate gives the probability of packet losses [5]

$$P_{PL} = \frac{E(AFR - C)^+}{E(AFR)}, \quad (11)$$

where *E* stands for the expectation operator, the numerator denotes the aggregated flow excess rate, and $(z)^+=\max(z,0)$.

Assuming the worst-case with *N* admitted calls (i.e. *i*=*N*) and using *R* as a transmission resource unit, the packet loss probability is derived by the following expression:

$$P_{PL} = \frac{\sum_{j=\left\lceil \frac{n-N\gamma}{1-\gamma} \right\rceil}^{N} P(j \mid N) \cdot \left[j + (N-j)\gamma - n \right]}{\sum_{j=0}^{N} P(j \mid N) \cdot \left[j + (N-j)\gamma \right]}, \quad (12)$$

where *j* represents the number of talk-spurts (active sources). In the case of multiple sources, we need to calculate the probability that *j* sources are active, given that *i* traffic sources are admitted. This conditional probability is given by the binomial distribution:

$$P(j \mid i) = \binom{i}{j} \cdot \alpha^j \cdot (1-\alpha)^{i-j}. \quad (13)$$

For the worst-case with *N* admitted calls, *i* = *N*.

Both the numerator and the denominator in (12) represent flows that are mixture of packets that carry voice (ACT frames) and packets that carry background comfort noise (SID frames). As Estepa [16][38] have correctly pointed out, since the loss of ACT packets forms the main influence on the perceived voice quality, it makes sense to account the loss of ACT packets only. The proportion between ACT and SID packets in the excess rate flow varies and depends on the number *j* of talk-spurts. In the aggregated excess rate flow exactly $\frac{j}{\left[j + (N-j)\gamma \right]}$ parts belong to ACT packets and the overall offered ACT flow is merely $\alpha \cdot N$. Therefore, ACT packets loss probability is expressed as:

$$P_{PL}^{ACT} = \frac{\sum_{j=\left\lceil \frac{n-N\gamma}{1-\gamma} \right\rceil}^{N} P(j \mid N) \cdot \left[j - \frac{n \cdot j}{j + (N-j)\gamma} \right]}{N\alpha}. \quad (14)$$

It should be noted that in [16][38] the following approximation formula for P_{PL}^{ACT} is used (rewritten here for the bufferless fluid-flow model and based on notations we have adopted):

$$P_{PL}^{ACT} \approx P_{PL} \cdot \left[1 + \frac{(1-\alpha)\gamma}{\alpha} \right], \quad (15)$$

where P_{PL} is evaluated by (12).

A common feature, when *N* is determined, is to consider the case *i*=*N* only and evaluate P_{PL} [5, pp. 141]. This corresponds to a system where all traffic sources suffer highest losses. For carrier-grade voice service the network operator has to guarantee QoS similar to that of circuit switched networks. For this reason, for a properly planned network the common assumption of *i*=*N* tends to be an extremely rare event. As a consequence, it is significantly important to take into consideration all possible system states for which losses are encountered, when performing an analytical evaluation of P_{PL}.

A VoIP source is either in an idle or busy state. In case of VAD function implementation the VoIP source is alternating OFF and ON states after the call is setup (Fig. 1a). An assumption is made that any call originates or terminates when it is in silence period only. In case a call can originate or terminate during an active period (during a talk-spurt) the call state-transition diagram is modified as shown on Fig. 1b. Probabilities of transition from state *busy ON* to state *idle*

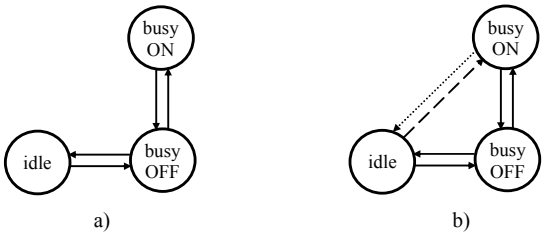


Figure 1. A VoIP source state-transition diagram

(dotted line on Fig. 1b) and particularly from state *idle* to state *busy ON* (dashed line on Fig. 1b) are negligible (normally there is a packet generation offset at the call start and seldom the call end will interrupt talk-spurt) and they will be omitted in our considerations.

At any time the radio link system can be in a state (i, j) where i $(i = 0,1,...,N)$ represents the number of accepted calls (N denotes the upper bound of the admitted calls) and j $(j = 0,1,...,i)$ is the number of active calls (number of talk-spurts in progress). The call flow forms a Poisson process with call rate Λ_c and call service time $1/\mu_c$, whereas the burst flow forms a Binomial process with single ON source burst arrival rate $\lambda_b = 1/T_{OFF}$ and single OFF source burst service rate $\mu_b = 1/T_{ON}$. The set of states (i, j) forms a two-dimensional Markov chain with the corresponding state-transition diagram shown on Fig. 2.

Following the assumption the call flow forms a Poisson process, the probability of exactly i sources being busy is given by

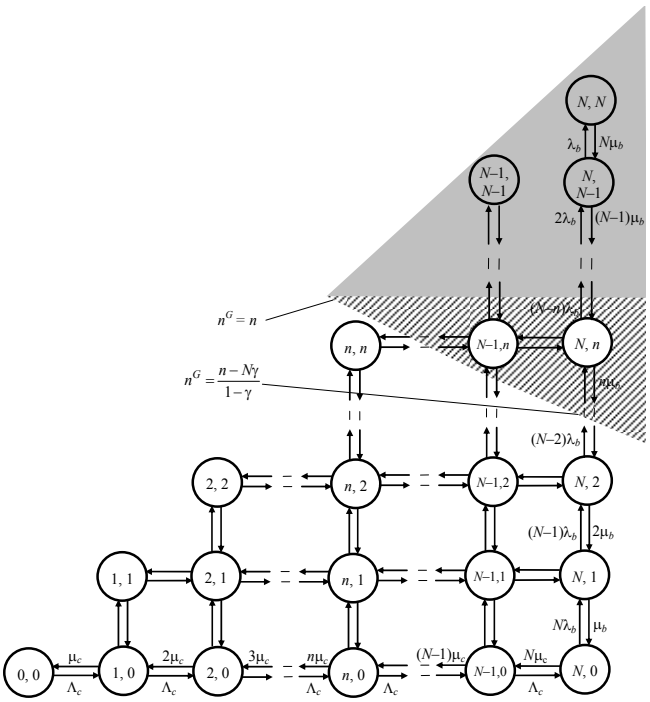


Figure 2. System state-transition diagram

$$P(i) = \frac{\frac{A_c^i}{i!}}{\sum_{x=0}^{N} \frac{A_c^x}{x!}}, \tag{16}$$

where

$$A_c = \frac{\Lambda_c}{\mu_c}.$$

The Markov process, which is represented by the two-dimensional state transition diagram on Fig. 2, is reversible. Thus, the performance measures of interest can be derived by the states probabilities, which are given on product form. A state joint probability $P(i, j)$, under the assumption of statistical equilibrium, is expressed as

$$P(i, j) = P(i) \cdot P(j \mid i), \tag{17}$$

where the conditional (burst flow) probability $P(j \mid i)$ is obtained by (13).

Let us first consider the case when the aggregated traffic flow is generated by multiple VoIP sources, where each one is represented as an ON-OFF traffic model.

Since we are interested in packet loss probability evaluation P_{PL}, in any state for which $j > n$ (the gray-filled area on Fig. 2) a packet is lost with certain probability. The offered rate in a state (i, j) is jR and the excess rate is $(j - n)R$. Therefore, the excess rate mean value is given by

$$\sum_{i=\lceil n \rceil}^{N} \sum_{j=\lceil n \rceil}^{i} R(j - n)P(i, j). \tag{18}$$

Based on (11) and after some rearrangements, the packet loss probability is given by the following relation:

$$P_{PL} = \frac{\sum_{i=\lceil n \rceil}^{N} \sum_{j=\lceil n \rceil}^{i} (j - n)P(i, j)}{\sum_{i=\lceil n \rceil}^{N} P(i) \sum_{j=0}^{i} jP(j / i)} = \frac{\sum_{i=\lceil n \rceil}^{N} \sum_{j=\lceil n \rceil}^{i} (j - n)P(i, j)}{\sum_{i=\lceil n \rceil}^{N} P(i) i\alpha}. \tag{19}$$

In order to perform a comparative analysis considering the worst-case scenario, i.e. $i = N$ [5, pp. 141], P_{PL} could be derived from (19)

$$P_{PL} = \frac{\sum_{j=\lceil n \rceil}^{N} (j - n)P(j / N)}{N\alpha}. \tag{20}$$

As a further step of the research, it should be noted that the traffic generation pattern of the contemporary VoIP

coding schemes comprises of SID packets generated during the OFF periods. Thus, the presence of such packets affects the traditional ON-OFF traffic model and hence, the packet loss evaluation methodology when GVoIP sources are employed.

Based on the state-transition diagram (Fig. 2) as well as the main properties of GVoIP sources, in any state for which $j > n^G$ a packet loss with certain probability occurs. On the other hand, it could be seen that the parameter n^G is tightly coupled with the number of admitted calls, which is expressed by (9).

Following (11), it is more realistic to consider the ratio of lost packets to all arrived packets just for VoIP calls with packet losses only, i.e. $i > n$ (Fig. 2).

Hence, the packet loss evaluation of aggregated GVoIP traffic flow could be obtained by the following expression:

$$P_{PL} = \frac{\sum_{i=\lceil n \rceil}^{N} \sum_{j=\left\lceil \frac{n-i\gamma}{1-\gamma} \right\rceil}^{i} P(i,j) \cdot [j + (i-j)\gamma - n]}{\sum_{i=\lceil n \rceil}^{N} \sum_{j=0}^{i} P(i,j) \cdot [j + (i-j)\gamma]}. \quad (21)$$

Both the numerator and denominator in (21) represent a packet flow, which is a mixture of packets carrying voice (ACT packets) and those carrying background comfort noise pattern (SID packets). As it was pointed out earlier, we are interested in the ACT packets loss evaluation, due to their main influence on the perceived voice quality. The proportion between ACT and SID packets in the excess rate flow varies and depends on both the number of calls (*i*) and the number of talk-spurts (*j*). It should be noted that the total offered ACT flow at states with *i* calls is $\alpha \cdot i$. Therefore, we propose the following exact packet loss formula:

$$P_{PL}^{ACT} = \frac{\sum_{i=\lceil n \rceil}^{N} \sum_{j=\left\lceil \frac{n-i\gamma}{1-\gamma} \right\rceil}^{i} P(i,j) \cdot [j - \frac{nj}{j + (i-j)\gamma}]}{\sum_{i=\lceil n \rceil}^{N} P(i) \alpha i}. \quad (22)$$

The main contribution of the models we have developed is to propose a new paradigm of CAC dimensioning for VoIP traffic pattern generated from homogeneous sources with VAD and CNG features. As further part of the research, we address this problem from the users' perspective as well, by incorporating the broad term of perceived voice quality.

C. CAC Dimensioning with respect to the perceived voice quality

The concept of user-perceived QoS and its evaluation for real-time services, such as VoIP, is a key issue, since it could provide a tight connection with the QoS metrics that shall be guaranteed by the network. In VoIP applications the delay and packet losses are the main impairments that have direct impact on the perceived voice quality, and the MOS is the most widely used measure for this purpose.

Due to variety of voice coding schemes (codecs) and their robustness to the network conditions, all the information about the type of codec and packet losses is suitably represented by an appropriate equipment impairment model I_e, while the delay impairment model I_d encompasses a number of impairments due to one way delay of voice packets.

Since the models' parameters are usually calculated by a set of complex equations [22], efficient regression models for objective, nonintrusive voice quality prediction in packet networks are proposed in [28], which combine the ITU-T speech quality measurement algorithms (either PESQ or PESQ-LQ) and ITU-T E-model.

A non-linear equipment impairment I_e regression model for a number of contemporary low-bit-rate codecs is derived in [29] and has the following form:

$$I_e = a \ln(1 + b\rho) + c, \quad (23)$$

where ρ denotes the packet loss probability (in percentage), and the parameters (*a*, *b*, and *c*) are derived under the PESQ or PESQ-LQ algorithm and depend on the particular coding scheme used. Employing complicated signal processing algorithms, voice codecs can also have an impact on the perceived quality under zero packet loss and delay conditions. This is expressed by the parameter *c* in (23).

Unlike I_e which is codec dependent, the I_d factor is common to all codecs and it could be derived by a polynomial fitting of 6-th order [28]. Among all the impairments included in I_d, we take into consideration the following ones (assuming the rest to be in perfect condition):

(a) *packetization delay* d_{pack} – time taken to fill the packet payload, i.e. for the purposes of the packetization process it is necessary the number of codec frames N_{fpp} that will be encapsulated in each packet to be set. This option has a great importance, because for a particular type of codec, it can influence the bit rate the ACT and SID packets are generated. The packet mean bit rate is significantly increased, especially for low values of N_{fpp}, and vice versa. On the other hand, there is a trade-off between the packets mean bit rate and the packetization delay;

(b) *network delay* – contemporary packet-based networks are too complex and large-scale systems. Thus, standard techniques for delay analysis are not well-suited and a common practice for such studies lies in the scope of statistical delay distribution modeling, based on a network configuration and acquired trace data. This topic is covered in more depth in [39] and several statistical distribution functions are investigated and applied in [40] for a network delay description. As a consequence of applying statistical analysis, the network delay can be expressed by its mean value, denoted by $\overline{d_n}$. The overall mean packet delay is simply represented by $\overline{d} = d_{pack} + \overline{d_n}$.

Taking as a starting point the VoIP packet loss model in which all possible system states are taken into consideration (Fig. 2), we extend it to include the models for perceived voice quality prediction addressed previously. We aim at

deriving the overall mean impairment factor $\overline{I_\Sigma}$, and hence the perceived voice quality, in terms of MOS, with respect to the parameters describing the underlying Markov process.

Carrying out the analysis, we can split up the state space in two non-overlapping areas, which are distinguished to each other by the packet loss occurrence. Thus, for $i \in (0, n]$, the packets are not lost and the mean impairment factor $\overline{I_\Sigma^{(0,n]}}$ includes the impairments of the codecs itself (under zero packet loss) and packets delay.

$$\overline{I_\Sigma^{(0,n]}} = (c + I_d) \sum_{i=0}^{\lfloor n \rfloor} P(i). \tag{24}$$

For $i \in (n, N]$ the influence of packet losses on perceived voice quality is quantitatively evaluated by applying the I_e model. The mean impairment factor $\overline{I_\Sigma^{(n,N]}}$ in such a case is obtained as follows

$$\overline{I_\Sigma^{(n,N]}} = \sum_{i=\lfloor n+1 \rfloor}^{N} P(i) \cdot \left[a \ln(1 + b\rho) + c + I_d \right], \tag{25}$$

where ρ denotes the mean packet loss rate (in percentage) for traffic flow of both ACT and SID packets and subject to $\forall i \in (n, N]$:

$$\rho = \frac{\sum_{j=\left\lceil \frac{n - i\gamma}{1-\gamma} \right\rceil}^{i} P(j \mid i)\left[j + (i-j)\gamma - n \right]}{\sum_{j=0}^{i} P(j \mid i)\left[j + (i-j)\gamma \right]} \cdot 100.$$

Therefore, for the overall mean impairment factor $\overline{I_\Sigma}$ we have

$$\overline{I_\Sigma} = (c + I_d) \sum_{i=0}^{\lfloor n \rfloor} \hat{P}(i) + \sum_{i=\lfloor n+1 \rfloor}^{N} \hat{P}(i) \left[a \ln(1 + b\rho) + c + I_d \right]. \tag{26}$$

If p denotes a threshold value of the set of states (at call-flow level) we are interested in during the perceived voice quality analysis, the following conditions shall be fulfilled: $0 \le p \le N$ and $p \le i \le N$. Hence, the common representation of the weighted factors in (26) is based on normalization of the set of probabilities $P(i)$, subject to the stated conditions, i.e.

$$\hat{P}(i) = \frac{P(i)}{\sum_{i=p}^{N} P(i)}, \qquad 0 < \hat{P}(i) \le 1. \tag{27}$$

$$\sum_{i=p}^{N} \hat{P}(i) = 1.$$

Having obtained the average impairment factor $\overline{I_\Sigma}$, and ignoring the other impairments (e.g., echo), the average R factor can be calculated as [22] and it commonly ranges from 50 (poor quality) to 100 (the best quality), i.e.

$$\overline{R} = R_0 - \overline{I_\Sigma} + A, \tag{28}$$

where R_0 incorporates the effect of noise, expressed by the basic SNR, and it does not depend on the network performance. For voice traffic it is assumed R_0 = 93.2. The advantage factor A takes into account the fact that some users could accept the voice quality reduction in return to the service convenience, for instance wireless access connection. Typical values of the advantage factor are A = 0 (wireline access) and A = 10 (GSM network access). Thus, the average R factor, expressed by (28), considers all the impairments as a result of particular network condition and the type of coding scheme employed.

The perceived voice quality is quantitatively expressed by the relationship between the R factor and MOS, as defined in ITU-T G.107 recommendation [22].

IV. NUMERICAL RESULTS

The current section is concerned with a quantitative analysis and comparative study of the methods for CAC dimensioning we have proposed.

In order to decrease the bandwidth usage the encoding scheme of each traffic source employs an activity detection function, which is quantitatively represented by the activity factor α. The offered traffic flow A_c is generated by multiple homogeneous (G)VoIP sources. The maximum number of calls (sessions) admitted to the system depends on the target call-level blocking probability B, which can be obtained by (16).

The commonly accepted ON-OFF model ignores the SID packets in the VoIP traffic pattern (γ = 0) and thus, its application could cause significant errors in estimating the bandwidth required to meet the performance bounds of aggregated traffic flow. Focusing on a more realistic case, where voice source traffic pattern includes both a silence suppression feature and comfort noise generation, we analyze a CAC mechanism assuming the worst-case scenario with N admitted calls, i.e. $i = N$. We set the ACT packets loss probability threshold (14) to $P_{PL}^{ACT} = 0.5\%$ [21]. The aim of the study is to determine the maximum number of admitted users N in case a fixed amount of network resource units n is allocated for the VoIP service. The quantitative relationship among the variables of interest is shown on Figure 3. In spite of improving the naturalness of conversations by introducing the CNG feature, SID packets generation during inactive periods (γ = 0.1) leads to additional consumption of allocated resources. This results in decreasing the number of users admitted to the system.

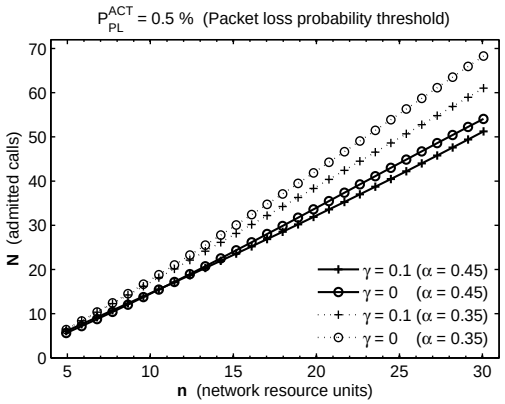


Figure 3. Number of admitted calls vs. allocated network resource units (a comparative analysis of multiplexing homogeneous VoIP and GVoIP sources, subject to $i = N$)

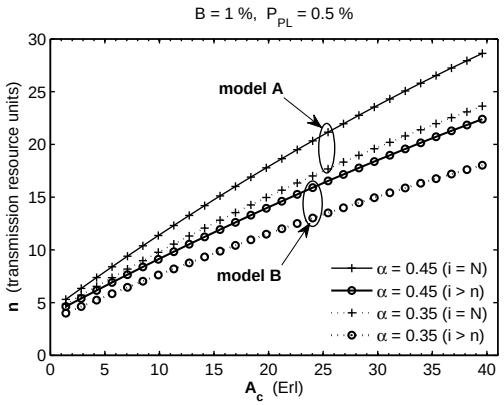


Figure 4. Access network dimensioning – CAC models comparison (homogeneous voice sources without CNG, subject to $i = N$ and $i > n$)

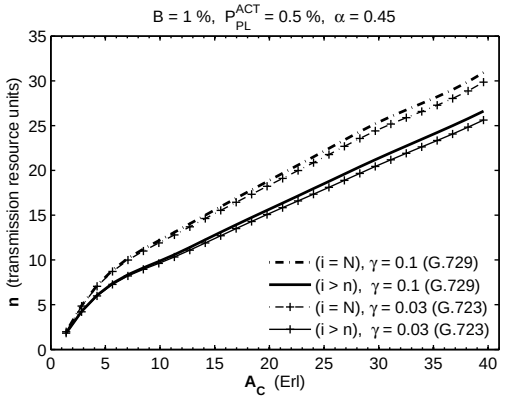


Figure 5. Comparative study of the dimensioning algorithm applicable for G.729 and G.723 coding schemes (homogeneous GVoIP sources, subject to $i = N$ and $i > n$)

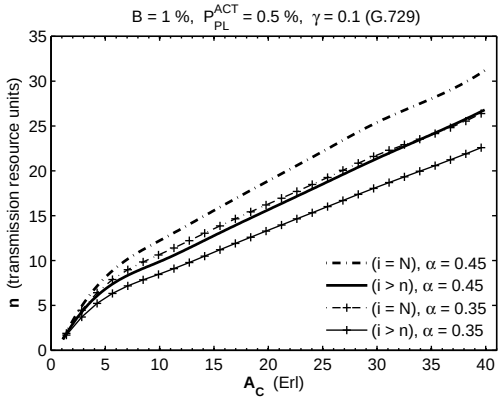


Figure 6. Comparative study of the dimensioning algorithm applicable for G.729 coding scheme

Based on the required performance thresholds, such as B and P_{PL}, as well as source traffic characteristics the task of CAC is to determine whether a connection can be accepted and, if accepted, the amount of network resources to be allocated. Fig. 4 depicts results of an access network dimensioning with typical values of P_{PL} and α, under assumption of offered traffic flow generated by homogeneous ON-OFF sources (without CNG feature). A comparative study has been carried out by applying both the model presented in [5], expressed by (20) (we will refer to it as "model A"), and proposed analytical model (19) (we will refer to it as "model B"). The "model A" corresponds to a system that encompasses the set of states for which $i = N$ is fulfilled. This is related to the most right column on the state-transition diagram (Fig. 2). Since network service providers are interested in more realistic system performance evaluation, it is necessary all possible system states to be taken into consideration (we can denote this condition as $i > n$). Numerical results show that this led to more efficient network resource (bandwidth) usage, because the system is not overdimensioned, as it is done by using (20).

For carrier-grade voice service it is of crucial importance a VoIP dimensioning framework for accurate estimation of the network resource, required to guarantee the performance bounds of aggregated GVoIP traffic sources, to be applied. This issue is addressed by the proposed analytical model (21) and (22), which is valid for any VoIP coding scheme, by setting both parameters α ($\alpha = 1$ corresponds to a CBR-type codec) and γ ($\gamma = 0$ – the model is valid for a VAD codec without CNG feature). We compare our dimensioning algorithm to the common approach of considering $i = N$. Comparative study includes the derived exact formula (14) for packet loss evaluation. We consider G.729 and G.723 coding schemes, both employing VAD and CNG features. Typical values of γ for both codecs are assumed to be 0.1 and 0.03 respectively [16]. Results depicted on Fig. 5 reveal the bandwidth (in term of transmission resource units) required in order to satisfy QoS constrains of aggregated traffic load A_c. It is demonstrated the bandwidth allocation margin that results from applying the proposed methodology for packet loss evaluation (22) ($i > n$).

On the other hand, silence suppression technology can considerably decrease the bandwidth usage needed. This is quantitatively represented on Fig. 6. Study results demonstrate that the same amount of network resource could be allocated to meet the call flow demands with higher value of activity factor ($\alpha = 0.45$) when the proposed approach is applied, compared to the case for which $i = N$.

Since we are interested in the voice (ACT) packets loss probability evaluation, in our analysis, we ignore the SID packet losses which are a part of the overall packet flow. This is represented by (22). On the other hand, SID packets are characterized with a small number and size (typically $\gamma<0.1$), which means they would not have great impact on the precision of voice packets loss probability evaluation, expressed by (21). The results presented on both Fig. 7 and Fig. 8 let us answer this question.

Fig. 7 depicts the comparison results of P_{PL} and P_{PL}^{ACT} versus transmission resource units n, for different values of offered traffic volume A_c and codec used. In order to get more accurate results, Fig. 8 shows the absolute difference $\Delta p = \left| P_{PL}^{ACT} - P_{PL} \right|$ of (21) and (22) for the same case. It could be concluded that there is no sense of using (22) instead of (21), when the offered traffic volume is high (above a certain threshold) as well as it is expected the packet loss probability is small enough (e.g., less than 1.10^{-3}).

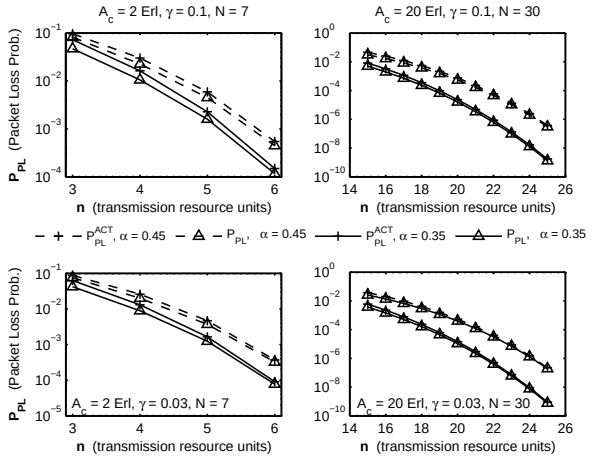


Figure 7. Comparative study of packet loss probability evaluation

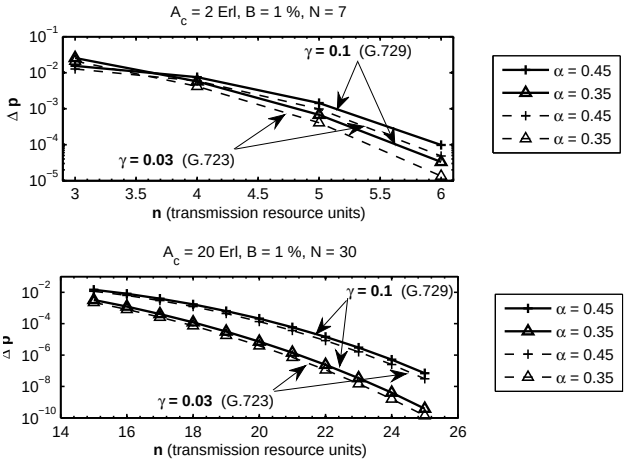


Figure 8. Comparative study of packet loss probability evaluation in terms of absolute difference (Δp)

The current trend of wireless networks dimensioning and performance evaluation is going towards application-specific quality measures, which takes into consideration the end user's satisfaction, rather than network parameters for QoS. This allows us to build a CAC mechanism that is capable of allocating the scarce network resource more precisely, based on the users' perceived QoS. The admission decision policy is codec-dependent and relies on the MOS value, which is expressed by the average impairment factor (26). We consider the following modern low-bit-rate coding schemes, supporting VAD and CNG: G.729, G.723.1, and the AMR codec (the highest mode – H, and the lowest – L), which has been adopted in the 3GPP networks. It is assumed the AMR codec maintains the either mode for an active call duration. The codec parameters for different values of N_{fpp} are calculated according to [16] and presented in Table I. The IP packets flow rate during a voice source active and inactive state is denoted by R and R_{off} respectively. A sequence of N_{fpp} consecutive codec frames (either ACT or SID) are sent in a single IP packet payload every $N_{fpp} \cdot T$ seconds, where the value T is codec-dependent and denotes the frame inter-arrival time.

The feature under investigation is towards a CAC mechanism dimensioning of an access point. We assume negligible packet delay ($I_d \approx 0$), which does not have a direct impact on the perceived voice quality evaluation. In order to achieve minimum packetization delay the number of codec frames that will be encapsulated in each IP packet is set $N_{fpp} = 1$ by the voice application.

Fig. 9 reveals the bandwidth (in terms of transmission resource units) required for offered aggregated GVoIP traffic flows to have a certain value of MOS, when a voice coding scheme of particular type is involved. The CAC decision policy is based on the target MOS value as well as the offered traffic load by setting call-level blocking probability B = 1 %. More generally, Fig. 10 depicts the link capacity

TABLE I. CODEC PARAMETERS

Codec	**AMR(H)**	**AMR(L)**	**G.723.1**	**G.729**
α	0.469	0.469	0.471	0.456
$\mathbf{N_{fpp}}$	1			
$\mathbf{N_{fpp} \cdot T}$ (ms)	20	29	30	10
R (bps)	19 600	20 800	17 067	40 000
$\mathbf{R_{off}}$ (bps)	1 662	2 400	900	4 583
γ	0.0852	0.1154	0.0527	0.1145
$\mathbf{N_{fpp}}$	2			
$\mathbf{N_{fpp} \cdot T}$ (ms)	40	58	60	20
R (bps)	14 069	12 800	11 733	24 000
$\mathbf{R_{off}}$ (bps)	1 662	2 410	789	4 583
γ	0.1181	0.1883	0.0672	0.191
$\mathbf{N_{fpp}}$	3			
$\mathbf{N_{fpp} \cdot T}$ (ms)	60	87	90	30
R (bps)	12 230	10 133	9 956	18 667
$\mathbf{R_{off}}$ (bps)	1 662	2 410	752	4 583
γ	0.1359	0.2378	0.0755	0.2456

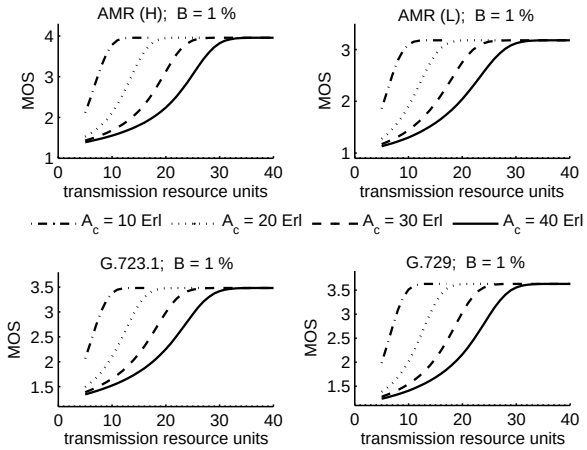


Figure 9. CAC dimensioning example – a common case

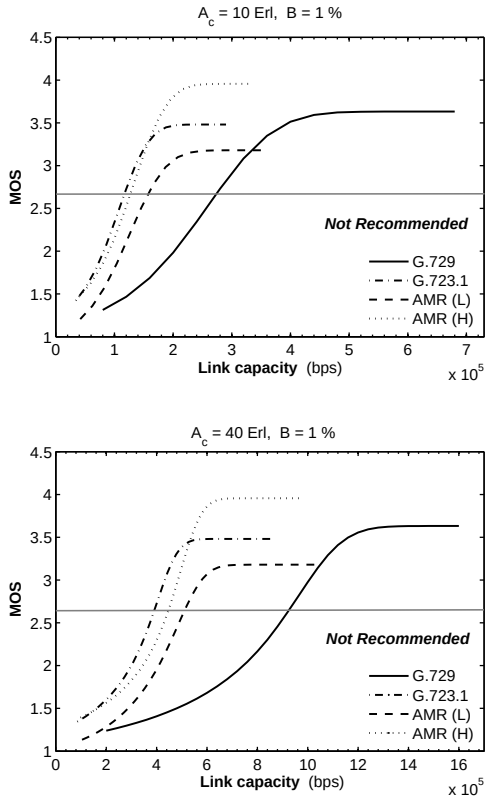


Figure 10. CAC dimensioning example for different coding schemes and GVoIP aggregated traffic load

that shall be guaranteed by an access point in order to reach certain perceived quality level. MOS rating is based on [22]. The method proposed could be incorporated into a codec negotiation procedure based on the allocated network resource and fulfilling the CAC decision criteria.

The VoIP traffic service over WAcN may encounter certain difficulties as a result of the wireless domain characteristics. The scarce radio resource may be wasted due to the traffic flow formed by packets with relative long headers, which could exceed the voice payload field several times. In most cases, this drawback can be minimized by adjusting the sequence of N_{fpp} consecutive codec frames building an IP packet payload field. From QoS perspective, this option leads to increase of packetization delay and it is not very appropriate, especially in case the voice packets could encounter considerable network delay. Applying the "new paradigm" of CAC dimensioning, Fig. 11 depicts the perceived voice quality, which could be achieved, under specific network parameters for a number of popular low-bit-rate voice codecs. It is assumed the aggregated GVoIP traffic load A_c = 40 Erl. It can be seen that for the same amount of consecutive voice frames per packet N_{fpp}, the access point shall allocate more resource when the average network delay d_n increases. This arises from the necessity of keeping the perceived voice quality at the same level when network conditions are getting worse. At the same time, the scarce resource is not wasted when the maximum possible value of MOS is requested to be obtained, since the capacity allocation to links could be accurately estimated by the model we propose.

V. Conclusion and Future Work

In this article, we have focused on a more realistic case where VoIP source traffic pattern includes not only a silence suppression feature, but comfort noise generation as well. The adoption of the traditional ON-OFF model may cause significant errors in estimating the bandwidth required in order to meet the performance bounds of aggregated VoIP traffic flow. Both call- and talk-spurt level considerations for a teletraffic design of access networks for voice services are proposed and comparative analysis has been carried out.

We propose a new paradigm of CAC dimensioning of wireless access networks serving packetized voice traffic. The new methodology has a broad area of application and is especially suitable for accurate link capacity estimation, taking into consideration the end user's service satisfaction, rather than network parameters for QoS treated separately. The approach is valid for a number of modern low-bit-rate voice codecs, employing VAD and CNG functionality, and is insensitive to the wireless technology in use.

Due to the great interest in emerging wireless access technologies, the research community is being interested in the system design, optimization and QoS requirements satisfaction of next-generation wireless access networks. The wireless environment features as well as user mobility draw the direction of the future research work. It will encompass the models we have developed and solve wireless communications resource management problems in order to maintain channel capacity and provide the performance guarantee. In order to achieve this goal, cross-layer adaptation and optimization mechanisms are going to be developed.

Acknowledgment

This article is a part of research work in the context of research project "Methods and models for resource management in convergent networks", funded by the Research and Development Sector of the Technical University of Sofia (RDS-TUS) under grant 102ni107-7.

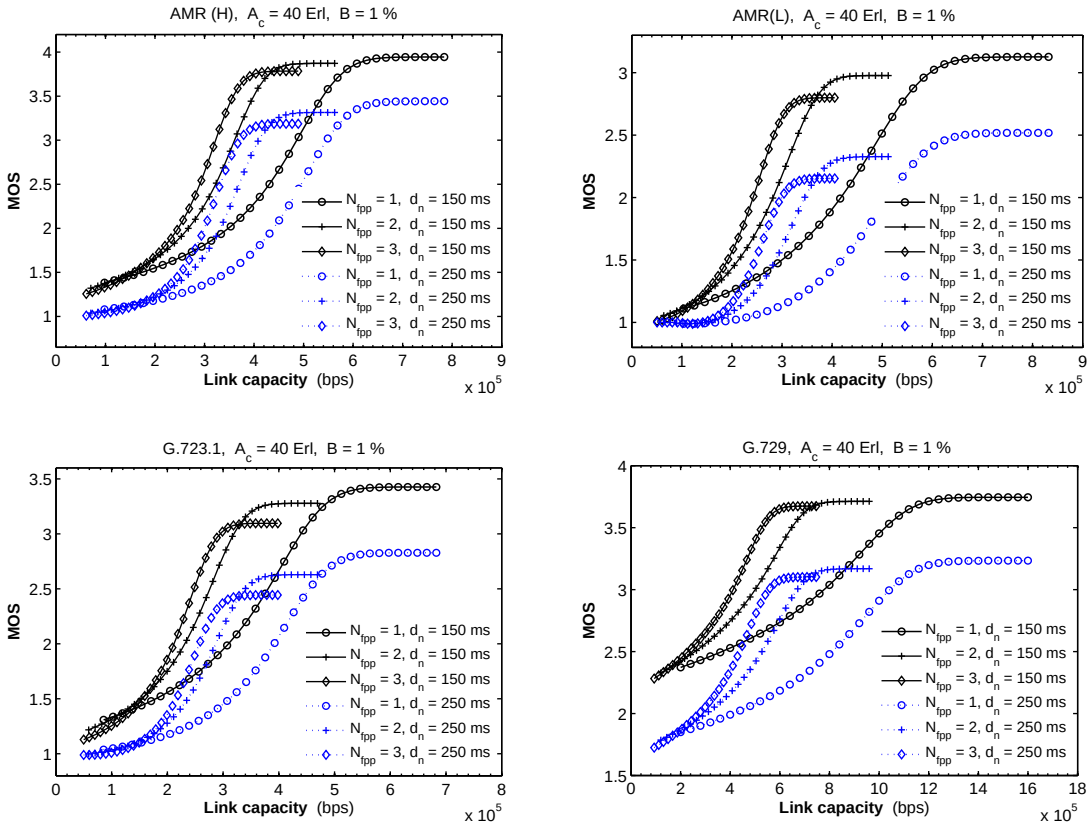


Figure 11. CAC dimensioning example for a number of low-bit-rate voice codecs, expressed in N_{fpp} and d_n

REFERENCES

[1] K. Kassev, Y. Mihov, A. Kalaydzhieva, and B. Tsankov, "A new paradigm of CAC dimensioning for VoIP traffic over wireless access networks," Proc. The Fourth International Conference on Digital Society (ICDS 2010), Feb. 2010, pp. 54-59.

[2] M. Chatterjee and S. Sengupta, "VoIP over WiMAX," in WiMAX Applications, S. Anshon and M. Ilyas, Eds. CRC Press, 2008, pp. 55-76.

[3] H. G. Perros and K. M. Elsayed. "Call admission control schemes: a review," IEEE Communications Magazine, vol. 34, no. 11, Nov. 1996, pp. 82-91.

[4] M. H. Ahmed, "Call admission control in wireless networks: A comprehensive survey," *IEEE Communications Surveys*, vol. 7, no. 1, 2005, pp. 50 – 69.

[5] J. M. Pitts and J. A. Schormans, Introduction to IP and ATM Design and Performance. Chichester, England: John Wiley & Sons, 2000.

[6] S. Shin and H. Schulzrinne, "Measurement and analysis of the VoIP capacity in IEEE 802.11 WLAN," IEEE Transactions on Mobile Computing, vol. 8, no. 9, pp. 1265-1279, Sept. 2009.

[7] A. Chan and S. C. Liew, "Performance of VoIP over multiple co-located IEEE 802.11 wireless LAN," IEEE Transactions on Mobile Computing, vol. 8, no. 8, pp. 1063-1076, Aug., 2009.

[8] S. Ramprashad and C. Pepin, "A study of silence suppression and real speech patterns amd their impact on VoIP capacity in 802.11 networks," Proc. IEEE Int. Conf. on Multimedia and Expo, 2007, pp. 939-942.

[9] E. Hossain, Y. Xiao, Li-Chun Wang and K. K. Leung, "Radio resource Management and protocol engineering for IEEE 802.16," (special issue), IEEE Wireless Communications, vol. 14, pp. 2-51, Feb. 2007.

[10] S. Shrivastava and R. Vannithamby, "Group scheduling for improving VoIP capacity in IEEE 802.16e networks," IEEE 69st Vehicular Technology Conf., VTC Spring, 2009, pp. 1-5.

[11] J. Puttonen et al, "Voice-over-IP performance in UTRA Long Term Evolution downlink," IEEE Vehicular Technology Conf 2008, VTC Spring 2008, pp. 2502-2506.

[12] J. Zhu, X. She and L. Chen, "Complementary recourse allocation for variable-size VoIP packet in E-UTRA," IEEE Wireless Communications and Networking Conf. (WCNC 2009), 2009, pp.1-5.

[13] H. Wang, J. Han and S. Xu, "Performance of TTI bundling for VoIP in EUTRAN TDD mode," IEEE 69st Vehicular Technology Conf., VTC Spring, 2009, pp. 1-5.

[14] A. Baiocchi, N. Mellazzi and A. Roveri, "Queuing performance and control in ATM," 13th International Teletraffic Congress, Copenhagen, 1991, pp. 13-18.

[15] R. C. F. Tucker, "Accurate method for analysis of a packet speech multiplexer," Electronics Letters, vol. 19, no. 14, pp. 536-537.

[16] A. Estepa and R. Estepa, "Accurate resource estimation for homogeneous VoIP aggregated traffic", Computer Networks, vol. 52, no. 13, Sep. 2008, pp. 2505-2517.

[17] W. Wang, S. C. Liew and V. O. K. Li, "Solutions to performance problems in VoIP over a 802.11 wireless LAN," IEEE Trans. Vehicular Techn., vol. 54, pp. 366-384, Jan. 2005.

[18] A. Benyassine et al, "ITU-T Recommendation G.729 Annex B: A silence compression scheme for V.70 digital simultaneous voice and data applications," IEEE Communications Magazine, vol. 35, no. 9, Sep. 1997, pp. 64-73.

[19] B. Goode, "Voice over Internet protockol (VoIP)," Proceedings of IEEE, vol. 90, no. 9, pp. 1495-1517, Sep. 2002.

[20] R. G. Cole and J. Rosenbluth, "Voice over IP performance monitoring," ACM Comput. Commun., vol. 31, no. 2, pp. 9-24, April 2001.

[21] A. P. Markopoulou, F. A. Tobagi, and M. Karam, "Assessing the quality of voice communications over Internet backbone," IEEE/ACM Transactions on Networking, vol. 11, no.5, pp. 747-760,. Oct. 2003.

[22] The E-Model, A Computational Model for Use in Ttransmission Planning, ITU-T Rec. G.107, Int. Telecommun. Union, July 2000.

[23] N. T. Moura *et al.*, "MOS-Based Rate Adaptation for VoIP Sources," Proc. IEEE international Conference on Communications (ICC2007), June.2007, pp. 628-633.

[24] J. Matta, C. Pepin, K. Lashkari, and R. Jain, "A Source and Channel Rate Adaptation Algorithm for AMR in VoIP Using the Emodel," Proc. 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video, 2003, pp. 92-99.

[25] M. T. Gardner, V. S. Frost, and D. W. Petr, "Using Optimization to Achieve Efficient Quality of Service in Voice over IP Networks," Proc. IEEE Performance, Computing, and Communications Conference, Apr.2003, pp. 475-480.

[26] K. Fujimoto, S. Ata and M. Murata, "Adaptive playout buffer algorithm for enhancing perceived quality of streaming applications," in Proc. IEEE Globecom 2002, Nov. 2002.

[27] L. Sun and E. Ifeachor, "Prediction of perceived conversational speech quality and effects of playout buffer algorithms," in Proc. IEEE Int. Conf. Communications ICC'03, Anchorage, AK, May 2003, pp. 1-6.

[28] L. Sun and E. Ifeachor, "Voice quality prediction models and their application in VoIP networks," IEEE Transactions on Multimedia, vol. 8, no. 4, pp. 809-820, Aug. 2006.

[29] C. Cicconetti, A. Erta, L. Lenzini and E. Mingozzi, "Performance evaluation of the IEEE 802.16 MAC for QoS support," IEEE Transactions on Mobile Computing, vol. 6, pp. 26-38, Jan. 2007.

[30] D. Zhao and X. Shen, "Performance of packet voice transmission using IEEE 802.16 protocol," IEEE Wireless Communications, vol. 13, pp. 44-51, Feb. 2007.

[31] S. E. Elayoubi and B. Fourestie, "Performance evaluation of admission control and adaptive modulation in OFDMA WiMAX systems," IEEE Transaction on Networking, vol. 16, no.5, pp. 1200-1211, Oct. 2008.

[32] ITU-T "One Way Transmission Time," Recommendation G.114, May 2000.

[33] R. J. Gibbens, F. P. Kelly and P. B. Key, "A decision-theoretic approach to call admission control in ATM networks," ," IEEE J. on Selected Areas in Communications, vol. 13, no. 6, pp. 1101-1114,. Aug. 1995.

[34] M. Reisslein, K. W. Ross and Rajagopal, "Guaranteeng statistical QoS to regulated traffic: the single node case," Proc. 18th, IEEE INFOCOM, 1999, pp. 1061-1072.

[35] G. Mao and D. Habibi, "Loss performance analysis for heterogeneous ON-OFF sources with application to connection admission control," IEEE/ACM Transactions on Networking, vol. 10, no. 1, pp. 125-138,. Feb. 2002.

[36] A. Estepa and R. Estepa, "Accurete VoIP dimensioning for WAN links," Electronics Letters, vol. 43, no. 23. Nov. 2007.

[37] A. Estepa, R. Estepa and J. Vozmediano, "A new approach for VoIP traffic characterization," IEEE Communications Letters, vol. 8, no. 10, pp. 644-646, Oct. 2004.

[38] A. Estepa, R. Estepa, I. Campos, and A. Delgado, "Dimensioning aggregated voice traffic in MPLS nodes," Proc. of ONDM 2008, IFIP 2008, pp. 211-215.

[39] A. Mukherjee, "On the dynamics and significance of low frequency components of Internet load," Technical Report – University of Pennsylvania, 1992. (available at http://repository.upenn.edu/cis_reports/300). Last access: 01.2011.

[40] L. Sun, "Speech quality prediction for voice over Internet protocol networks," Ph.D. dissertation, University of Plymouth, U.K., Jan. 2004. (available at http://www.tech.plym.ac.uk/spmc/people/lfsun/publications/LSunPhDthesis.pdf). Last access: 01.2011.

[41] V. Frost, B. Melamed, "Traffic modeling for telecommunications networks," IEEE Communications Magazine, vol. 32, no. 3, Mar. 1994, pp. 70-81.

[42] A. Adas, "Traffic models in broadband networks," IEEE Communications Magazine, vol. 35, no. 7, Jul. 1997, pp. 82-89.

Model-based Performance Anticipation in Multi-tier Autonomic Systems: Methodology and Experiments[1]

Nabila Salmi*‡, Bruno Dillenseger*, Ahmed Harbaoui*†, and Jean-Marc Vincent†
* Orange Labs, Grenoble, France
Email: bruno.dillenseger@orange-ftgroup.com, nabila.salmi213@gmail.com
† LIG, MESCAL Project, Grenoble, France ‡ LISTIC, University of Savoie, Annecy, France
Email: {ahmed.harbaoui, jean-marc.vincent}@imag.fr

***Abstract*—This paper advocates for the introduction of performance awareness in autonomic systems. Our goal is to introduce performance prediction of a possible target configuration when a self-* feature is planning a system reconfiguration. We propose a global and partially automated process based on queues and queuing networks modelling. This process includes decomposing a distributed application into black boxes, identifying the queue model for each black box and assembling these models into a queuing network according to the candidate target configuration. Finally, performance prediction is performed either through simulation or analysis. This paper sketches the global process and focuses on the black box model identification step. This step is automated thanks to a load testing platform enhanced with a workload control loop. Model identification is based on statistical tests. The identified models are then used in performance prediction of autonomic system configurations. This paper describes the whole process through a practical experiment with a multi-tier application.**

***Keywords*-Autonomic systems; performance; automatic modelling; queuing network model; load injection**

I. INTRODUCTION

A. Autonomic computing and performance management

Management of modern distributed systems is becoming increasingly complex and costly. Autonomic computing typically addresses this issue by providing systems with self-management capabilities. A common approach to building self-managing systems has been sketched by [1], through the well-known MAPE-K control loop (Monitor, Analyze, Plan, Execute - Knowledge): some self-* features (e.g., optimization, configuration, healing and protection) are implemented in the system in the form of feedback loops that result in system reconfiguration plans to be executed when special undesired situations are met. Reconfigurations typically result in removing, adding or replacing one or several system constituents, thus resulting in a new configuration. Here, we consider changing constituent parameters (e.g., tuning) as a component replacement inasmuch its behavior changes, especially from a performance point of view.

Reconfiguring a distributed application may result in performance changes, ranging from anecdotal to dramatic. In the case of critical or Service Level Agreement-ruled systems, it may be quite relevant to evaluate the performance of a candidate new configuration before actually deploying it. This remark applies to any self-* feature-driven reconfiguration, but it particularly applies to self-optimization. This sort of feature may typically compare different candidate configurations, looking for an optimized trade-off between an expected performance level and operational constraints and costs.

This paper deals with the introduction of a strong performance-awareness in autonomic systems, in order to drive the Analyze step of the MAPE-K loop with relevant performance Knowledge, combined with performance analysis or simulation capabilities. To do this, our approach consists in relying on performance models of a distributed application's constituents, and then composing these models according to interactions between constituents, to get a performance prediction of an application configuration. We typically address distributed applications where some constituents may be replicated in order to increase the overall application performance (e.g., multi-tier web applications). In this introduction, we sketch the global process, as presented in [2].

B. Identifying black boxes

The first roadblock we meet is getting the constituents' performance models. Applications, middleware and systems based on common information technologies typically come with poor performance-related specification, if any. At a certain granularity, the inner architecture of some constituents is either so complex or under-specified that trying to infer a performance model for each one would practically take far too much effort. However, a certain granularity of decomposition seems to be humanly affordable, at least for distributed applications. For instance, an HTTP front-end, an EJB container and a database is a straightforward level of decomposition in the context of multi-tier Java EE applications. Based on this, our approach is two-fold:

1) decompose a distributed application into constituents, called *black boxes*, with a relevant granularity,
2) automatically get a performance model of each black box through an experimental stimulus-response observation principle.

The relevant granularity level is a trade-off between the decomposition feasibility (with regard to available information and complexity) and the final model accuracy and sizing opportunities. The major criterion is sizing opportunity: if one sub-element of a black box can be replicated to increase the workload capacity of the sub-feature it supports, then there is

[1]This work is supported by the French ANR, through the Selfware and SelfXL projects, and ANRT.

a big motivation in decomposing this black box into sub-black boxes. Accuracy is another motivation for decomposition, since a queuing network model will be closer to reality than a single queue model representing the same element. Last, the black boxes model identification process may be quicker and simpler, for smaller black boxes, and may have less weird behaviors than for bigger ones.

A black box is a constituent whose content is unknown. You may only know its external interfaces and be able to invoke their operations, and observe the outputs resulting from your invocations. This black-box may (or may not) provide an interface to give some information about its state. It runs in an execution environment whose resources usage may be observed (CPU, RAM, network bandwidth, etc.). We particularly address software black boxes running on an operating system. Commercial, off-the-shelf software elements, as well as complex open source middleware, would be typically black boxes. In case of distributed software, network interactions give decomposition opportunities.

C. Automatic model identification

Once we have decomposed the global system into black boxes, we need to get a performance model for each of them, and then to combine these models into a single one representing the global system. To achieve this, we choose to model black boxes as queues, and the global system as a queuing network. The idea is that we can experimentally identify queuing models that best represent the performance of black boxes, and then build the resulting queuing network for performance prediction. Model identification is based on non-parametric statistical tests. This enables to determine the best distributions fitting service times and inter-arrival times.

The other idea is to get experiments on black boxes automatically performed by a load testing platform, enhanced with self-regulated load injection capabilities [3]. The workload is automatically adjusted according to measures and policies that define workload steps, levels and saturation criteria. There are three reasons for this step-by-step increasing workload injection. First, we have no knowledge and we make no assumption about the maximum capacity of each black box: we start with a minimal capacity assumption, and then we gradually increase the assumed capacity. Second, we prevent load injection from actually reaching a critical saturation level that would result in a black box crash, with a possible necessity to reboot and restart. Third, we want to observe the black box in permanent, stable states, which practically requires to have these steps.

This experimental process uses research results in terms of component-based architecture for building autonomic computing systems [4].

D. Performance prediction

Once the Knowledge part of our autonomic system is fed with the black boxes queuing models, the Analysis function of any self-* control loop is able to evaluate performances of possible target configurations. This prediction may be based on queuing network simulation or analysis. When several queuing model candidates have been successfully identified for a single black box, the actual model selection may be driven, on the one hand, by its accuracy, and, on the other hand, by its ability to be quickly analyzed or simulated. The more accurate is the model, the more difficult is the analysis (see Figure 1).

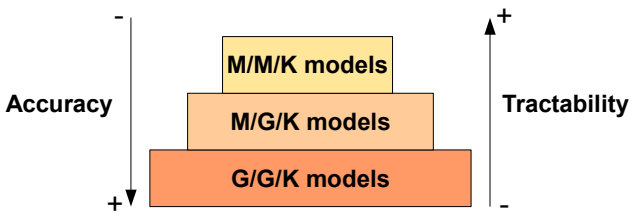


Figure 1. Accuracy versus tractability

As a matter of fact, the efficiency of this performance prediction influences the speed and effectiveness of the self-* control loop. The global process is summarized by Figure 2. This paper develops the second step (model identification) of the approach, as a continuation of [2]. An example of the use of identified queue models in performance prediction is given.

This paper is organized as follows: first, we position our work with other related work in Section II. Then, Section III describes how the self-regulated load injection process is achieved: we compute the duration of an injection period and explain how to estimate stabilization time, injection step duration, and sampling period. In Section IV, we detail the black box model identification process by first presenting inter-arrival and service sampling, and then by explaining how to determine the distribution shape and the whole identification process. We also present how to estimate, from the observed parallel processing level of a given black box, the corresponding queue model's number of servers. In Section V, we show how to use identified models in performance prediction. We show a practical application of our model identification process in Section VI on a typical use case and we give experimental results. Finally, we conclude in Section VII and give some open questions and perspectives.

II. Related work

Several works have been proposed to model systems for autonomic computing purposes. Some authors used regression models [5] for transactional systems, but most of them proposed queuing networks as predictive models [6], [7], [8], [9]. Kamara et al. [7] modelled a 3-tiers architecture with a single queue; Rafamantanantsoa et al. [8] described a simple web server with an M/G/1/K-PS queue model. The parameters of this model (queue capacity and mean service time) are estimated by the maximum likelihood technique, given data obtained by extensive experiments. Other proposals [9] used queuing networks instead of a single queue model. This last modelling seems to be more appropriate for distributed systems.

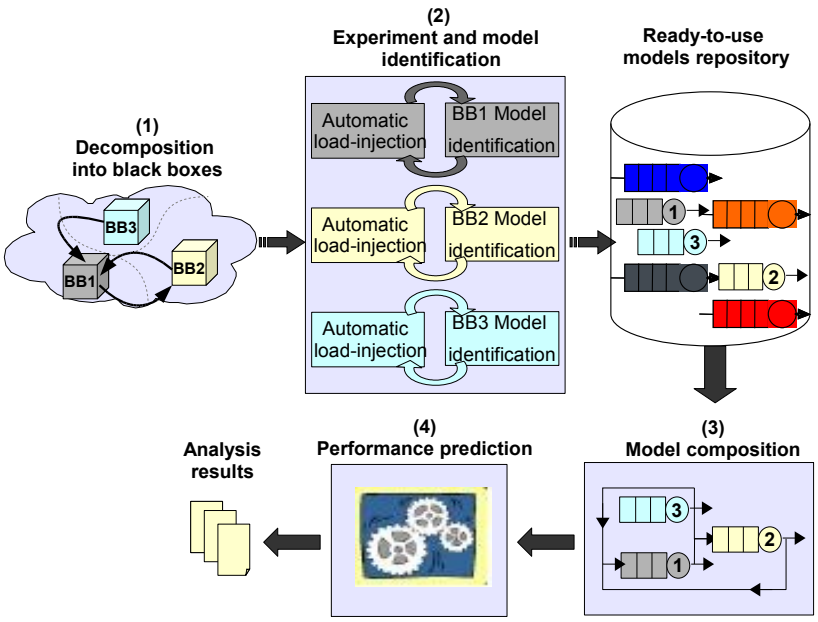


Figure 2. Performance prediction of a distributed application configuration

Begin et al. [10] approximate the measured behavior of a variety of systems by selecting and calibrating a limited set of queuing models. More recently, using a black box approach, Menasce [11] addresses the problem of finding an unknown subset of service demand parameters in queuing network models, given the known values and given the values of response times for all workloads.

Woodside, Zheng and Litoiu [12] worked also on tracking parameters of queuing network models for an autonomic system. They used extended Kalman filters, while integrating various kinds of measured data such as response times and utilization. Using Kalman filters is quite valuable since they are known to be predictor-corrector estimators: they make the obtained model optimal when dealing with error covariance minimization.

These proposals are interesting, however autonomic systems need to be dynamically analyzed with precision, to be able to choose the best solution when a problem occurs.

This fact led us to estimate an accurate queuing model, that might represent the observed system: we propose to model inter-arrival and service times, as well as the number of servers. We don't use for that Kalman filters because they are not suitable for our approach: first, Kalman filters are not sufficient in our case, since we estimate shapes of distributions. Second, convergence of these filters is not guaranteed for a number of queuing models, which makes their use without a predefined model more difficult in an autonomic approach. Rather, we can identify distributions with more precision, using non-parametric statistical tests. Most of our experiments show more Lognormal and other distributions than exponential distributions, which were used in most work.

Our approach is thus a generalization of previous methods proposed in literature. It also provides rich distributions modelling systems behavior, and giving more information. The final contribution of this paper is an implementation of the developed approach in a prototype for modelling multi-tier autonomic systems and anticipating their performances.

III. Self-regulated load injection

Our approach relies on injecting a step by step increasing workload (see Figure 3). To allow estimation of a coherent model, we inject a workload composed of a single traffic type. Basically, this consists in automating a benchmarker work, trying to find the performance limits of a system through load testing. It injects a first load level, observes the system behavior (response time, resource usage…) and decides the amount of the next workload step. It repeats the procedure until reaching - or more probably overpassing - a workload high limit, beyond which the system becomes unstable or the delivered quality of service is no more satisfactory.

To automate this process, we rely on a load injection framework: the CLIF [13] framework provides injectors, for generating a workload modelled as *virtual users* (vUsers) and measuring requests response times, and probes, for measuring usage of arbitrary computing or networking resources. Moreover, we need to define an injection policy specifying several parameters, mainly: the workload level in each step (injection step), the length of an injection period, the time required to get the system in a stable state (stabilization time), the System Under Test saturation limits where the load testing process must stop.

In the remainder of this paper, we use the Kendall notation [14] of an elementary queuing system, denoted by $T/X/K$ where T indicates the distribution of the inter-arrival times, X the service times distribution and K the number

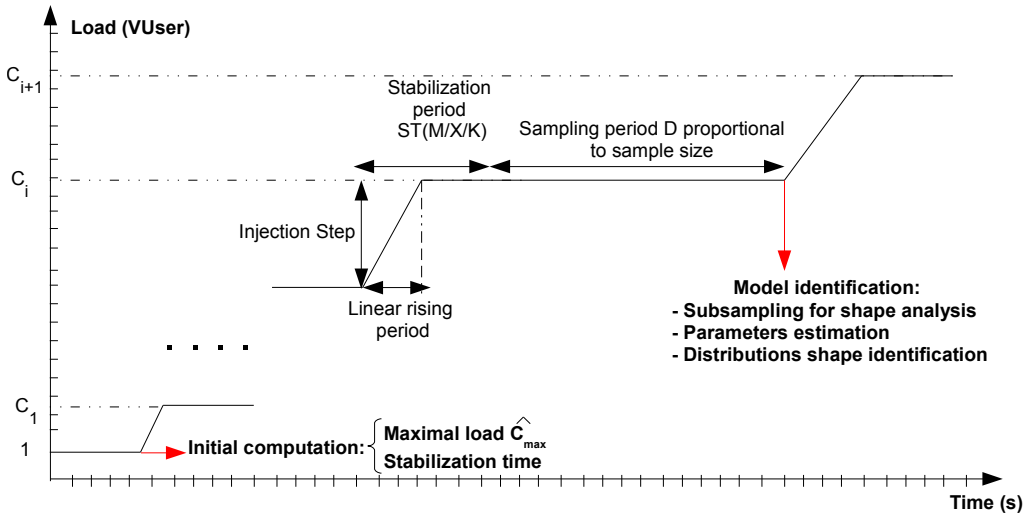


Figure 3. Model identification during a ramp of load injections

of servers ($K \geq 1$). Note the number of servers represents the observed parallel processing capability and not the actual number of physical computers or processors. This capability practically depends on the multi-threading support and on the computation profile (e.g., CPU-intensive or Input/Output-intensive). So, it is both hardware-dependent and software dependent (operating system, middleware, application). As a simplification, we still consider it as an integer number in this, but it is more likely a decimal number.

A. Injection policy

The main issue related to self-regulated injection is to determine automatically the injection policy parameters defining the steps of increasing workload (see Figure 3). These parameters are computed at runtime, step by step.

1) Estimation of maximal load: An initial load injection phase is undertaken to estimate the maximal supported load $\hat{C_{max}}$. In this phase, we load our system with markovian interarrivals requests of one virtual user. We collect response times and compute a first approximation of $\hat{C_{max}}$, as $\frac{1}{\mu}$, μ being the service rate. This result comes from the fact that, when dealing with one customer arriving in an empty queue (no concurrence), the mean waiting time is null ($\overline{W}$=0), leading to the following mean response time:

$$\overline{R} = \overline{W} + \overline{X} = \overline{X} = \frac{1}{\mu}$$

When the queue model is M/G/1, the arrival rate of requests converges to μ. An example of this convergence is depicted in Figure 4, obtained when experimenting our example. The value of $\hat{C_{max}}$ is experimentally corrected when the estimated number of servers K increases (see Section IV-D).

2) Injection step: The load injection step should be carefully defined, as a small step may result in a huge experimental time, whereas a big step may brutally saturate the system. We use an additive increase while checking if the experiment is close to the value of the estimated maximum load $\hat{C_{max}}$. The

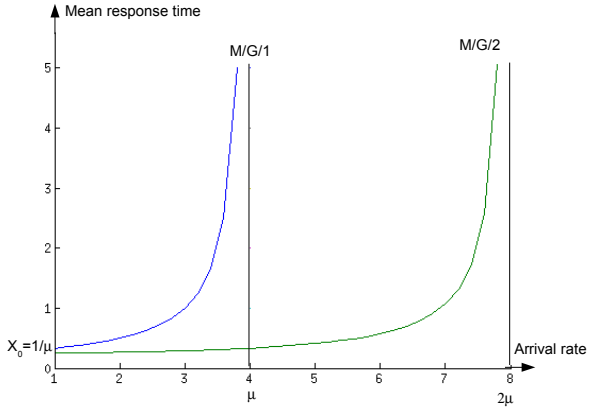


Figure 4. First estimation of maximal load Cmax

increment is defined through a decomposition of the estimated maximum workload into a user-set number of iterations. The greater this parameter is, the more workload steps will be performed, thus giving more accurate information, but taking much more execution time.

3) Rising period: After an injection phase, the load is increased with an increment and submitted. To avoid a malfunctioning of the system due to a big injection step, we choose to inject the increment of requests gradually, drawing thus a ramp. The rising period is the period during which the injection of a new load increment is done. In practice, injecting 10 vusers/sec is acceptable. The rising duration is then automatically computed as a function of the injection step, while maintaining 10 vusers/sec.

4) Sampling period: This should be computed such that the system behavior remains stationary and the sampling is sufficiently large to get good confidence in measures. Rai Jain in [15] proposes a formula for determining the sample size n required to achieve a given level of accuracy $r\%$ and a

confidence confidence interval of $100 * (1 - \alpha)\%$:

$$n = (\frac{100 z \sigma_n}{r \bar{m}})^2,$$

where z is the normal variate of the desired confidence level (for a 95% confidence interval, $z \approx 1.96$), $\bar{m}$ is the mean value of the parameter to estimate and σ_n stands for the sample standard deviation.

B. Estimation of the stabilization time

When collecting measures, it is important to distinguish the transient and stationary periods. The variance of measured data gives a first insight in system stability. This is not sufficient as there may be measurements peaks when some phenomenon like the garbage collector appears. Thus, a combination of theoretical and experimental methods is required.

We estimate the stabilization time at each injection step, as the convergence time of the Markov chain [14], [16] underlying the associated queue model. We restrict ourselves to Engset models.

In other words, the stabilization time ST is considered as the time required to get the equilibrium (stationary) state probabilities, denoted as the probability vector π, when the Markov chain is ergodic.

It can be computed by studying the transient behavior of the system. As the queue model of the step (i) is not yet defined, we rely on the queue model (denoted $model_{(i-1)}$) determined in the previous step (i-1). We compute ST by:

(1) deriving the transition probability matrix P of $model_{(i-1)}$, which has a dimension equal to MxM, M being the amount of load submitted in step (i);

(2) obtaining the probability vector:

$$\pi^{(n)} = \pi^{(n-1)} P = \pi^{(n-2)} P^2 = \ldots = \pi^{(0)} P^n,$$

$\pi^{(0)}$ being the initial vector and n the number of iterations required to reach the equilibrium state;

(3) computing the stabilization time ST as:

$$ST = \frac{n}{\lambda + \mu},$$

λ being the inter-arrival parameter of step (i) and μ is approximated by the service rate parameter of $model_{(i-1)}$.

As we base on $model_{(i-1)}$, which is not necessarily the model of step (i), we correct the obtained stabilization time by adding an error ϵ_i, computed experimentally by observing the variation coefficient of measures collected in step (i).

IV. Identification of the performance model of a black box

As previously said, a black box is modeled with a queuing model. Identification of such model requires to define the distribution of inter-arrival times, the distribution of service times and the number K of servers. The identification of these distributions requires first to capture adequate measures from the injection framework, then deduce a sample to analyze, i.e., an interarrival sample and a service sample. The obtained samples are submitted to statistical tests from which is estimated the corresponding distribution shape. Parameters of the identified distributions (interarrival and service) are also estimated to complete the definition of final models.

A. Inter-arrival sampling

This implies, for a distributed system, to identify the shape of the inter-arrival process received on upstream of each black box. The interarrival process of a black box depends, on one side, on the rate of submitting requests to the global system and, on the other side, on the system architecture. As a matter of fact, we investigate inter-arrival times distribution of a black box only when being in the context of a configuration of the system to which it belongs. To achieve that, load injections are submitted to the system and arrival times of requests are captured for each black box. For a given black box, we compute inter-arrival times, obtaining the sample T.

B. Service identification process

This process consists in submitting load injection to the black box (see Figure 3). The workload is increased through several steps, until reaching the maximal estimated load $\hat{C_{max}}$. As many theoretical results exist for the M/G/1 and M/G/K models, we choose to inject requests through exponential inter-arrival times, obtaining at worst a M/G/K model.

Let us detail this process. It is done through two major phases : an initialization phase and an identification phase.

1) Load injection steps: Two major steps are followed:

a) Phase 1: Initialization: This phase consists in submitting to the black box a flow of similar requests representing only one customer. We measure the response time mean, denoted $\overline{R_0}$, during the sampling period computed as explained in Section III-A4. As a single customer uses only one server, we can infer that the black box behavior follows the M/G/1 model in the worst case (see Section III-A1). Hence, the value of service rate during this phase, μ_0, is used to get the maximal estimated load C_{max} and the next injection step.

Note that a realistic traffic typically involves a mix of different kinds of requests, e.g. user connection and authentication, requests involving or not database read or write operations, etc. In fact, we could also address such heterogeneous traffics, as long as response times are of the same order of magnitude from one request kind to another. Our steps might last longer because of the higher service time variability, but the resulting average service time would be still representative of the traffic mix. We would assume that the mix is the same whatever the workload level. Experiments about this would be an interesting complement to our work.

b) Phase 2: Identification: This phase is carried out through several steps , where each step (i) consists of:

1) Submitting a self-regulated load injection C_i following a Poisson distribution.
2) Waiting for stabilization (stabilization time already computed as shown in Section III-B) and collecting experimental measures during the computed sampling period: response times, interarrival times and utilization of the black box for this workload step.

3) Inferring service times $(X_k)_{1\leq k\leq n}$ from the samples of response times $(R_k)_{1\leq k\leq n}$ and interarrival times $(t_k)_{1\leq k\leq n}$.
4) Removing aberrant values from the service time samples. This is done by removing a fixed percentage (for instance 5%) of greater values. These great values may be considered as experimental measurement errors, resulting, for instance, from by some phenomena such as the occurrence of garbage collector on the load injector.
5) Identifying the shape of the service times sample using statistical tests.
6) Computing the injection parameters for the next step: injection step and the stabilization time.

2) Stop condition based on saturation checking: During the load injection steps, it is necessary to test if the black box is getting saturated to stop the experiment. This is done by monitoring the black box state and detecting whether its utilization reaches some predefined limits. In our context, we define the black box utilization through computing resources utilization (CPU, memory, JVM heap memory). This is achieved by deploying a probe for each monitored resource. Load injection is stopped as soon as one or several resources get(s) saturated. Resource saturation is defined as reaching a given threshold, determined by an expert of the system.

When the black box reaches the estimated maximal load and no saturation appears, we correct the maximal load and continue load injection tests, and so on, until saturating the black box (see Section IV-D). This technique of reaching maximal load level allows us to capture all possible behavior of the box against all possible load levels. Hence, the obtained model is the closest and the best one fitting the service offered by the black box per load level.

3) Service sampling: To obtain the service sample of an injection step, the load injection framework delivers several measures. We use mainly response times $(R_k)_{1\leq k\leq n}$, interarrivals $(t_k)_{1\leq k\leq n}$ and utilization U of all resources. We need also to estimate the scheduling policy to be able to compute the service sample. So, in a first time, we assume that the black box relies on a process sharing (PS) scheduling policy, then when getting close to saturation, the scheduling policy becomes FIFO. In both cases, service times $(X_k)_{1\leq k\leq n}$ are computed as follows:

1) For a PS policy:
$$X_k = R_k * (1 - \lambda * \overline{X}) \text{ [14]}$$
where λ is the interarrival rate used during the load injection and $\overline{X}$ is estimated with the fix point algorithm using the estimated $\overline{X}$ of the previous step as an initial value.
2) For a FIFO policy: We use an extended result relating service times $(X_k)_{1\leq k\leq n}$, response times $(R_k)_{1\leq k\leq n}$ and interarrival times $(t_k)_{1\leq k\leq n}$ [14]:
$$R_k = [R_{k-1} - t_k]^+ + X_k$$
This result is valid for a model using one server and a FIFO policy. So, if we get to identify, in step (i), a model characterized by K servers (K>1), this result cannot be used. To generalize this result, we propose to use a similar result:
$$R_k = [R_j - t_{j,k}]^+ + X_k$$
where j corresponds to the previous request that quitted the server, which served the k^{th} request, and $t_{j,k}$ is the interarrival between the j^{th} and k^{th} requests. The j^{th} request is determined by recomputing iteratively service times $(X_k)_{1\leq k\leq n}$, beginning from the first served request and using the R_j and $t_{j,k}$ computed from collected measures.

C. Distribution shape identification

To automatically determine the shape of inter-arrival times and service times distributions, we use a statistical test based approach, which selects the distribution that fits well the samples.

1) Identification using statistical tests: The statistical goodness-of-fit hypothesis test is a process that consists of making statistical decisions using experimental data. Several hypothesis testing approaches exist. In our case, we use the Kolmogorov-Smirnov statistical test [17], since it is appropriate to continuous distributions. We also use the Anderson-Darling test, which is appropriate to distributions with heavy queue.

However, these tests are only suitable for small samples and cannot apply to large samples. To avoid this drawback, we uniformly select a sub-sample from our data, on which we perform the test. Distributions that give a p-value (output value of a statistical test) greater than 0.1 are selected as good distribution representatives for our measures.

2) Estimating distributions shape: To seek the most appropriate distribution fitting an inter-arrival/service sample, we test several distribution families, known in the literature as distributions appearing naturally in computing systems [18]: exponential family, heavy-tail distribution family, etc.

We begin by choosing a distribution from the exponential family. We estimate parameters of each distribution using a *Maximum likelihood* estimator. We then keep distributions that give a p-value greater than 0.1.

To achieve that, we compute the variation coefficient CV^2 of the sample and its confidence interval. Depending on its value, we test a set of distributions. If the confidence interval of CV^2 contains 1, we test the exponential distribution. If $CV^2 \in]0,1[$, we test the hypoexponential(k) distribution, the Erlang(k) distribution and the gamma distribution. If $CV^2 \in]1,+\infty[$, we test the hyperexponential(k) distribution, the Uniform, the Normal, Lognormal, and Weibull distributions.

For each distribution d, we analyze a sample S as follows:

- If necessary, we make transformations (for instance a shift) on the sample $(S_k)_{1\leq k\leq n}$ to fit distribution d,
- We estimate parameters of d with a *Maximum likelihood* estimator.
- We choose a small sample S^* from $(S_k)_{1\leq k\leq n}$.
- We perform a statistical test for S^* and d with estimated parameters. As previously said, we choose to work with the Kolmogorov-Smirnov test. We repeat several times

this statistical test, and take the mean of obtained p-values, so as to ensure a correct p-value result.
- We then discard distributions whose statistical test gives a p-value less than 0.1. The set of remaining distributions, denoted L, is considered as the possible behavior of the black box service, resulting in a black box model M/X/K specified by several service distributions.

D. Estimating the number of servers

The number of servers (parallel processing capability, see section III) observed for a black box is determined experimentally. When reaching the estimated maximal load $\hat{C_{max}}$, we observe the black box utilization. If it indicates the black box is saturated, the number of servers remains 1. Otherwise, we progressively (step-by step) increase the load and check if saturation is reached. If no, we correct the estimated maximal load $C_{max} = k * \hat{C_{max}}$ and increment the assumed number of servers by 1. We resubmit new increasing load injections. We observe again the black box utilization and repeat the procedure until reaching saturation.

If during step (i), we identify a number of servers K>1, we need to correct models of previous steps, so we repeat samples analysis of these previous steps, by recomputing the service sample and re-identifying the models of each step.

Note that the estimation of servers number representing a black box is done independently from distributions shape estimation. The final estimated number is deduced at the end of the step-by-step process (at saturation), while shape distributions are estimated at each injection step.

E. Validation of the black box queue model

The identification process produces one or several candidate queue models possibly corresponding to different load levels. These models are validated by comparing empirical performance measures with theoretical ones, typically mean response time, mean waiting time and throughput.

V. Using identified models in performance prediction of autonomic systems

Let us consider a system configuration C as a possible solution for ensuring an autonomic feature. The goal is to be able to evaluate performances of C before its application on the system.

To reach this objective, the first step is to feed the autonomic system with its black boxes queuing models, following the identification process sketched in Section IV: a model repository for the system is hence created. Then, the global model of the configuration C is built, so that to launch the Analysis function of any self-* control loop and predict performances of C. This is done by picking from the model repository and by composing them.

A. Composition of black box models

A queuing network is entirely defined by the number of its nodes, the parameters of each node (queue) and the routing probabilities between nodes (probability that a request is transferred to the j^{th} node after service completion at the i^{th} node). To compose the set of black boxes models, it is important, in one side, to get the topological structure of the interconnection, and in the other side, to describe transitions between the models in this topology.

1) Transitions between black boxes models: To compute the routing probabilities between nodes, we rely on traces of incoming and outgoing traffic of each node. We propose so to conduct a typical experimentation, during which we capture input and output requests of each black box. The capture is done using log files and is specific to each software product. The number of outgoing requests of each black box is deduced from this capture and so are the incoming requests to the corresponding black box addressees (notice that common log files give generally for each incoming request the sender address). A ratio of the traffic distribution between the black boxes is then computed, resulting in the definition of routing probabilities.

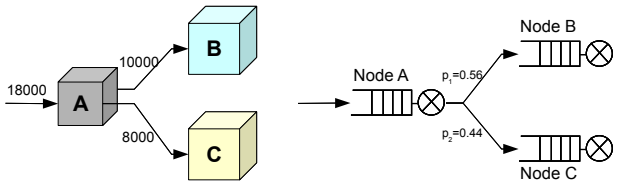


Figure 5. A black box interconnection example

Let us take an example of a system S (Figure 5), made of three black boxes A, B and C, where A is linked to B and C and each of B and C are only linked to A. In this case, we compute the number of outgoing requests of A and the number of incoming requests of each of B and C. Let us say there are 18000 outgoing requests for A, 10000 incoming requests for B and 8000 incoming requests for C. The routing probabilities are $p_1 = 10000/18000 = 0.56$ between A and B and $p_2 = 8000/18000 = 0.44$ between A and C.

2) Building the global model of a system: Once the transition probabilities between black boxes queues are obtained, we compose the queues and get a queuing network, the global model of the system. In our models, we deal only with open networks, as we study distributed infrastructures where requests are received and leave the system after service processing completion.

B. Analysis of the global model

To predict performances of the configuration C, we solve the obtained queuing network model using a specific algorithm allowing the computation of theoretical performance parameters. Typical parameters are:
(i) for the whole system: mean response time $\overline{R}$, throughput D and mean customer number $\overline{N}$;
(ii) for each queue Q_i: mean response time $\overline{R_i}$, mean number of customers $\overline{N_i}$ and utilization U_i.

The resolution algorithm to use depends on the complexity of queues composing the whole model :

- If the queuing network is composed of only M/M/K models, the exact MVA (Mean Value Analysis) algorithm is the most suitable to use [14], [19]. This algorithm is suitable for many systems as the markovian distributions are known in the literature to appear naturally in various systems [18].
  The MVA method allows to compute the mean values of parameters of interest such as the mean waiting time, throughput and the mean customer number at each node. Another algorithm to use is the AMVA algorithm [14], [19], which is an approximation improving the computation time of MVA.
- In other cases and depending on the structure of the resulting queuing network, we use the appropriate algorithm such as the method of Raymond Marie [20], [21]. This algorithm has been defined as an approximate solution for studying the asymptotic behavior of a network of queues with a general service distribution. When the network is composed of different types of queues, we propose to compute performance bounds. In the worst case, when analysis is impossible, we use simulation to determine global performances.

VI. Illustration

The automatic model identification process is currently implemented as a framework prototype. This framework provides: (i) an automated benchmarking controller, based on CLIF [13], for performing the self-regulated load injection steps, (ii) a model identification tool, based on Matlab/R statistical environments [22], [23] and, (iii) an editor for composing identified queue models and launching analysis/simulation of obtained queuing networks for performance prediction.

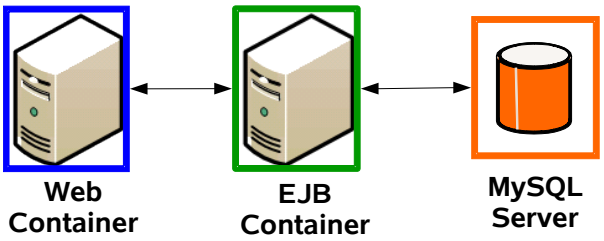


Figure 6. Use case: SampleCluster JONAS application

To illustrate the steps of our identification process, we experimented a three-tiers Java EE application, called *SampleCluster*, that runs in the *JONAS* application server, an open source Java EE implementation developed by the OW2 consortium [24]. This application was developed as a testing application of a JONAS cluster. This cluster is composed of a Tomcat web container server, an Enterprise Java Beans (EJB 3.0) container and a MySQL database storing EJB sessions information (see Figure 6).

The system is decomposed into three black boxes: the Web container tier, the EJB container tier and the database tier. This first level decomposition matches exactly the multi-tier architecture and this is very useful to operate an adequate and good system sizing. These black boxes are modeled using our automatic identification process. To inject requests to a black box, we use CLIF load injectors. We use a load injection scenario featuring virtual users whose behaviors represent real user behaviors. Network latency is considered as negligible for these experiments, since we operate with a high-speed local area network (Gigabit Ethernet), whose latency order of magnitude is microseconds. To get resources utilization measures and detect system saturation, CLIF probes have been used for CPU, JVM heap memory and RAM. We defined the black box saturation limits as 80% for the CPU (high limit), 80% for RAM (high limit) and 5% for the JVM's free heap memory (low limit).

To model a black box, it must be isolated from the other black boxes. Isolating the database tier is straightforward, since it is the last tier and it does not call any other black box. Isolating the Web and EJB tiers requires more work:

- either develop respectively an RMI plug and an SQL plug, i.e., a fake RMI or SQL server that accepts requests and give responses in place of the real server, in deterministic response time that can be subtracted from the black box measured response times. This technique requires some non-trivial programming, since responses must hold correct information. A record-and-replay solution may be applied, by observing requests and responses with the real server and then replaying known responses on known requests with the plug. This is not too complex to achieve with text-based protocols (such as SQL) when no socket secure layer is used (cf. encryption), but it is much more complex with binary protocols like RMI. Moreover, plugs must be benchmarked in order to know its response time and to be able to compute the black box service times;
- or follow a step-by-step approach, starting the modeling process from the downstream black box (the database black box in our example). In next step, we run the modeling process on the previous black box (EJB container) and, thanks to the model obtained during the first step, we subtract response times of the last box to the measured response times in order to be able to compute the corresponding real service times. Then, we iterate for next steps until the first tier is reached.

We used the second solution (step-by-step) for practical reasons: not only do we avoid developing a plug, but we also avoid benchmarking the plug, while in the step-by-step solution, next black boxes are benchmarked de facto.

In the following, we present modelling results for the three black boxes, then we give performance prediction of the system and an example of fulfilling an autonomic feature.

A. Modeling the database black box

The database black box runs on a Linux server with two 1.4 GHz PIII processors, and 1 GByte of RAM. We used a Linux

Load	Identified Model(s)	Parameters
1	$M/\Gamma/4$, **M/LN/4**, M/Norm/4, M/Wbl/4	m=-8.010, sigma=0.089
6	$M/\Gamma/4$, **M/LN/4**, M/Norm/4, M/Wbl/4	m=-8.034, sigma=0.276
11	$M/\Gamma/4$, **M/LN/4**, M/Norm/4, M/Wbl/4	m=-8.051, sigma=0.333
16	$M/\Gamma/4$, **M/LN/4**, M/Norm/4, M/Wbl/4	m=-8.020, sigma=0.396
21	$M/\Gamma/4$, **M/LN/4**, M/Norm/4, M/Wbl/4	m=-8.030, sigma=0.421
26	$M/Hr_2/4$, $M/\Gamma/4$, **M/LN/4**, M/Norm/4, M/Wbl/4	m=-8.053, sigma=0.428
31	$M/Hr_2/4$, $M/\Gamma/4$, **M/LN/4**, M/Norm/4, M/Wbl/4	m=-8.033, sigma=0.464
36	$M/Hr_2/4$, $M/\Gamma/4$, **M/LN/4**, M/Norm/4, M/Wbl/4	m=-8.074, sigma=0.476
45	M/M/4, $M/Hr_2/4$	p=0.055,$mu_1 = 0.945$,$mu_2 = 465.9$
54	M/M/4, $M/Hr_2/4$	p=0.036,$mu_1 = 0.964$,$mu_2 = 239.3$
63	M/M/4, $M/Hr_2/4$	p=0.064,$mu_1 = 0.936$,$mu_2 = 453.0$
72	M/M/4, $M/Hr_2/4$	p=0.060,$mu_1 = 0.940$,$mu_2 = 419.5$
86	M/M/4, $M/Hr_2/4$	p=0.062,$mu_1 = 0.938$,$mu_2 = 452.0$
100	M/M/4, $M/Hr_2/4$	p=0.065,$mu_1 = 0.935$,$mu_2 = 510.8$
119	M/M/4, $M/Hr_2/4$	p=0.040,$mu_1 = 0.960$,$mu_2 = 231.0$

Table I
MODEL IDENTIFICATION RESULTS FOR THE DATABASE BLACK BOX

server with two 2.8 GHz Xeon processors and 2 GBytes of RAM as a load injector. The automated self-regulated load injection phase was carried out within 14 workload steps, reaching more than 120 virtual users in 5 minutes on average. The saturated resource was the CPU with a 82% usage.

The model identification tool got measures and computed the corresponding service time samples for each workload step. Each service time sample was analyzed using goodness-of-fit tests and graphical methods. Various distributions were identified with their parameters, including the Exponential, Hyper-exponential with two stages, Log-normal, Gamma and Weibull. The candidate models identified during experimentation are given in Table I. Notations for this table I are the following: λ_i refers to the interarrival rate, μ_i the service rate and $\overline{X}_i$ its mean service time. LN refers to the Lognormal distribution, Hr_2 to the Hyperexponential with two stages and Wbl to the Weibull distribution.For each load level, we select the most appropriate model (given in bold characters) according to the statistical tests best results (best p-values and fitting scores). We also give the best model's parameters (*Parameters* column): λ is the inter-arrival rate, μ the service rate for the exponential distribution (μ_1 and μ_2 for the Hyperexponential distribution and p is its probability to go to a stage), μ, σ are the shape and scale parameters of the Lognormal and Normal distributions, and a,b are the Γ distribution parameters.

As the table shows, for light and medium loads (load levels varying from 1 to 36 virtual users), we select the M/LN/4 model, as the statistical tests gives greater p-values for the lognormal distribution. For higher loads, we select the $M/Hr_2/4$ model. Graphs of Figure 7 show service times histograms with identified fitting distributions.

B. Modeling the EJB container black box

The EJB tier runs on a Linux server with two 2 GHz Xeon processors, and 1 GByte of RAM. We used a Linux server with two 2.8 Ghz Xeon processors, and 2 GBytes of RAM as an injector machine. The automated self-regulated load injection phase was carried out within 12 injection steps, reaching more than 162 virtual users in 8 minutes and 0.2 seconds. Figure 8 shows the resulting load profile. The saturated resource was the CPU with 86% usage.

The candidate models identified during experimentation are given in table II. As the table shows, for light and medium load (load levels varying from 1 to 118 virtual users), we select the M/LN/3 model except for one load level ($M/\Gamma/3$ model for load=55 virtual users). For higher loads, we select the $M/Hr_2/3$ model. Graphs of Figure 9 show service times histograms with identified fitting distributions.

C. Modeling the Web container black box

The Web tier runs on a Linux server with two 1.2 GHz PIII processors and 1 GByte of RAM. We used a Linux server with two 2.8 GHz Xeon processors and 2 GBytes of RAM as an injector machine. The automated self-regulated load injection phase was carried out within 6 workload steps, reaching more than 16 virtual users in 7 minutes and 48 seconds. The saturated resource was the JVM heap memory with 4% free space.

The candidate models identified during experimentation are given in table III. As the table shows, the load levels exhibit either an M/LN/1 or an $M/\Gamma/3$ model. Graphs of Figure 10 show service times histograms with identified fitting distributions.

D. Validation of the obtained global model

Our validation of the identified models is two-fold:

- First, we build a global queuing network model representing the SampleCluster application, using the three black boxes' models, and we perform its performance analysis or simulation at a given load level (16.2 requests/s). Then, we compare obtained performance values with real measures at the same workload level, to check the accuracy of our modelling.
- Second, we apply the automated load injection and model identification process on the whole SampleCluster architecture considered as a single black box. Then, we do

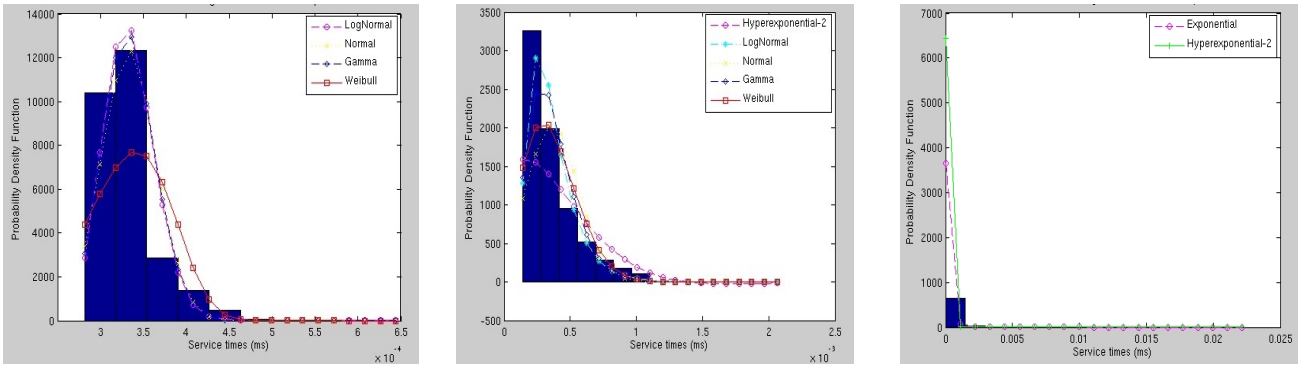


Figure 7. Service sample analysis for the database black box: light (left), medium (middle) and heavy (right) loads

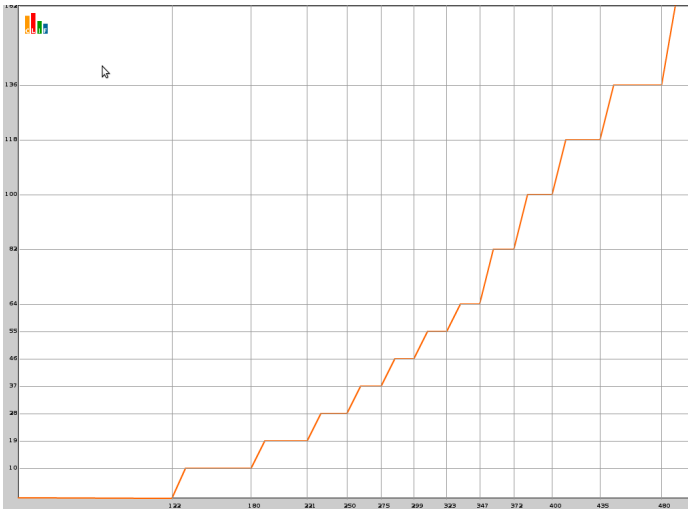


Figure 8. Load profile resulting from self-regulated load injection performed on the EJB tier. The X axis is time, from 0 to 480 seconds. The Y axis is the number of active virtual users, from 1 to 162. 12 steps have been completed, and the 13th step has been aborted because of system saturation detection.

Load	Identified Model(s)	Parameters
1	$M/\Gamma/3$, **M/LN/3**, M/Norm/3, M/Wbl/3	m=-4.282, sigma=0.132
10	$M/\Gamma/3$, **M/LN/3**, M/Norm/3, M/Wbl/3	m=-4.458, sigma=0.166
19	$M/\Gamma/3$, **M/LN/3**, M/Norm/3, M/Wbl/3	m=-4.545, sigma=0.202
28	$M/\Gamma/3$, **M/LN/3**, M/Norm/3, M/Wbl/3	m=-4.602, sigma=0.189
37	$M/\Gamma/3$, **M/LN/3**, M/Norm/3, M/Wbl/3	m=-4.653, sigma=0.247
46	$M/\Gamma/3$, **M/LN/3**, M/Norm/3, M/Wbl/3	m=-4.708, sigma=0.263
55	$M/\Gamma/3$, M/LN/3, M/Norm/3, M/Wbl/3	a=15.828, b=0.001
64	$M/\Gamma/3$, **M/LN/3**, M/Norm/3, M/Wbl/3	m=-4.708, 0.306, sigma=0.476
82	$M/\Gamma/3$, **M/LN/3**, M/Norm/3, M/Wbl/3	m=-4.739, sigma=0.363
100	$M/Hr_2/3$, $M/\Gamma/3$, **M/LN/3**, M/Norm/3, M/Wbl/3	m=-4.818, sigma=0.376
118	$M/Hr_2/3$, $M/\Gamma/3$, **M/LN/3**, M/Norm/3, M/Wbl/3	m=-4.792, sigma=0.381
136	M/M/3, $M/Hr_2/3$, M/Norm/3	p=0.138, $mu_1 = 0.862$, $mu_2 = 77.65$
162	M/M/3, $M/Hr_2/3$	p=0.094, $mu_1 = 0.906$, $mu_2 = 27.25$

Table II
MODEL IDENTIFICATION RESULTS FOR THE EJB CONTAINER BLACK BOX

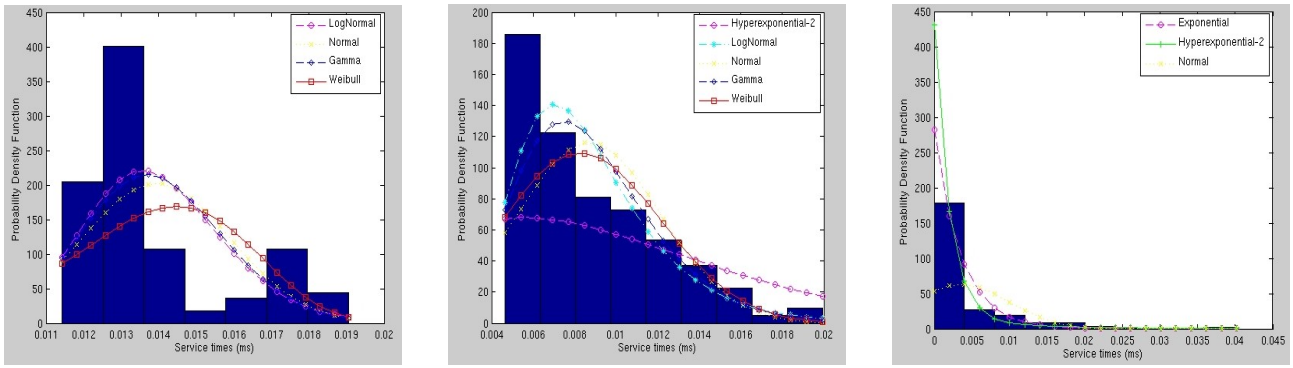


Figure 9. Service sample analysis for the EJB black box: light (left), medium (middle) and heavy (right) loads

Load	Identified Model(s)	Parameters
1	$M/\Gamma/1$, M/LN/1, M/Norm/1, M/Wbl/1	a=76.87, b=0.001
4	$M/\Gamma/1$, **M/LN/1**, M/Norm/1, M/Wbl/1	m=-3.254, sigma=0.133
7	$M/\Gamma/1$, M/LN/1, M/Norm/1, M/Wbl/1	a=62.92, b=0.001
10	$M/\Gamma/1$, **M/LN/1**, M/Norm/1, M/Wbl/1	m=-3.493, sigma=0.145
13	$M/\Gamma/1$, M/LN/1, M/Norm/1, M/Wbl/1	a=36.82, b=0.001
16	$M/\Gamma/1$, **M/LN/1**, M/Norm/1, M/Wbl/1	m=-3.679, sigma=0.166

Table III
MODEL IDENTIFICATION RESULTS FOR THE WEB CONTAINER BLACK BOX

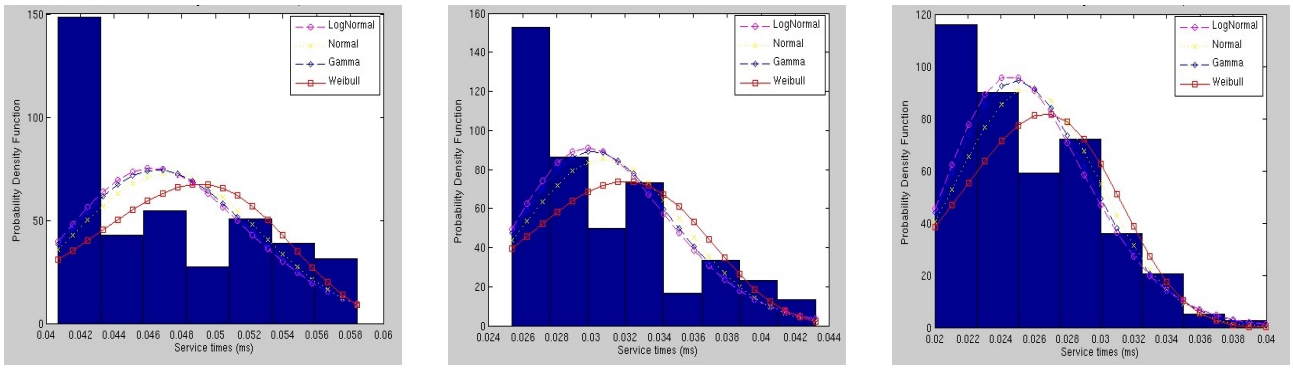


Figure 10. Service sample analysis for the WEB black box: light (left), medium (middle) and heavy (right) loads

Performance index	Estimated value for the system decomposed as 3 black boxes	Estimated value for the system seen as a single black box	Measure
Response time	52 ms	51 ms	52 ms
Throughput	16.2 requests/s	16.2 requests/s	16.2 requests/s
Clients number	0.87	0.80	-

Table IV
COMPARISON BETWEEN THEORETICAL AND EMPIRICAL PERFORMANCE INDEXES

performance analysis of the obtained model at the same load level and we compare with the queuing network results and the real measures. The goal here is to check the model accuracy, and especially to see if decomposing the whole system into three black boxes gives more accurate results than when considering a single model for the whole system.

Table IV shows mean values of theoretical performance indexes computed using the identified models of each tier and of the system modelled as a single black box, at the load level of 16.2 requests/s. We see that these values are very close to the mean empirical values. The relative error for the mean response time is 0.75% for the 3-tiers decomposition and 2.47% for the single global model. This is a partial validation of our full automated benchmark and process, on this particular sample application. This result also shows that accuracy is actually better with the 3-tiers decomposition, and with a finer granularity (in our example, relative error is 3.3 times as small), which partially validates the interest of decomposing the global system into several black boxes and building a queuing network.

E. Performance prediction for self-sizing feature

In this section, we sketch an example of autonomic reaction to possible bottlenecks, which may appear in the SampleCluster system. Whenever a bottleneck appears, we show how the Analysis function of the self-sizing control loop is able to find the best system configuration to apply, through performance analysis/simulation of possible target configurations.

Of course, the notion of "best configuration" is a matter of viewpoint. From the system user's viewpoint, only quality of experience criteria count, such as end-to-end response time, service availability and reliability. From the system operator's viewpoint, a trade-off must be found between investment and operating expenditures on the one hand, and client satisfaction on the other hand. Request throughput capability is a good criteria for the operator since it rules how many clients may be simultaneously served by the system. Other criteria such as usage of processor, memory or network bandwidth are also of interest to optimally size the system's resources. However, the operator must also take quality of experience criteria (e.g. end-to-end response time) into account. The best configuration typically consists in minimizing the infrastructure costs, while meeting a service level agreement with respect to given workload assumptions (number of users and resulting workload).

In our example, we assume that a load rate of 180 requests/s is submitted to the system. Performance simulation of current configuration gives a utilization equal to 1 for the Web container black box, thus showing saturation of this tier, and $9778s$ global response time, i.e., 2.71 hours, which is an unacceptable quality of experience.

In this case, the self-sizing control loop would launch a decision process, which chooses the best solution. Possible target configurations are depicted in Figure 11. Table V shows, in one hand, global response time and global throughput for the multi-tier system, and in the other hand, utilization indexes of each tier. These performance results are computed by the performance analysis/simulation function of the control loop. We can see that solution 3 results in an improved global response time and an enhanced throughput. This configuration is hence the best one and more adequate to our multi-tier system, and then will be chosen by the autonomic control loop to be applied to the system.

VII. Conclusion and Future Work

This paper addresses automated performance modelling of software elements considered as black boxes. Our goal is to be able to predict the performance of a distributed application configuration composed of these black boxes, and to use it in autonomic systems so that self-* features can integrate performance awareness while they plan system reconfigurations. Target applications are those being able to evolve to more strengthened configurations, through replication of constituents.

For this purpose, we have proposed a performance model identification process for black boxes. The process automatically delivers, for each black box under test, one or several queuing models with their parameters, according to a number of workload ranges. This process has been implemented as a framework prototype, reusing the CLIF open source load testing platform for workload generation and resource utilization monitoring. The process usability has been assessed through the experimentation of a three-tiers web application and the first results are promising. A first, partial validation is shown on an clustered Java EE sample application, showing a good level of response time prediction accuracy, and even better when the application is decomposed into black boxes instead of considering it as a single black box.

However, some issues and difficulties were met:

- Isolating a black box from dependent servers (i.e., servers that are subsequently invoked by the black box when it processes a request) is mandatory but not straightforward to achieve. Two solutions may be adopted:
  (i) build plugs to replace dependent servers and characterize their performance; this solution is specific to each protocol, and involves programming, benchmarking and possibly some network-level wire-tapping efforts;
  (ii) follow a step by step approach starting from the final black box in the architecture and using identified models of the characterized tier. We preferred the latter solution for it is simpler to implement. But, while the plug can be designed for high performance (it is a fake server), a real server may saturate before the tested black box. In this case, the black box modelling will partially complete, with missing high load steps, because of the bottleneck. The solution is to replicate the dependent server causing the bottleneck.
- We provide no particular support for capturing traffic routing between black boxes. First, we consider a simplified vision of the traffic, assuming a pipe call topology between black boxes, with no feedback calls.

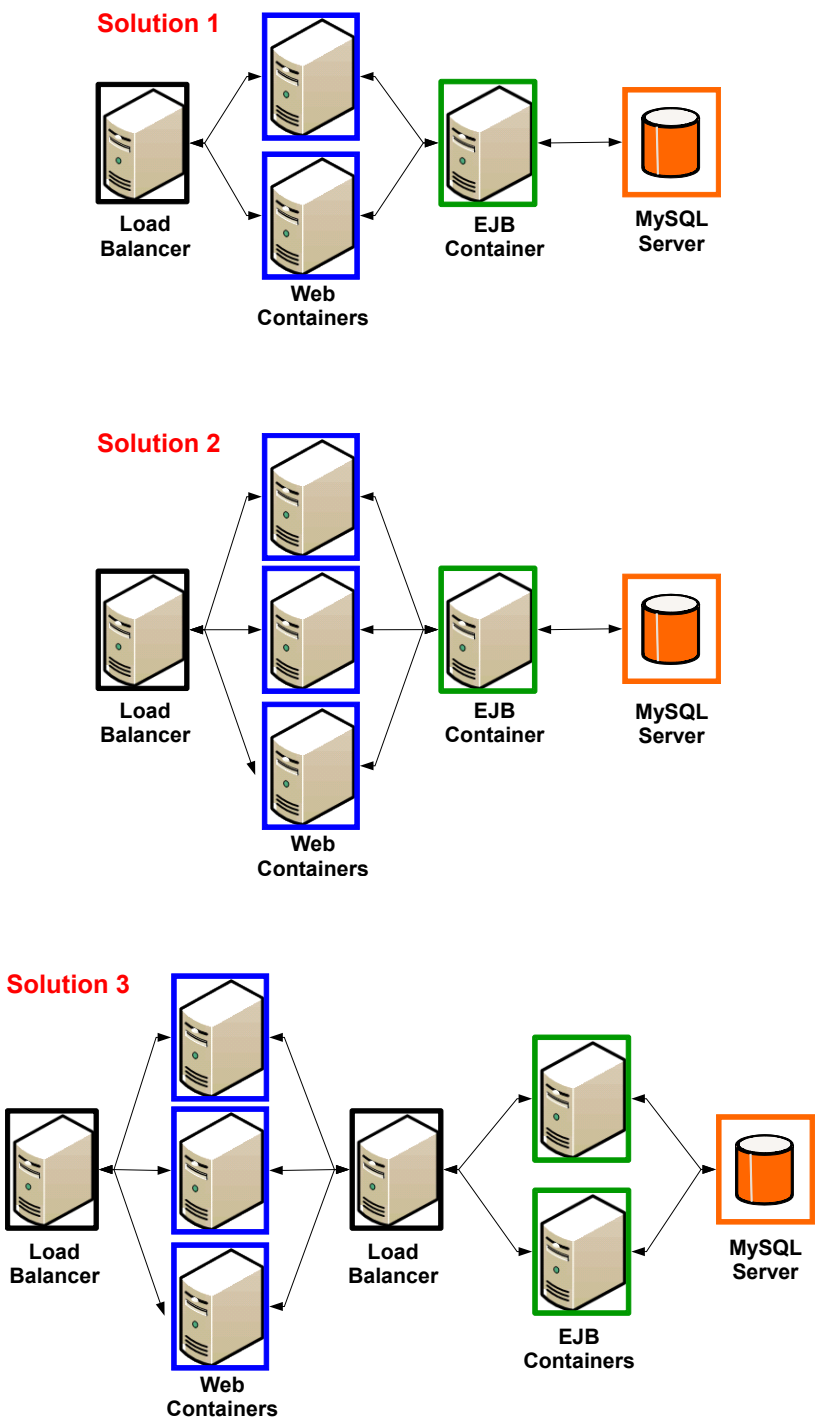


Figure 11. Possible target configuration for solving the detected bottleneck

Performance index	Solution 1	Solution 2	Solution 3
Response time	4186 sec	1855 ms	1454 ms
Throughput	70.88 requests/s	106.80 requests/s	117.20 requests/s
Utilization Web 1	0.90	1	1
Utilization Web 2	0.90	1	1
Utilization Web 3	-	1	1
Utilization EJB 1	0.70	1	0.59
Utilization EJB 2	-	-	0.60
Utilization MySQL	0.09	0.14	0.15

Table V
PERFORMANCE ANALYSIS RESULTS FOR POSSIBLE TARGET CONFIGURATIONS

However, this assumption is met with common multi-tier applications, which are our key targets. Second, we don't provide a solution to capture multiple round-trip calls between black boxes, in which an incoming call in tier n may result in more than a single call to tier $n+1$. However, our queuing network builder supports a multiplication factor, which makes it possible to specify that a given request on black box n generates r requests on black box $n+1$.

- Our work considers a traffic of homogeneous requests. Considering heterogeneous traffics with different request kinds coming with highly variable service times would require some more work. The issue is quite wide if you consider also heterogeneous admission policies depending on the request kind (priorities, preemption, etc.). But this would be typically not the case for the class of multi-tier applications we consider. Complementary experiments would give valuable feedback about the influence of requests heterogeneity in terms of service times on the different stages of the process and the accuracy of final performance predictions.

This work makes little assumptions about observation capabilities of black boxes: response time measurement as it is experienced by a client, and monitoring utilisation of host operating system resources. To improve and extend our framework, it would take some more intrusive observation capabilities. For instance, calls profiling and network analyzer tools should be integrated to help capture information about call routing or to help build plugs.

This work is essentially processor-centric, but it could be extended also for modelling other resources utilization (e.g., network bandwidth, RAM, disk space or disk transfer rate). Similar statistical techniques may also apply, but the set of relevant candidate distributions are likely to differ. This would be valuable for sizing each server, and not only the replication level of tiers.

Finally, our future work concentrates on the autonomic vision, since we plan to integrate this performance prediction platform to an autonomic system manager, responsible for checking or proposing new system configurations matching given performance requirements. Within the SelfXL project [25], applications of this "performance oracle" are foreseen for anticipating and dynamically adjusting the number of virtual machines required for a given service in a cloud computing environment.

REFERENCES

[1] IBM, "An architectural blueprint for autonomic computing," http://www-03.ibm.com/autonomic/pdfs/AC Blueprint White Paper V7.pdf. Last accessed: January 2011, June 2005.

[2] A. Harbaoui, B. Dillenseger, and J. Vincent, "Performance characterization of black boxes with self-controlled load injection for simulation-based sizing," in *Proceedings of CFSE'2008*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 172–182.

[3] A. Harbaoui, N. Salmi, B. Dillenseger, and J. Vincent, "Introducing queuing network-based performance awareness in autonomic systems," in *Proceedings of ICAS'2010*. Cancun, Mexico: IEEE Computer Society, march 2010, pp. 7–12.

[4] ANR Selfware project, "Selfware: Lessons learned to build autonomic systems," http://sardes.inrialpes.fr/~boyer/selfware/documents/SP1-L3-Architecture.pdf. Last accessed: January 2011, 2008.

[5] J. L. Hellerstein, D. Yixin, P. Sujay, and M. T. Dawn, *Feedback Control of Computing Systems*. John Wiley & Sons, 2004.

[6] D. A. Menasce and M. N. Bennani, "On the use of performance models to design self-managing computer systems," in *Proc. 2003 Computer Measurement Group Conf*, 2003, pp. 7–12.

[7] A. Kamra, V. Misra, and E. M. Nahum, "Yaksha: a self-tuning controller for managing the performance of 3-tiered web sites," in *IWQoS*, 2004, pp. 47–56.

[8] P. L. Fontaine Rafamantanantsoa and A. Aussem, "Analyse, modélisation et contrôle en temps réel des performances d'un serveur web," LIMOS, Tech. Rep. LIMOS/RR-05-06, 10 Mars 2005.

[9] M. Litoiu, "A performance analysis method for autonomic computing systems," *TAAS*, vol. 2, no. 1, 2007.

[10] T. Begin, A. Brandwajn, B. Baynat, B. E. Wolfinger, and S. Fdida, "Towards an automatic modelling tool for observed system behavior," in *In proceeding of the 4th European Performance Engineering Workshop (EPEW 2007)*, Springer, Ed. Berlin, Germany: Lecture Notes in Computer Science, 27–28 September 2007, pp. 200–212.

[11] D. A. Menascé, "Computing missing service demand parameters for performance models," in *Int. CMG Conference*, 2008, pp. 241–248.

[12] C. M. Woodside, T. Zheng, and M. Litoiu, "Performance model estimation and tracking using optimal filters," *IEEE Transactions on Software Engineering*, vol. 34, no. 3, pp. 391–406, 2008.

[13] B. Dillenseger, "Clif, a framework based on fractal for flexible, distributed load testing," *Annals of Telecom*, vol. 64, no. 1-2, pp. 101–120, Feb. 2009.

[14] L. Kleinrock, *Queueing Systems*. New York: Wiley-Interscience, 1975.

[15] R. K. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and modelling*. Canada: John Wiley and Sons, Inc, April 1991.

[16] W. Stewart, *Introduction to the Numerical Solution of Markov Chains*. Princeton: Princeton University Press, 1994.

[17] Chakravarti, Laha, and Roy, "Kolmogorov-smirnov test," *Handbook of Methods of Applied Statistics*, pp. 392–394, 1967.

[18] R. D. Smith, "The dynamics of internet traffic: Self-similarity, self-organization, and complex phenomena," 2008, last accessed: January 2011. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:0807.3374

[19] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi, *Queueing networks and Markov chains. modelling and Performance Evaluation with Computer Science Applications*. Canada: JOHN WILEY and SONS, 2006.

[20] R. A. Marie, "An approximate analytical method for general queueing networks," *IEEE Trans. Softw. Eng.*, vol. 5, no. 5, pp. 530–538, 1979.

[21] I. F. Akyildiz and A. Sieber, "Approximate analysis of load dependent general queueing networks," *IEEE Trans. Softw. Eng.*, vol. 14, no. 11, pp. 1537–1545, 1988.

[22] The MathWorks, Inc., "MATLAB and simulink for technical computing," http://www.mathworks.fr/. Last accessed: January 2011, 1994-2010.

[23] D. of Statistics and Mathematics, "The R project for statistical computing," 2003, http://www.r-project.org.

[24] OW2 Consortium, "JONAS, java open application server," http://wiki.jonas.ow2.org/xwiki/bin/view/Main/WebHome. Last accessed: January 2011.

[25] ANR SelfXL project, "Selfxl - self-management of complex and large scale systems," http://selfxl.gforge.inria.fr/dokuwiki/doku.php. Last accessed: January 2011.

Layers Optimization Proposal in a Post-IP Network

João Henrique de Souza Pereira[‡], Eduardo Souza Santos[*],
Fabíola Souza Fernandes Pereira[†], Pedro Frosi Rosa[†] and Sérgio Takeo Kofuji[‡]

[‡]*Department of Electrical Engineering*
University of São Paulo, São Paulo, SP, Brazil
Email: joaohs@usp.br, kofuji@pad.lsi.usp.br

[*]*Department of Electrical Engineering*
Federal University of Uberlândia, Uberlândia, MG, Brazil
Email: eduardo@mestrado.ufu.br

[†]*Department of Computer Science*
Federal University of Uberlândia, Uberlândia, MG, Brazil
Email: fabfernandes@comp.ufu.br, pedro@facom.ufu.br

***Abstract*—In this work, a post IP structure is proposed, which eliminates the use of network and transport layers in networks with layer 2 connectivity. The goal is to optimize the network structure for distributed systems at the next generation Internet and to propose a delivery guarantee mechanism to FINLAN packets, that emphasizes the optimized relation between applications and lower network layers. The results compared with Internet protocols are also presented.**

Keywords-Network Layers Optimization; Local Networks; Post TCP/IP; Delivery Guarantee.

I. INTRODUCTION

Applications that use the computer networks infrastructure have evolved rapidly in recent years increasing the need to establish communication with high throughput and low end-to-end delays (among other requirements). Many of these applications are supported by TCP/IP (Transmission Control Protocol/Internet Protocol) architecture, which was developed to support the communication when the Internet was used to interconnect a limited number of nodes and the applications, in most cases, were used for simple exchange of messages and file transfers.

It may be noted that, in TCP/IP architecture, there are redundancies and obsolete fields in its protocol stack that increase the network overhead. For example, the checksum field is used both for the IP and the TCP headers and this could be reduced or even eliminated in certain cases, since the detection and correction of errors is the link layer's responsibility. Also, the Type of Service (ToS) field was remodeled to be used as Differentiated Services Code Point (DSCP).

In view of such enhancement possibilities in the current TCP/IP architecture, the purpose of this work is to propose an alternative for this architecture, given a structure that can meet the requirements of current applications in a simplified and optimized way, taking into account the real needs of applications such as Voice over IP (VoIP) communication, which was developed about fifteen years later than the TCP/IP and, therefore, suffer impacts as jitter and packet delay.

One reason that encourages this initiative is the possibility to collaborate in a field that has very few proposals and whose objective is to contribute with the studies in next generation Internet technologies, that can hold the applications needs better than the IP, TCP and UDP (User Datagram Protocol).

The principal idea of a new structure, called Fast Integration of Network Layers (FINLAN) and introduced in [1], is to eliminate the protocols of network and transport layers, which will be possible by re-structuring the link layer (Ethernet) protocol, which will serve directly the application layer. It is important to emphasize that this proposal does not have the intention to eliminate the use of TCP/IP protocols, but to make Ethernet packets hybrid using the current structure of layers and the new proposed structure.

In this work is also proposing a mechanism to guarantee the data delivery in FINLAN, prefaced in [2]. With this mechanism, the operational system will receive the information over the needs of the applications and guarantee the data delivery, when necessary, without the need to use distinct transport protocols, such as UDP or TCP.

This paper is organized as follows. Section 2 presents the related work and a network ontological overview that motivated this research for a optimized communication structure. In Section 3 the FINLAN structure is presented, highlighting its functional features. In Section 4, the proposal of delivery guarantee to FINLAN packets is detailed. In Section 5, are shown details about implementation progress and in Section 6 the preliminary results by the layers simplification are

discussed. Finally, in Section 7, a conclusion is presented and future works in this research are suggested.

II. Related Work

It is possible to find different communications structures in networks, like ATM (Asynchronous Transfer Mode) and X.25, that were proposed and adopted years ago.

About TCP/IP architecture, generally there are more improvements at the lower and application layers, but there are not so much evolution at the intermediate layers. Among these improvements, it is important mentioning proposals that deal with deficiencies in this architecture, with the advancement of new applications and, consequently, new requirements, like the protocols overhead optimization [3].

Even so, Jin and Yoo [4] show that the recent networks of high speed suffer from overhead protocols, placing them as obstacles for the high performance applications that explore high speed connections, for example, in clusters.

In the context of distributed systems evolution, it is worthy mentioning alternative technologies to the Ethernet networks. The Local Area Network (LAN) of high speed Myrinet is one of them, having less protocol overhead than standard Ethernet networks, supplying more throughput, low latency, and less interference [5].

Another example in the new technologies for high speed networks is the Infiniband. Such technology for high speed interconnection supports new protocols of low latency and high broadband, which nowadays only have an inferior performance compared to a Gigabit Ethernet. In [6] it is possible to check the high performance of IP protocol integrated into the IPoIB (IP over Infiniband) technology.

Old technologies as X.25 were created with a different layer structure that meets specific requirements as safety and reliability [7]. Another old one is the Frame Relay [8], an evolution from X.25 networks, developed to transmit data in a specific architecture, modeled in 3 layers, detached from the TCP/IP architecture. Such structures were proposed some years ago and until today are present in networks.

In this genealogy of technologies, it is necessary to highlight the SNMP (Simple Network Management Protocol) over Ethernet specification in the TCP/IP architecture [9]. According to this proposal, the network management protocol can be used over the MAC Ethernet layer, instead of going by the stack of UDP/IP protocols. So the data transfer occurs through a logical mechanism that avoids the need of network and transport layers protocols.

Also, several studies have been developed facing an alternative network architecture: user-level network. The idea is to use techniques that transfer messages directly to the user level, releasing the use of the stack of the operating system and thus reducing network overhead. One example are the techniques of zero-copy [10], used in [11], as an architecture of network interface for high speed user level devices and in [12] for communication over InfiniBand.

It is also worthy mentioning proposes in the context of mobile networks, that deal with TCP/IP architecture difficulties, for example, TCP congestion windows [13]. Analyzing the network-based mobility management scheme instead of host-based mobility, the work of [14] becomes also an example that the mobility networks need evolution and changes in their architecture.

So, it is possible to realize the proposal about simplified network layers, shown in this paper, to the context of mobility networks.

Several works have been developed also in the area of next generation Internet with the proposal of new address solutions, joined with the search for mobility and safety, according to the works [15]–[18]. In [16], it is presented a new model of inter-connection among network elements through flat routing, and in [17], an architecture is proposed for address, which meets challenges such as dynamicy, safety, and multi-homing.

In this context, this work proposes a post IP study for a structure, called FINLAN, which eliminates the use of network and transport layers in networks with layer 2 connectivity, differently from the work [18], which proposes the creation of an intermediate identification layer for a new address way.

Therefore, the idea of FINLAN is simplify the way the information is addressed and transmitted, optimizing the network structure and reducing the neighborhoods dependency. This next generation Internet layer structure can help for a horizontal addressing, as proposed in the correlate works discussed in [19], [20].

A. Network Ontological Overview

The TCP/IP architecture is powerful and flexible to handle different application needs. However, for the last 30 years, the main protocols at the Network and Transport layers have not evolved to support the new application requirements.

This statement is comproved by the evolutionary review of the RFC 760, 761 and 768 described in [19]. In this analysis is verified that the IP, TCP and UDP protocols had not evoluted substantially since 1981. Since the 80's, at the Network and Transport layers, are others specifications, as the IPv6 specified in 1995 and the SCTP (Stream Control Transmission Protocol), in 2000.

These specifications solve some problems in the intermediate layers level, but still remain some gaps (or opportunities) for contributions to improve the Internet communication mechanisms.

Proposals as the horizontal addressing by entity title can contribute to reduce the increasing of the Internet architecture protocols complexity. This complexity is increasing as a result of the new communication requirements that appeared after the specification of the main protocols of the TCP/IP intermediate layers. The Figure 1 built from the Internet Engineering Task Force (IETF) RFC index, shows

the Internet protocols complexity evolution, since the first IETF specification through nowadays [20].

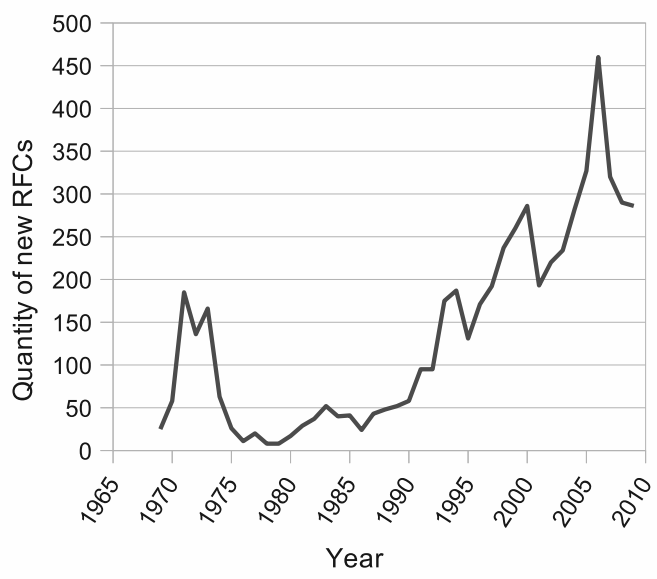


Figure 1. RFCs specified by year

By this graphic information there is a risk of complexity collapse in the Internet because, usually, new specifications demand new technological requirements for some network elements (hardware and/or software). To amplify this discussion, in another direction, the W3C (World Wide Web Consortium) has specified an ontological application architecture for the Semantic Web, and this architecture does not have the service support for semantics in the actual Transport and Network layers.

This ontological architecture for the Semantic Web has the power of the semantic communication limited inside the application layer, as this layer can not send meaning to the Network and Transport layers. The application layer generally just can choose the transport protocol (TCP, UDP, SCTP, etc.), its ports, and set the destination IP.

In fact, using the traditional TCP/IP layers 3 and 4 protocols, the application layer can not inform completely all of its needs as mobility, security, Quality of Service (QoS), Quality of Experience (QoE), and others. In this scenario, the FINLAN, also, is not able to handle some fundamental concepts of ontology, as the "formal representation of one conceptualization" defined by Gruber, neither to understand semantics in a more comprehensible way, as can be done by the use of OWL (Web Ontology Language) Full, OWL DL (Description Logics) or OWL Lite.

At the actual step, the FINLAN studies aims to contribute for a Prove of Concept (PoC) to the application layer be closer to the lower layers. Also, this PoC checks the possibility to the application layer be able to inform the lower layers the necessity of data delivery guarantee, or not. In this way, it is possible to use the same protocol to the communication between different applications, with or without delivery guarantee in a not reliable connection, as the Local Area Networks.

Another FINLAN contribution is to be one step to the possibility to implement the applications addresses horizontally by entity title, to expand the unified addresses to different communication entities like hosts, users, sensor networks, and others, as proposed by Pereira in [21] to a world wide network.

B. Delivery Guarantee Work

Besides the application layer be able to inform the delivery guarantee need, this work also proposes to improve FINLAN with a delivery guarantee mechanism. This can qualify FINLAN to support others next generation Internet requirements [15], in an optimized way.

It is technically complex to guarantee reliable data transmission over the networks. There are different technologies for loss detection and packet re-transmission up to architectures that do not worry about guaranteeing a reliable transmission.

Among the existing technologies it is possible to point out the old Frame Relay [22] as an example of protocol that works in the lower layers and does not worry about guaranteeing the data delivery. The idea is that the application be in charge of dealing with packet loss. The ATM [23] is another example of technology that does not implement the delivery guarantee. In this technology, there is a great trust in the transmission medium.

The ATM architecture is different from the TCP/IP architecture, because in TCP [24] the delivery guarantee can be done in a non reliable transmission medium through packet confirmation. This occurs similarly in SCTP, which is also a transport protocol in TCP/IP architecture.

There is also the MPLS (Multi Protocol Label Switching), which is a low layer protocol with a large capacity of traffic management and therefore, with more reliability, although it is not designed to guarantee that all data packets will get to the destination [25].

Even in the transport layers, it is possible to point out protocols, which do not meet the delivery guarantee requirement, for example, the UDP [26] of TCP/IP architecture. Such fact can be explained by the purpose of each protocol or architecture, that is, they transmit data that do not have a delivery packet guarantee as a necessary requirement.

On the other hand, it is also possible to find solutions that guarantee the data delivery, implemented in different layers. There are also the old X.25 networks [27], which guarantee the delivery based on confirmation of each data packet received.

In more recent technologies, as Myrinet and Infiniband [28], it is also possible to verify a structure, which provides a reliable message transmission through the sending of

messages of destination requiring the missed packets [29], [30].

There are still works in wireless and mobile networks that have solutions to guarantee data delivery due to the flexibility of a mobile host. In [31], for example, a protocol that uses Automatic Repeat reQuest (ARQ) and Forward Error Correction (FEC) mechanisms is proposed, aiming at low rate retransmission. In [32], there is the analysis of the problem of using variable paths aiming at low loss rate and the proposal for a load balancing algorithm as a solution.

According to the past and current architectures and mechanisms that deal with delivery guarantee or not, this work proposes a flexible approach, which applications can choose if they need or not of this requirement. This possibility to attend the real requirements of applications is the major purpose of FINLAN.

III. Layer Optimization Proposal

The creation of an alternative layer structure to computer networks, that can meet the current technological needs, can enable a better use of applications needs, that did not exist yet when the specifications of IP, TCP, and UDP protocols occurred.

A good example is the VoIP applications that were developed in the 90's, around 15 years after TCP and IP protocols came up, and faces a lot of QoS problems. To solve some of these problems would be necessary to think about the structure of the current networks trying to accomplish adequations to the new technologies.

In this aspect, the optimization of TCP/IP architecture through the redesign of fields that throughout the years have lost their meaning, along with the desire to meet the requirements of new applications is an inherent need in the growing use of communication networks and the future new applications.

This way, the modeling of a new communication structure among applications can be on focus, and thinking about it, an optimized TCP/IP stack is proposed, previously introduced in our work [1], with some changes in the Ethernet layer protocol.

Figure 2 shows a comparison between the current protocol stack and the new suggested one. It can be noticed that in the new structure, the packets are delivered directly from the link layer to the application layer, eliminating the transport and network layer protocols.

To realize this change, the initial proposal consists of establishing communication between two applications in distinct hosts enabling the exchange of data with the use of only the information from the network interfaces from these hosts for the addressing, in other words, the addressing of applications in Local Area Networks is done with the physical addresses from the machines (MAC Address), direct to the processes without the TCP or UDP ports.

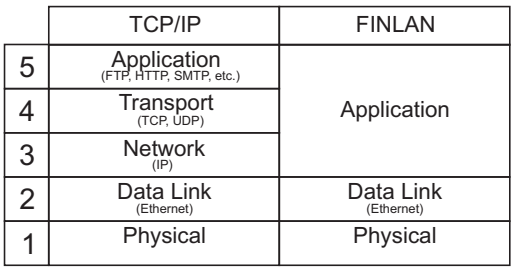


Figure 2. Comparison among the protocol stack

So, to meet the requirements of this new structure, it is proposed some changes in the Ethernet heading in a way that it can still support the TCP/IP structure and also allow the use of the new layer structure. Thus, the separation among packets that use the TCP/IP layer structure and FINLAN will be performed through EtherType field from Ethernet heading.

Hence, the transfer of packets that have the designated EtherType for the new layer structure will use this new communication way, based on the direct addressing of applications through communication flows in networks with connectivity in layer 2.

This proposal was performed in a way that the current structure of Ethernet heading is kept, with fields that consist of identification of the number flow bytes, the packet, the sequence number, and the fields responsible for transporting these values. Therefore, the heading of a FINLAN application can be described by the Figure 3.

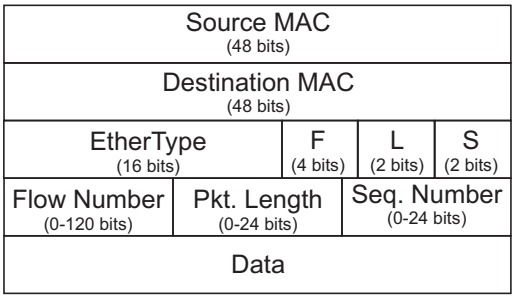


Figure 3. Ethernet heading structure for FINLAN applications

The identification bits contain three fields, "F", "L", and "S", which are the number of bytes used in the fields "Flow Number", "Packet Length", and "Sequence Number". The "F" is represented by a nibble, enabling the aforementioned field to have from 1 to 15 bytes of size and therefore the field "Flow Number" can have the values shown in Table I.

It is possible to notice that the field "Flow Number" can have values about $2^{120} - 1$, that show the great number of simultaneous connections. Likewise, the "S" and "L" fields inform that the field "Sequence Number" and "Packet Length" can have from 1 to 3 bytes of size, in other words, values from 1 to 16777216.

TABLE I
RANGE OF POSSIBLE VALUES FOR THE FIELD "FLOW NUMBER"

Number of bytes	Range of values
1	0 to 255
2	0 to 65535
3	0 to 16777215
4	0 to 4294967295
...	...
15	0 to $2^{120} - 1$

Moreover, considering that the "Packet Length" identifies the number of bytes in the data field and in the heading, it is possible to identify a packet size of 16 Megabytes, that meets part of the future networks needs. For the communication between two stations to take place in a network connected in layer 2, a data packet can be addressed using the physical address. However, more than just physical stations, it is also necessary to address the applications.

According to the current TCP/IP architecture, the IP address is used to locate a host in a network and for each IP there is a series of TCP and UDP ports, where different applications run. Thus, an application can be identified by the TCP or UDP port it is using.

According to the proposal, it was developed a way to deal with and manage the communication channel between applications. So, it will not be necessary the use of port and IP addresses. In this proposal, the identification of hosts will be done by the MAC address and applications will be identified by a Flow Number, and a Sequence Number identify sessions.

When an application is started, a flow number is associated with it. Such association can be performed in two different ways: in case the application already has a reserved port according to the current architecture, it will have the same flow number. The numbers from 1 up to 65535 (64k) are reserved for correlation with the TCP/IP ports. Otherwise, the application will request that the operational system chooses the flow number that therefore will be above 65535.

When one application is initiated, it requests an available flow number to the FINLAN daemon that communicates with the operational system that will be in charge of informing the other hosts what flow the mentioned application has. To establish communication with another host, the application needs to create a new thread, which will request from the daemon a sequence number to establish the communication.

This way, the thread sends a packet to the application flow number running in the destination host. The destination host, when receives this packet, will check that there is no communication established before with this sequence number, so it will create a new thread that will use the same sequence number.

After establishing the sequence numbers for the applications in both hosts, the communication can be initiated. A packet sent from host A to host B will have a sequence number related to the application thread that is running. When the packet gets to the destination host, the operational system daemon will deliver the packet to the application that is connected to the specified flow and the thread of this application will receive the data.

The idea is that with this new communication structure, more simplified, will be possible to improve the header with mechanisms like delivery guarantee, security, error detection and correction, in a flexible way, according to the needs of each application. As one example, in the next section is presented a mechanism that realizes delivery guarantee in FINLAN.

IV. DELIVERY GUARANTEE

One of the FINLAN proposes is to meet the application needs. In this context, there is a need to create mechanisms of delivery guarantee, that could be used for services, which require reliable data transmission, such as sending and receiving files.

In this sense, the work shown in [2] proposes a mechanism to delivery guarantee for FINLAN, where the need to make the guarantee is informed by the application via the G flag inserted in the FINLAN header.

Thus, when G=0, there is no delivery guarantee and FINLAN works as described on the initial proposal. When G=1, the field "Packet Number" is enabled with 16 bits. This field is located after the "Sequence Number".

According to Round Trip Time (RTT) algorithm created by Jacobson [33], the delivery guarantee mechanism in FINLAN is done by periodical confirmation according to network behavior characteristic at each instant in time. In this confirmation, the network elements inform the next sequence number to receive the confirmation, indicating the next packet to be confirmed or a packet loss.

This confirmation packet is similar to a keep-alive and does not have data field, having the field L (the quantity of bytes of "Packet Length" field) equal to "00", so the "Packet Length" is suppressed in FINLAN packet and the "Confirmations Quantity" (CQ) field is added, with 8 bits, after the "Packet Number" field.

Depending on the value of the field "Confirmations Quantity", the fields C1, C2, C3, ..., C255 are filled, to inform from 1 up to 255 "Packet Number" not received. Each "Cx" field has 16 bits, since this is the size of the field "Packet Number". For this kind of packet (L=00) the FINLAN structure has the format shown in Figure 4.

Similar to TCP [24], when one keep-alive is sent, a timer is activated and if there is the receive confirmation, the timer is switched off. Otherwise, the keep-alive is re-transmitted. To optimize the use of network in cases of communication failures, when the network element notices that the keep-

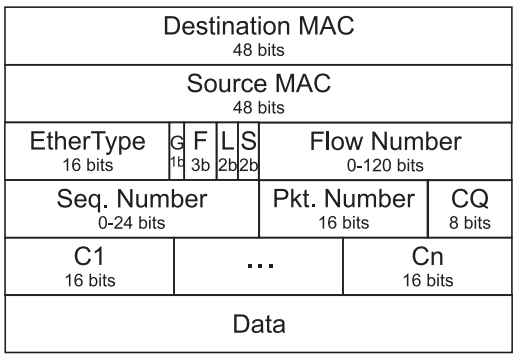


Figure 4. FINLAN confirmation packet [2]

alive is missing, the data transmission is interrupted and will only keep the periodical sending of its keep-alive.

In [2], the timeout interval to re-send a packet is calculated with the use of algorithm created by Jacobson [33], that calculates dynamically the timeout based on continuous evaluation of network performance.

Taking into account the success of the method above, this proposal will be included in implementations that will be performed directly on the Linux kernel, allowing also comparative data in relation to the mechanism implemented in TCP/IP architecture in relation to various scenarios, such as failures on communication network, packet loss and end-to-end delay.

V. Implementation Progress

The FINLAN proposal was implemented in one library using C language and RAW socket. However, the delivery guarantee mechanisms are not in this library yet. Thereby, for the next steps, this library and the delivery guarantee mechanisms will be implemented at a Operational System (OS) Kernel level.

Therefore, the actual stage of this work is the implementation of the proposal using the low level libraries of the Linux Kernel. An important point of this level is to become hybrid the sending and receiving of the FINLAN packets, allowing that the source host be able to deal with errors about identification of these packets. If a FINLAN packet cannot be recognized due the lack of the FINLAN stack, the TCP/IP stack will be used, as shown in the Figure 5.

The Figure 5 shows a scheme representing such structure. It is possible to observe that two elements are considered and they will do the selection of the packages according to the protocol in use. The first one, called “Packet Manager”, is responsible for directing the setting up of the package according with the application layer request, which will inform the transport layer protocol or will inform if the package should be delivered to the FINLAN stack.

The other element, named “Packet Director”, works when it receives a package and its function is to verify if the package is using the FINLAN structure, in this case such

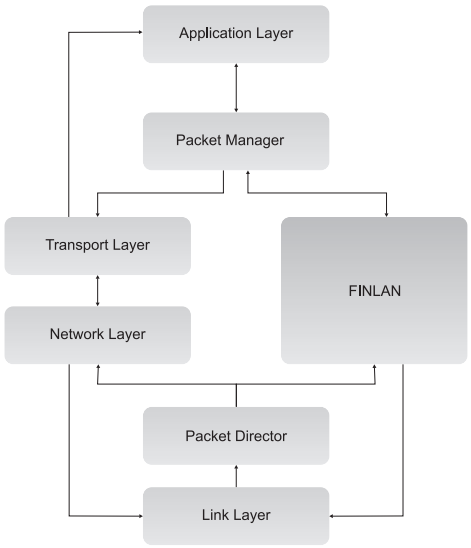


Figure 5. FINLAN model for hybrid communication

package will be delivered to the FINLAN stack, otherwise it will be sent to the OS standard flow.

In addition, mechanisms like delivery guarantee and error detection, that will be used according to the application needs, are being developed. Thereby, a header of variable length will be used, helping the network overhead decreasing.

The next section presents the implementation results at this stage and some comparative tests between FINLAN and the TCP/IP architecture, at the same environment.

VI. Results

In order to validate the proposal of this work, it was necessary the implementation of the suggested structure. So, in a first step, libraries in C language that supply the services and characteristics presented in the FINLAN proposal had been developed by [34], providing the necessary methods to communicate using FINLAN for the application layer.

For so, it was used a Linux operational system with Kernel 2.6.28-14 and the GCC (GNU Compiler Collection) 4.3.3 compiler was used.

As the goal is to address hosts without the TCP/IP traditional intermediate layers, the library RAW Socket, available in Linux OS [35], was used allowing the directly communication between the application and link layers.

Thus, the library implemented aims to make transparent to the programmer the manipulation of packets that use the FINLAN structure, as well as the creation of RAW Socket. For this purpose, are available to the developer several methods, including:

- *“create_header_eth”*: creates the header for the packet to be sent according to the flow number, packet size, source and destination addresses;

- *"create_socket_finlan"*: responsible for initiating the communication channel using the RAW Socket with the required parameters for sending and receiving FINLAN packets;
- *"send_finlan"*: do the sending of data, being also responsible for the dimension of the packets;
- *"receive_finlan"*: monitors the network interface specified in its parameters. According to the number of established flow and address of the source host, receives the packets and reassembly the file.

It is noteworthy that in the current stage of development, the FINLAN packets are marked with Ethertype 0x0880, which is currently available on the Internet Assigned Numbers Authority (IANA) and the user must inform the MAC address of destination machine.

With this application level implementation, it was possible to do comparative tests between FINLAN and the TCP and UDP protocols. These tests are related to the transfer of files with different lengths and were executed in a unique environment.

Initially, tests were performed aiming to compare sending packets with FINLAN and the TCP protocol, in this case were taken the following values for the sizes of their headers:

- TCP:
  - Ethernet header: 14 Bytes;
  - IP header: 20 Bytes;
  - TCP header: 20 Bytes;
  - Total: 54 Bytes.
- FINLAN:
  - Fixed length (MAC addresses, Ethertype, F, L and S): 15 Bytes;
  - Flow number (equal to port numbers of TCP/IP): 2 Bytes;
  - Packet length (equal to IP packet length): 2 Bytes;
  - Sequence number (maximum capacity): 3 Bytes;
  - Total: 22 Bytes.

It is important to remember that this experiment does not use the full capacity of addressing and packet size of FINLAN proposal, to put these capabilities similar to the protocols of TCP/IP architecture. However, for information that use the full capacity of FINLAN, the header can have the following values:

- Fixed length: 15 Bytes;
- Flow number (maximum capacity): 15 Bytes;
- Packet length (maximum capacity): 3 Bytes;
- Sequence number (maximum capacity): 3 Bytes;
- Total: 36 Bytes.

From the values agreed it was possible to perform tests with files of varying sizes. The Figure 6 shows a graph comparing the total amount of data sent to a file of 10 GBytes, ignoring re-transmissions and packet loss.

Looking to the diagram (Figure 6) may be noted that TCP sends 0.224 GBytes (∼229.9 MBytes) more than FINLAN,

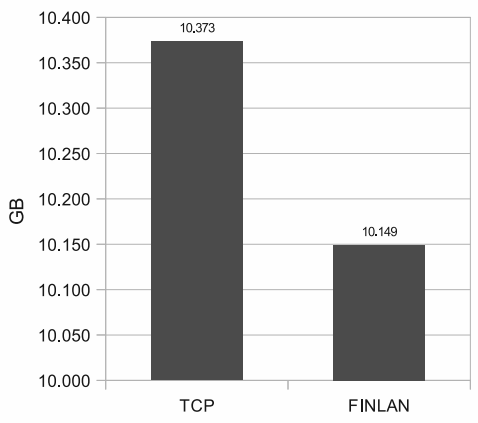


Figure 6. 10GB file transference with FINLAN and TCP

which is equal to about 2% of the total file size. This occurs because the total number of packets sent, which varies due to difference in the size, headers and confirmation.

Despite the occurrence of gain relative to the reduction of overhead compared to TCP, this implementation of FINLAN does not have a mechanism to delivery guarantee, so we did the similar tests using the UDP protocol, since this protocol does not guarantee the package delivery.

Just as in the tests with the TCP, was used standard values for FINLAN to the similar capabilities with UDP, as shown below:

- UDP:
  - Ethernet header: 14 Bytes;
  - IP header: 20 Bytes;
  - UDP header: 8 Bytes;
  - Total: 42 Bytes.
- FINLAN:
  - Fixed length (MAC addresses, Ethertype, F, L and S): 15 Bytes;
  - Flow number (equal to port numbers of TCP/IP): 2 Bytes;
  - Packet length (equal to IP packet length): 2 Bytes;
  - Sequence number (maximum capacity): 3 Bytes;
  - Total: 22 Bytes.

So, as happened in the tests with TCP, the FINLAN provided a gain compared with UDP, relative to the total amount of data sent (Figure 7), reducing the overhead.

Another comparative test with UDP was the percentage of packets successfully received at the destination. In this case, were sent files with a size of 1 KByte to 1 GByte. Performing the sending of each file four times and averaging the rate of packets received, was obtained the graph shown in Figure 8. It may be noted that, as it grows the amount of data and, by consequence, the number of packets sent, the FINLAN structure has a better performance against lost packets.

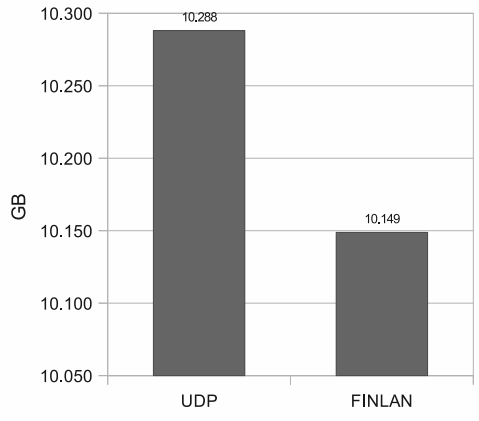


Figure 7. 10GB file transference with FINLAN and UDP

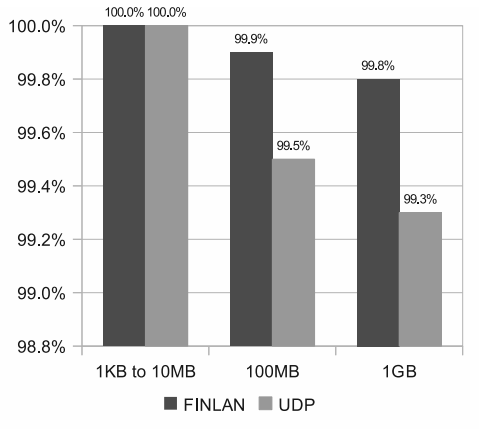


Figure 8. Percentage of packets successfully transferred (UDP vs. FINLAN)

By the tests, FINLAN has good results in comparison with UDP protocol on local networks, related to overhead reduction and lower rate of packet loss. These results brings FINLAN as one possible option for some services, as data streams applications, used in VoIP communication.

VII. Conclusions and Future Works

A proposal for communication in LAN networks was presented in this paper for contribution in the next generation Internet studies. It was also shown one way to establish communication between applications through flows that enables an optimized scenario for network use and presented the comparative results with the TCP/IP traditional intermediate layers.

In addition, the FINLAN increases the possibility of addressing an enormous quantity of applications and to send very large data packets, keeping the header slim and guaranteeing a good network usage.

This work also contributes with a proposal to guarantee the delivery of packets in FINLAN. By this, the applications do not need to use different transport protocols to have or not data delivery guarantee. In this proposal, the applications only need to inform the operational system their need about data delivery guarantee by using FINLAN library. In turn, FINLAN enables the network overhead reduction by reducing the redundancy and changing the packet confirmation way done by the in use protocols.

So, this proposal is just a step to improve FINLAN with a variety of QoS guarantees, a required feature for technologies for next generation Internet [14]. The idea is append new basic requirements like security and isolation in FINLAN in future works.

Currently, the FINLAN proposal had been implemented in a C library that uses RAW Socket to establish the communication directly with the link layer.

As future work, it is necessary to design algorithms for security, error correction and error detection to FINLAN, according to applications need. In parallel to this design, the FINLAN structure is being implemented directly in the kernel of the Linux operational system, to handle the packets in the network interface.

The idea is that FINLAN stack will be implemented in a hybrid way, allowing to redirect both IP and FINLAN packets to their respective stack. This will allow the FINLAN approach interact with the existing one.

For the kernel implementation is necessary to do performance tests in real environments with different types of packets and network collapses simulation, providing others comparative analysis between FINLAN and the TCP/IP architecture in local area networks.

References

[1] E. S. Santos, F. S. F. Pereira, J. H. S. Pereira, P. F. Rosa, and S. T. Kofuji, “Optimization Proposal for Communication Structure in Local Networks,” *ICNS 2010, The Sixth International Conference on Networking and Services*, pp. 18–22, 2010.

[2] F. S. F. Pereira, E. S. Santos, J. H. S. Pereira, P. F. Rosa, and S. T. Kofuji, “FINLAN Packet Delivery Proposal in a Next Generation Internet,” *ICNS 2010, The Sixth International Conference on Networking and Services*, pp. 32–35, 2010.

[3] D. Clark, V. Jacobson, J. Romkey, and H. Salwen, “An Analysis of TCP Processing Overhead,” *IEEE Commun Mag*, 1989.

[4] H. Jin and C. Yoo, “Impact of Protocol Overheads on Network Throughput over High-speed Interconnects: Measurement, Analysis, and Improvement,” *J. Supercomput*, vol. 41, pp. 17–40, 2007.

[5] A. Barak, I. Gilderman, and I. Metrik, “Performance of the Communication Layers of TCP/IP with the Myrinet Gigabit LAN,” *Comput Commun*, vol. 22, pp. 989–997, 1999.

[6] R. E. Grant, M. J. Rashti, and A. Afsahi, “An Analysis of QoS Provisioning for Sockets Direct Protocol vs. IPoIB over Modern InfiniBand Networks,” *Proceedings of the 2008 International Conference on Parallel Processing*, 2008.

[7] A. Malis, D. Robinson, and R. Ullmann, "Multiprotocol Interconnect on X.25 and ISDN in the Packet Mode," *RFC 1356, BBN Communications, Computervision Systems Integration, Process Software Corporation*, 1992.

[8] T. Bradley, C. Brown, and A. Malis, "Multiprotocol Interconnect over Frame Relay," *RFC 1490*, 1993.

[9] M. Schoffstall, C. Davin, M. Fedor, and J. Case, "SNMP over Ethernet," RFC 1089, Rensselaer Polytechnic Institute, MIT Laboratory for Computer Science, NYSERNet, Inc., University of Tennessee at Knoxville, Tech. Rep., 1989.

[10] H. Kitamura, K. Taniguichi, H. Skamoto, and T. Nishida, "A New OS Architecture for High Performance Communication over ATM Networks - Zero-copy Architecture," *Proceedings of the 5th international Workshop on Network and Operating System Support For Digital Audio and Video*, 1995.

[11] T. von Eicken, A. Basu, V. Buch, and W. Vogels, "U-Net: a User-level Network Interface for Parallel and Distributed Computing," *Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles*, pp. 40–53, 1995.

[12] G. Santhanaraman, J. Wu, W. Huang, and D. Panda, "Designing Zero-Copy Message Passing Interface Derived Datatype Communication Over Infiniband: Alternative Approaches and Performance Evaluation," *Int. J. High Perform. Comput. Appl.*, pp. 129–142, 2005.

[13] Y. Lu, C. Huang, and T. Sheu, "Three-color Marking With MLCN for Cross-Layer TCP Congestion Control in Multihop Mobile Ad-hoc Networks," *Proccedings of the New Technologies, Mobility and Security 2007 Conference*, pp. 145–157, 2007.

[14] D. Damic, "Introducing L3 Network-based Mobility Management for Mobility-Unaware IP Hosts," *Proccedings of the New Technologies, Mobility and Security 200*, pp. 195–205, 2007.

[15] R. Jain, "Internet 3.0: Ten Problems with Current Internet Architecture and Solutions for the Next Generation," *Miliray Communications Conference, 2006, MILCOM 2006*, pp. 1–9, 2006.

[16] R. Pasquini, F. L. Verdi, and M. F. Magalhães, "Towards a Landmark-based Flat Routing," *27th Brazilian Symposium on Computer Networks and Distributed Systems - SBRC 2009, Recife - PE, Brazil*, May 2009.

[17] R. Pasquini, L. Paula, F. Verdi, and M. Magalhães, "Domain Identifiers in a Next Generation Internet Architecture," *IEEE Wireless Communications & Networking Conference - WCNC 2009, Budapeste*, 2009.

[18] W. Wong, R. Villaca, L. Paula, R. Pasquini, F. L. Verdi, and M. F. Magalhães, "An Architecture for Mobility Support in a Next Generation Internet," *22nd IEEE International Conference on Advanced Information, Networking and Applications - AINA 2008. Okinawa, Japan*, March 2008.

[19] J. H. S. Pereira, S. T. Kofuji, and P. F. Rosa, "Distributed Systems Ontology," *IEEE New Technologies, Mobility and Security Conference - NTMS, Cairo*, 2009.

[20] ——, "Horizontal Address Ontology in Internet Architecture," *IEEE New Technologies, Mobility and Security Conference - NTMS, Cairo*, 2009.

[21] J. H. S. Pereira, P. F. Rosa, and S. T. Kofuji, "Horizontal Addressing by Title in a Next Generation Internet," *ICNS 2010, The Sixth International Conference on Networking and Services*, pp. 7–11, 2010.

[22] T. Bradley, C. Brown, and A. Malis, "Multiprotocol Interconnect over Frame Relay," *Internet Engineering Task Force Document IETF RFC 1490*, pp. 1–25, 1993.

[23] A. E. Joel, *Asynchronous Transfer Mode Switching*. Institute of Electrical & Electronics Enginee, 1993.

[24] J. Postel, "RFC: 793: DoD Standard Transmission Control Protocol," *Information Sciences Institute of the University of Southern California*, 1980.

[25] E. Rosen, A. Viswanathan, and R. Callon, *Multiprotocol Label Switching Architecture*. RFC Editor, 2001.

[26] J. Postel, "RFC 768: DoD Standard User Datagram Protocol," *Information Sciences Institute of the University of Southern California*, 1980.

[27] "Draft Recommendation X-25," *CCITT Study Group VII*, 1976.

[28] Top500.org, "Interconect Family Share Over Time," retrived 2011-01-10. [Online]. Available: http://www.top500.org/overtime

[29] I. T. Assoc., "InfiniBand Architecture Specification, Volume 1, Release 1.2," 2004, retrived 2011-01-10. [Online]. Available: http://www.infinibandta.org

[30] I. Myricom, "Myricom," retrived 2011-01-10. [Online]. Available: http://www.myri.com

[31] A. Boukerche, D. Ning, and R. B. Araujo, "UARTP - A Unicast-based self-Adaptive Reliable Transmission Protocol for Wireless and Mobile Ad-hoc Networks," *2nd ACM international workshop on Performance Evaluation of Wireless Ad hoc, Sensor, and Ubiquitous Networks - WASUN '05, Canada*, pp. 255–257, 2005.

[32] P. Djukic and S. Valaee, "Reliable Packet Transmissions in Multipath Routed Wireless Networks," *IEEE Transactions on Mobile Computing*, 2005.

[33] V. Jacobson, "Congestion Avoidance and Control," *SIGCOMM '88: Symposium proceedings on Communications architectures and protocols, USA*, pp. 314–329, 1988.

[34] G. Malva, E. Dias, B. Oliveira, J. H. de Souza Pereira, P. F. Rosa, and S. T. Kofuji, "Implementação do Protocolo FINLAN," *8th International Information and Telecommunication Technologies Symposium*, 2009.

[35] M. M. Alves, *Sockets Linux*. Brasport, 2008.

Delivery of CCNA as part of a Distance Degree Programme

Nicky Moss
Faculty of Mathematics Computing and Technology
The Open University
Milton Keynes, United Kingdom
n.g.moss@open.ac.uk

Andrew Smith
Faculty of Mathematics Computing and Technology
The Open University
Milton Keynes, United Kingdom
a.smith@open.ac.uk

Abstract—**This paper reports upon the success that The Open University of the United Kingdom has had in delivering the Cisco Exploration Curriculum, as an option in an undergraduate BSc degree, using a Blended Distance Learning Model. It is argued that a constructivist learning approach was taken when designing this course, which is demonstrated by this blended learning model. This is an important pedagogical distinction, as many would see the practical focus of this course as training. The importance of Supported Open Learning as a method of teaching students, and the key role of simulators, remote access tools and day schools are also discussed as contributors to the pedagogy. Bended delivery has proven to be an excellent way of delivering Cisco courses to adult learners, as supported by student feedback and attainment. Distance teaching offers the Cisco Networking Academy program an opportunity to extend reach in both existing and new markets.**

***Keywords*-Cisco Networking Academy; Blended Distance Learning (BDL); Supported Open Learning; Netlab; Constructivism; CCNA; Distance Teaching; Pedagogy.**

I. INTRODUCTION

The Open University of the United Kingdom (UKOU), as a member of the Cisco Networking Academy program, delivers the Cisco curriculum to students who study on-line at home. These courses provide them with degree level qualifications and the preparation necessary to take the Cisco certification examinations, which are widely accepted as an industry standard, offering students the opportunity to gain employment in the information technology and telecommunications sector. A shortened version of this paper was presented at the IARIA conference [1] in Cancun, Mexico in March 2010.

The UKOU has been providing higher education at a distance since 1969. At the time of writing it had 180,000 [2] students studying undergraduate and postgraduate courses, mainly in the UK, but also considerable numbers in Continental Europe and some other Countries. The faculty of Mathematics Computing and Technology also has a history of providing courses relevant to employer needs. In comparison, the Cisco Networking Academy currently has 470,000 [3] registered students in 160 Countries. Just in terms of outreach, both organizations have been successful in bringing education opportunities to very large numbers of people, and often to groups that do not have access to other types of education. Both have been successful in developing the courses and information systems to support their learners, and both provide courses that are valued by employers and employees.

The UKOU became a Cisco Networking Academy in 2003, and delivered its first CCNA (Cisco Certified Network Associate) course in 2005. Since then it has recruited more than 3000 students to study the CCNA, and is currently enrolling about 600 per year to study CCNA Exploration. The UKOU also started to deliver the CCNP (Cisco Certified Network Professional) curricula in 2009, and has already recruited 300 students to study the first CCNP modules. All UKOU students study the CCNA and CCNP using blended distance learning (BDL), or more precisely, a variation on what the University calls 'supported open learning'.

The UKOU has been a very successful University that has enjoyed growth, year on year, since it opened its virtual doors to students forty years ago. Much of the success can be attributed to the ability to offer learning opportunities to students who find it difficult to access traditional classroom based educators. For example, students in full time employment, those with family commitments at home, those in the military and those with disabilities. It is these same groups of students who have enrolled in the Cisco courses at the UKOU, and this has brought about 5% extra students to the CCNA program in the UK, at a time when the overall program appeared to have reached saturation.

This paper will expand upon the model of supported open learning (BDL) that the UKOU is using, arguing that the CCNA program is ideally suited for this form of delivery, and that BDL offers opportunities to extend the reach of the Cisco Academy Program to students in existing and new markets. Attention is also given to the experience of the students as learners, with consideration of the possible learning style being used by these courses. Specific reference is made to UKOU course T216, which delivers the CCNA.

II. CISCO NETWORKING ACADEMY

The Cisco Networking Academy was first established in 1997 with the specific aim of helping educators to develop a sustainable way to design, maintain, troubleshoot and updates their networks [4]. In line with the original ethos Cisco still provide a complete curriculum free to schools, colleges and universities that join the academy program. All of the teaching and assessment material is provided on-line via the academy VLE. The on-line material is content rich making extensive use of animations, rich pictures, inter-active quizzes and of course text. A typical page is shown in Fig. 1.

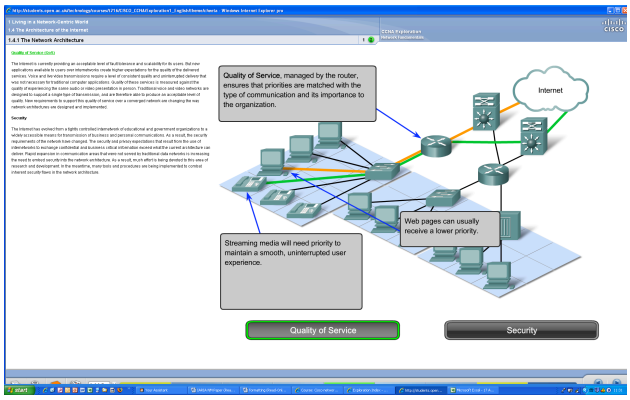


Figure 1. Example VLE page

Each page is normally divided with the text on the left and animation and other rich media on the right. Generally the text will explain an idea of concept, and the activity on the right will aim to deepen the student's understanding by engaging them. In this example a common network is used to illustrate two views, one focusing on quality of service, the other on security. Users can select either view using the active buttons, enabling them to compare the differences when applied to the same base network.

Laboratory activities are provided at the end of each chapter to enable the student's to develop understanding through a series of practical exercises. Many of these can be carried out using a simulation tool called Packet Tracer that is explained later.

There are also on-line tests and exams that allow students to assess their progress. These also provide feedback and direct students to the relevant part of the curriculum.

As well as the assessment that is built into the courses Cisco also provide a series of certification examinations that can be taken at local testing centers. Students who successfully pass these exams gain a qualification that is widely recognized by employers as they provide evidence of networking competences that are directly relevant to the work place. Cisco CCNA and CCNP certification is highly valued in the workplace.

III. SUPPORTED OPEN LEARNING

The style of distance learning used by the UKOU is often described as supported open learning. The key features and pedagogical aspects of this style are described below, followed by the specifics of how this style has been adapted in the case of students' studying the Cisco Networking Academy program using BDL.

A. The UKOU Model

In the UKOU model students study at a distance, normally at home in their own time, using material provided by the University. Course related support is provided centrally by the University and by the student's own tutor. The materials the students' use for their studies can be broadly divided between teaching and assessment. Teaching materials make up the bulk, can be either electronic or print, and are often a mixture of both. Most of this material is produced in house by the University, although some third party material, such as books, journal articles, video or software is used. Teaching texts, books and DVDs are sent to the student's home, and on-line materials are accessed via the student's home page, using the usual password access controls.

The front page of the student's course home page is shown in Fig. 2.

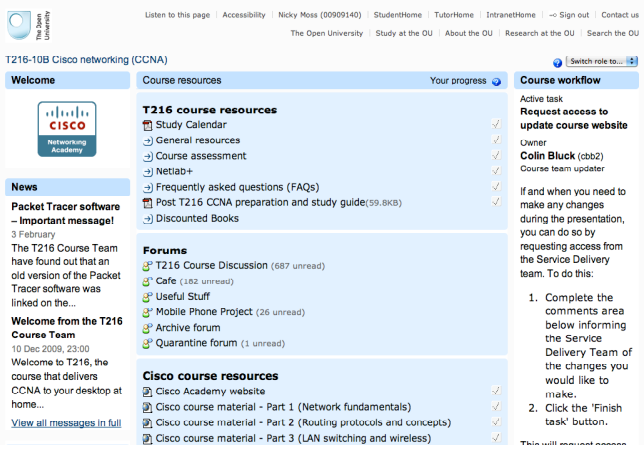


Figure 2. Student home page

The home page provides access to all of the course components, for example, a course calendar, the resource centre (library), course assessment, the academy curriculum and links to download Packet Tracer. Students can also

login to a discussion forum and follow various links to useful University wide resources via link at the top of the page. This will include formal rules about assessment or raising issues, for example.

Each group of 20 students is allocated to a dedicated tutor, who in the case of students studying the Cisco program is also a qualified Cisco instructor. Each group share a class within the academy. Tutors support the students with their study by managing the academy assessments, providing on-line and telephone support and by providing feedback when they mark each student's work.

Broadly three types of assessment are used, tutor marked assessment (TMA), computer marked assessment (CMA) and examinations. Each course will have more that one TMA or CMA and a single exam at the end.

The TMA is piece of written work that is completed by all students on the course. The work is submitted by the student to the University using an electronic handling system, and is marked by their tutor, and returned via the same system. Marked work is returned to the student with personal written feedback provided by their tutor. Marks are collected centrally for assessment purposes. The TMA provides a good opportunity for students to complete an extended piece of course work, one that tests both their theoretical grasp of the subject and their practical skills. Fig. 3 shows a network that has been used to tests students skills to plan, implement and test a network through a scenario.

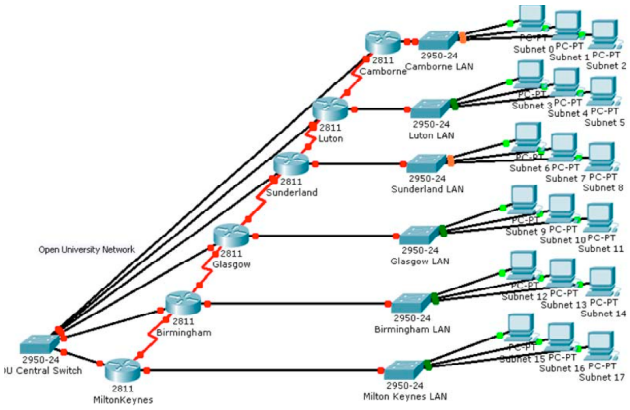


Figure 3. Network used in TMA

The network in Fig. 3 includes routers, switches and PCs. Students are provided with a scenario that states the organizations requirements and gives basic address ranges. Using this information students have to plan and configure all of the network devices and implement them using Packet Tracer. They are then required to demonstrate compliance by running various testing parameters. Their completed answer will have to be supported with written evidence to demonstrate their thinking as well as their implementation skills. Scenarios of this type are typical of what may be encountered in a work situation and for this reason provide an excellent test of student progress.

The CMA takes the form of multiple-choice questions that are completed by the student on-line. Marks and feedback are provided to the student immediately if these are used formatively, and after a common cut off date if summative.

The final examination can take one of two forms, an extended piece of course work, which is managed as the TMA, or a formal written exam held in an examination centre local to the student's home. Students are given an overall course result once the exams have all been marked. Students are free to go on and take the Cisco certification at any time.

Guidance on studying course material is provided using an electronic calendar that provides all key dates, especially the cut-off dates for the various forms of assessment, recommended start and completion dates for individual modules and the dates for day schools. The calendar is however only a guide to student study patterns, as flexibility about how and when students study is essential for those in work or with demanding home lives. Students are provided with general study support via their university and course specific home pages. Additional course specific support is provided via on-line forum, moderated by professional teachers (tutors). Students can also call upon their tutor for support using e-mail or phone, and tutors can use their own home page to monitor their student progress and take action pro-actively.

The University also has a long history of providing stand- alone week residential schools (called summer schools) for many of its courses, especially those that are science or technology based. These are now less common, as modern on-line tools and simulations have provided good alternatives for these summer schools, even in subjects such as engineering [5]. When studying T216 students attend four separate day schools.

B. Blended Delivery of CCNA

There are some obvious parallels between the way the CCNA curriculum is delivered through the Cisco Networking Academies and the UKOU's supported open learning model. Looking within the Cisco CCNA program for parallels, these include student home, an on-line curriculum, the use of simulation tools such as Packet Tracer, and on-line assessment, both formative and summative. The one obvious difference is that the CCNA has mostly been delivered in a classroom setting, one where

a teacher can guide students through the curriculum and labs. The experience of the UKOU suggests that classroom need not necessarily continue to be the de-facto option, although it will continue to be the dominant for most students.

Teaching the practical skills using real equipment is an essential learning outcome for the CCNA curriculum. The integrity of the final examinations is important for maintaining the credentials of the program. Maintaining both of these features is therefore critical, even if blended teaching is used. Both of these provided a challenge for the UKOU, where normal practice is for students to take much of their formative assessment at home unsupervised, and when the use of residential schools was diminishing as a result of advances in on-line labs. On the other hand, the ordered structure of the curriculum and the end of chapter tests, both fitted naturally with the flexible timetabled teaching used on other courses. Fig. 4 shows some of the assessment pages from the academy VLE.

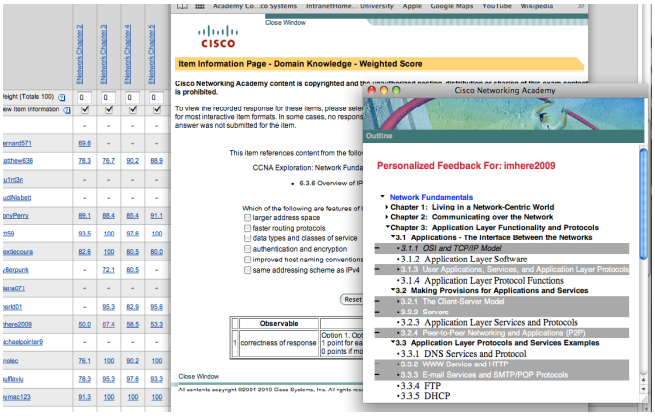


Figure 4. Assessment pages on VLE

Each student has a gradebook that shows all of their marks for the assessment taken within the academy class. The example in the background is the tutor version that shows the results for all students in that class. Students (and tutors) are able to review the results of any question taken, and see the question in full, with the answer. This is shown in the second pop up window. The feedback directing students to the curriculum areas they need to revise, based on their individual performance in a given exam, is shown in the third pop up.

This comprehensive assessment and feedback system allows students to reflect and learn from the results of their exams. As tutors can see the performance of the whole class, it also allows them to target additional teaching to areas of shared difficulty.

The final solution, that enabled the UKOU to make full use of its experience in supported open learning and meet Cisco's requirements for hands-on practical and proctored final exams were achieved with the use of dedicated day-schools and Netlab [6]. The opportunity for students to develop and practice their skills with configuring networks has also been enhanced by the rapid developments of Packet Tracer. How the UKOU has used each of these elements to deliver the CCNA Exploration curriculum is explained below.

1) Day Schools

Students who wish to study the CCNA Exploration courses with the UKOU can only do so as part of an undergraduate degree program. Currently all four CCNA Exploration courses are offered as a single undergraduate course titled Cisco networking, given the designated university code T216. Because this course is part of a degree program students are expected to have some prior knowledge of networking computers, their use in the workplace and basic study skills; what is termed experienced learners in the Cisco Academy.

On the understanding that our students were experienced learners, together with recognition that T216 would also include Netlab, it was agreed with the UK Cisco Networking Academy managers that there would be four days dedicated to practical skills development. As UKOU students live all over the country, it is not practical to get them to all attend one centre, so students are given a choice of dates and venues. Generally each day school follows the completion of one part of the CCNA (there are four) as this allows maximum use to be made of Packet Tracer and Netlab to prepare the students for the day, enabling them to gain maximum benefit from getting to work with real equipment.

Partnerships have been established with seven Cisco Networking Academies in the UK and one in the Republic of Ireland to deliver the four schools. This co-operation has brought benefits to both students and academies. Students can now attend day schools closer to their homes, they are taught by experienced Cisco qualified instructors, and in some of the best equipped UK academy labs. The academies have gained extra business on a Saturday, which is not a normal teaching day in the UK, allowing them to use facilities that would normally be dormant, leveraging extra benefit from the investment in networking equipment needed for teaching their normal academy students.

Students are able to book each of their day schools, from a selection of venues and dates, using an on-line booking system developed from the normal UKOU residential management system that now allows for four separate days. This system also feeds an attendance mark, necessary to

check the student meets the course requirement for compulsory day schools, to each student's assessment record. A written handbook is produced for each day school setting out the learning outcomes and activities to be carried out. This is supplied to all students and day school centers, and aims to ensure that all students gain a similar learning experience.

2) Netlab

The Netlab Academy Edition provides remote access to Cisco networking equipment such as routers and switches. It has been specifically designed by NDG to host Cisco training equipment on the Internet for student and instructor use, and is particularly well suited for blended distance learning [7]. It is important to remember that Netlab is not a simulator, and allows students to access the console ports of real networking equipment, such as routers and switches. Once logged into a booked session the users sees a topology such as the one shown in Fig. 5.

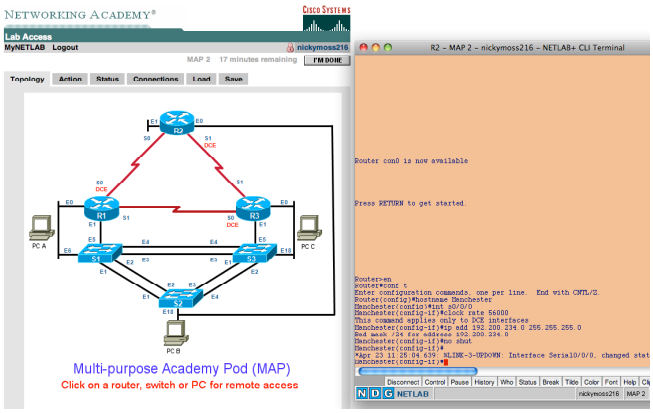


Figure 5. Netlab topology

This topology is a popular multi-purpose configuration consisting of 3 routers, 3 switches and 3 virtual PCs. This allows a wide range of scenario to be investigated by students. The right hand screen is the command line interface for one of the network devices, in this example router 1. It is this interface through, which the users set up the various devices. The script in the window is a list of commands to configure the router, and it is proficiency with the full range of Cisco commands that is one of the key learning outcomes in this program. A window can be opened for each of the nine devices shown in the topology, enabling the whole network to be configured. Netlab will save students configuration files so that they can continue with their work in another session.

All UKOU academy students are given access to Netlab for the full duration of their study, normally 9 months. Student's accounts on Netlab are organized in tutor groups to enable tutors to monitor student use and lead teaching sessions as necessary. Some will have accounts on the UKOU's own Netlab, others will use systems belonging to our partner academies, who lease access to the UKOU. Student access is provided 24/7 using the self booking facility provided by the system.

Students can access Netlab at any time to undertake labs as specified in the curriculum, or just to practice and develop their configuration skills. All students are required to use Netlab and not to rely entirely upon Packet Tracer. Activities specific to Netlab are included in the UKOU's assessment to ensure that students complete practical work that can be assessed by their tutor and counted towards their assessment score.

3) Assessment

Students study the Cisco Exploration curriculum in the recommended order, starting with Network Fundamentals and finishing with Wide Area Networks. Students take all the chapter exams, normally at their own pace, although working within certain limits set by the study calendar. Their practical work is assessed at the day schools and through specific additional activities using Netlab and Packet Tracer. Each day school is scheduled to take place when all students have completed each course. For example, the first day school is at the end of the study period allocated to Network Fundamentals. Students also take their Cisco final examinations at the day school. Students who successfully graduate from each Cisco course gain the appropriate certificate and/or letter from the Cisco Academy.

The UKOU awards credit towards a BSc Hons degree to all students who complete the four Cisco Exploration courses and pass the additional assessment set by the university. This assessment consists of five assignments (TMA) taken during the course, and the final written examination. Students must also gain a satisfactory attendance for each of the four, day schools. Successful completion of this course gives the student the equivalent of ¼ of a full years graduate study.

Each TMA is completed by all students and submitted to the same deadline. All students complete the same tasks in the TMA, which is then marked by their tutor. A range of question types are used, for example, written explanations, sub-netting calculations, Netlab activities and network design and implementation activities using Packet Tracer.

The final written examination lasts for 3 hours and draws upon the entire CCNA Exploration curriculum. Again a full range of questions are set that aim to test the students understanding, by asking them to explain, calculate and problem solve under closed book examination conditions.

The combination of Cisco Academy exams and the OU assessment provides a well-designed assessment strategy for the students. Assessment has long been seen as an essential part of teaching and learning [8], especially when it plays a vital part in getting the students to engage with the study material and keep them motivated. All students are encouraged to take the CCNA certification exams and full use is made of the preparation exams in the gradebook. Anecdotal data suggest that those that do well in the course go on and gain the certification exam.

4) Supported Learning

In many respects the CCNA curriculum is ideal for students to study on their own at a distance. For example, all of the teaching material is on-line, so easily available at home or work, it has embedded simulations and activities that engage the students, it can be studied linearly without teacher direction, also Packet Tracer can be used to develop practical skills and there is assessment with feedback, which allows students to assess their own progress.

Unfortunately, providing students with easy on-line access to good study materials, with optional access to tools and assessment does not often lead to successful study. Technology alone is not sufficient [9], and students benefit immensely from a learning environment that offers support and fosters ambition to learn.

A central feature of the UKOU's supported open learning model is the role of the tutor (associate lecturer). Each student is assigned to a tutor group, nominally with 19 other students. Tutor groups are based on the student's geographical location, and this allows for face-to-face meetings, although these are not central to the teaching model. Tutors will make early contact with their students using e-mail or telephone. Students will also receive their login information for the academy, login detail to their OU home page and a welcome letter from the chair of T216. Together these contacts should give the student a sense of belonging, in some ways similar to their first day at a conventional college. Students will also have contact details for their tutor on their home page, and are free to contact her if they have any queries.

During the first two weeks of the course students are allowed time to explore the UKOU learning resources and familiarize themselves with the academy site and materials. Additional study materials have been produced to assist the students in getting to grips with the basics of the academy gradebook and Netlab. A local face-to-face session is also arranged where each tutor can meet their students and go through all of the on-line learning materials and tools. Very few students have any difficulty in getting on with studying the course once they have reached this point.

Additional study support is provided through a national on-line forum. This is open to all students studying the course, and is primarily intended as self-help, where students are encouraged to exchange ideas, ask each other questions, and generally build up a sense of belonging to the UKOU and the Cisco Networking Academy. The forum is moderated by tutors, who provide an input to discussions when necessary, perhaps to correct a thread started by a student that is re-enforcing misinterpretation, or just giving wrong information. They also ensure, through a light touch that students behave appropriately in their exchanges with other members of this on-line community, an example of an exchange is shown in Fig. 6.

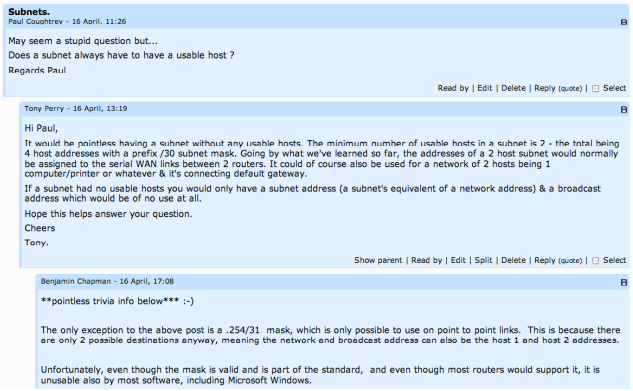


Figure 6. Forum discussion

This is a good example of a thread where one student is asking a question and other students are providing the answer. This self-help benefits all students - the one that asked the question because he gets a quick answer, within 2 hours in this case; the students that provide the feedback as they are rehearsing and developing their own understanding by putting it in their own words; and the many students who were stuck on the same point but had not asked. This is an example of a short thread, and there are many that extend to 20 or more replies. Moderators regularly review the forum, mainly to make sure that no thread is developing in a way that would be misleading to the students, but also to answer more specific questions such as clarification for assignment questions. By using the forum for this type of interchange between tutor and student all benefit from the answer and not just the one that asked.

All tutors are also trained academy instructors, and each one has an academy class with the same membership as their tutor group. This means that each tutor can see the progress of their students by checking their chapter exams in the academy gradebook. The tutor's own home page on the UKOU site also has the progression and assessment particular to the university study path. This information allows tutors to be pro-active in supporting their students if they are falling behind, or having other difficulties with

their studies. Tools on their homepage also allow tutors to send e-mails to all or some of their students as they choose. This provides a very easy means of contacting groups of students, for example, a sub-group that all had difficulty with a particular set of questions in an end test.

Students have to submit a TMA about every six weeks. This process establishes a dialogue between each student and their tutor that is particular to that student at that time. The student's performance in the TMA will give the tutor a clear idea of how he is progressing. This will allow the tutor to tailor the feedback to the needs of that student. Some examples of feedback include explanation of sub-netting, or the suggestion to try a lab again, or perhaps just reassuring the student that they are coping with the course, or explaining what might happen at a day school if they express some anxiety.

There is good synergy between the Cisco Networking Academy and the UKOU as all tutors are also qualified Academy Instructors, and many of these also teach at day schools. As a result of this partnership the UKOU now employs more than 30 academy instructors on a part time basis, and role that most see as an enhancement to their CVs.

Following the success of delivering the Cisco Academy, as distance learning courses at undergraduate and post graduate level, the UKOU has started to include material from other vendors in our degrees. Vendor related courses being delivered, or planned to start soon, include Linux, Microsoft and VMware.

5) Packet Tracer

Packet Tracer provides students and teachers with a vast range of learning opportunities, from helping students to learn the basics of configuring routers, through to the design, implementation and fault finding of complex internetworks. The UKOU has used Packet Tracer extensively, both by actively encouraging students to attempt all of the labs, and by including scenarios as part of the TMA assessment as described earlier.

Packet Tracer is a very powerful simulation tool that enables users to build, configure and test many types of networks from one containing only two devices joined by a single cable, through extensive networks containing many different types of devices and connection types and finally networks containing many sub-networks. Perhaps the limit is only the imagination of the user.

An example screen taken from Packet Tracer is shown in Fig. 7.

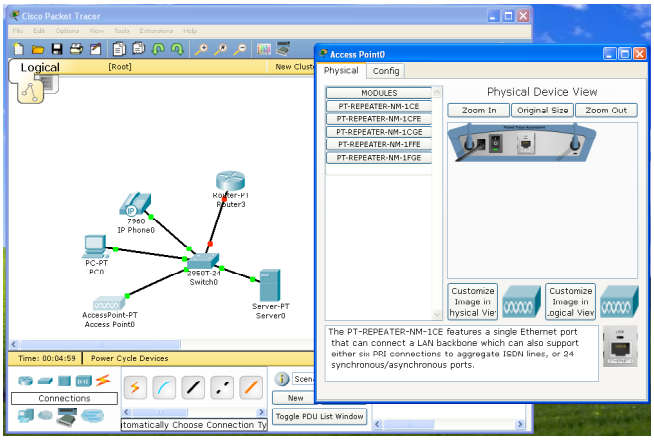


Figure 7. Packet Tracer screen

A network layout is shown on the left hand side of the figure is easily constructed by the user dragging the appropriate icons onto the screen and joining them with the appropriate connection types. In this example a router, switch, server, wireless access point and an IP phone are shown, variously connected using serial or twisted pair cables. Once a device is put on the desktop it can be accessed by a window that gives access to the command line interface and other choices. In the figure a physical representation of the device is shown. This allows the user to carry out function such as power on or off. This may at first seem a little artificial, but actually replicates the requirement in the real world of taking account whether equipment is powered before connecting or disconnecting cables. There is also a rear view of the device, which is important as recognizing a device from the rear is necessary when making connections. Packet Tracer also has an advanced facility for tracing data packets through the network, allowing users to analyse protocols at different points in their travel across the network.

It is important to fully appreciate how powerful a learning tool Packet Tracer is. While it is a simulator it does support the full command line for many devices and therefore offers students a very real experience when building networks. This effectively gives students the opportunity to build networks that would cost many thousands of dollars using real devices, and explore scenarios that would only normally be within the scope of the largest networking companies.

IV. STUDENT LEARNING

Having looked at the essential elements that are available to the students to learn about networking and seeing how these are combined into a blended distance learning package, it is important to consider how this blended delivery of the CCNA can be understood in terms

of learning theory. For example, is the pedagogy predominantly instruction based, that is where student's mainly focus on what is presented to them, checking their understanding through tests that confirm whether they got the right or wrong answer. Or is there evidence of deeper learning, where students engage with the learning materials in a way that enables them to reflect upon the goal and their action in working towards it.

It will be argued that the latter is the case, and that there is evidence of deeper learning that may be explained in terms of constructivism [10]. There is also evidence of socio-cultural learning through forum and collaborative learning, primarily at day schools.

It is equally important to find out how the students view the learning experience and how well are they think they are learning? In this respect student feedback can confirm, or provide contrary evidence, that they are learning at a deeper level, although it is unlikely that students would express their experience in terms of learning theories, but they are likely to use the kinds of language that can be associated with a particular theory.

A. Learning Theory

The CCNA curriculum uses a media rich presentation of the learning materials together with self-assessment activities and end of module on-line multiple-choice examinations with personal feedback. The UKOU also provides additional teaching material and assessment in the form of written tutor marked assignments. In summary these can be divided into two classes, on-line study materials and assessment. In addition to these teaching and assessment material there are also practical activities in the form of labs, some embedded in the curriculum and others associated with assessment. There are also the intensive day schools and supported learning provided by tutors and the forum. The division of study time between learning materials, assessment and labs is approximately even, that is 1/3 of the available time on each. Although a distinction is drawn between the different classes of activity it is important to recognize that the essence behind supported learning is to make sure that all of the activities are integrated together, and it is the context of the overall course that learning should be judged.

In the context of learning the key question is whether learning is what is termed surface or deep. In the former the student's understanding is typified by rote learning, with success being measured by them knowing the correct answers to questions that they encounter in the course of their study. This can be referred to as instuctionalism, where the priority is for the teacher to deliver and correct the students. In the latter a deeper understanding is achieved by engaging the student actively in the goal and feedback path so that they can not only judge right and wrong, but also see how they can work towards the goal. This process leads to students who can apply what they have learned to novel situations, for example. In some ways more importantly, they actually learn to learn for themselves.

Constructivism is a theory that can be traced back to Piaget [11], although there has been significant development in thinking about this topic since [12], including the application of the theory to distance learning as well as the more conventional face-to-face situation. Constructivism proposes that learning is achieved through the forming and modification of internal mental representations, and in this context encourages teaching to focus in a way that draws on various cognitive processes that underlie learning. Significantly, this approach allow learners to accommodate their new learning into the context of their existing view of the subject, or even more broadly into their general view of the world, which is particularly pertinent if their newly acquired learning is to be applied in the workplace. A central tenet of constructivism is to get students to take responsibility for their learning, what is termed active learning. This requires a shift in the role of the teacher towards facilitation.

It has been argued [13] that a teaching approach based upon constructivism alone may not be the most effective, that the transmission of knowledge is still a key overall part of learning. The authors' believe that the components of this BDL model provide this balance between giving the necessary knowledge and providing them with the tools for the cognitive integration of this knowledge. For example, the on-line curriculum sets out the basic knowledge for student to build upon and creative assessment, with feedback, informs students whether they are gaining a deep or shallow understanding. For example, the feedback on the multiple choice questions in the Cisco academy allows students to try the question again before giving them the answer. It also directs them to the relevant learning material so that they can revise a topic to see where they have gone wrong. The tutor marked assignments, provided by the UKOU, set challenging practical based scenarios, where written individual feedback is provided with the specific aim of helping students understand where they have gone wrong, not just correcting their answer. A certain percentage of the feedback given is of a type called feed-forward, which is more general and specifically aims to help students with general learning that they can apply to future assignments.

The taking of labs, using packet tracer or Netlab, challenge students and build their confidence, as they put the knowledge they have gained from the curriculum into practice. One of the key features of the labs is the extensive practice using the command line interface. Students need to

become fluent with a wide range of commands that enable them to configure various networking devices such as routers, switches and servers. Configuring a network is not simply a matter of following a menu, the choice of routing protocol, addresses and other network parameters needs to be worked out before choosing a particular sequence of commands to achieve the planned aim. If an error is made it is then necessary to carry out detailed fault finding diagnostics to identify the error and take corrective action. It is this combination of using knowledge, planning, implementing and testing that take students repeatedly through the loop between the task, their conception of the task and the learnt concepts that develops their deep understanding. It is this combination of theory and practice that allows the claim that a constructivist learning is taking place.

On-line forum allow students to exchange ideas and support each other, and day school activities are organized around group activities. These two provide opportunities for the social and co-operative aspects of learning. Tutors and conference moderators take on the role of a facilitator as they guide student discussion within the forum and help them to clarify their thinking when they are experiencing difficulty with some aspect of learning or doing labs. The challenging network scenarios in TMAs push students to consolidate their learning in the context of real network design problems. Overall, this combination of components ensures that independent learners become successful in achieving the learning outcomes of the course.

More work is required to fully understand and support the claim that courses like this one, courses that have an emphasis upon the development of practical skills, lead to deep learning because the underlying pedagogy is constructivist. In fact many educators may well be surprised, and even disagree, with this argument, as many have dismissed the CCNA as training, and therefore not really fit for inclusion in a degree program at all. The emphasis upon training would definitely point towards instructionism as the underlying pedagogy. The point, which a crude classification between training and education misses, is the level of knowledge that is needed to carry out some of the practical activities and the recognition that situating these activities in real world examples enhances the students ability to recall and use their knowledge in other real world situations, exactly the goal of constuctivism.

B. Student Achievement and Feedback

Students were surveyed during the 2008 presentations of T216, and 71 students responded. The statements in table I compare student satisfaction with the normal for all level 2 ICT (information communication technologies) courses.

TABLE I. STUDENT SATISFACTION

Student Satisfaction Question	T216	ICT
Overall, I am satisfied with the quality of the course.	94.4%	88.8%
The course met its stated learning outcomes.	97.1%	88.5%
I enjoyed studying this course.	91.6%	81.2%
The workload on this course was higher than I expected	66.2%	37.9%
I would recommend this course to other students	91.6%	82.1%

As can be seen from the first three entries in table I, students rated the course very highly in terms of their satisfaction with quality of the course (94.4%), the fact that it met the stated learning outcomes (97.1%) and that they enjoyed their study (91.6%). These are very good scores, especially when set against the fact that the UKOU is one of the highest rated Universities in the UK in terms of student satisfaction.

The fourth entry in the table did show a higher percentage (66.2%) of students who stated that the workload was higher than they expected. Discussions in the forum suggest that this may be because of two reasons. First, because of the frequent formative assessment, students are continually checking their understanding and reviewing topics where they have not gained a high score in the exam. Second, because of the very large number of labs that must be completed if students are to become proficient with the practical skills. Although rated with a higher workload, overall student performance is good, with many students gaining high marks.

Perhaps the last entry is the most significant, with 91.6% of students saying that they would recommend this course to a friend.

During the survey students were also asked to make their own comments. These were then collated and other students were asked to respond, either as agreeing or not agreeing with the statement of another. Table II show student's ratings for statements generated by other students.

TABLE II. STUDENT RATINGS

Student Question	Response
TMAs and continual assessments were essential to keep studies on track.	Mostly agree.
The simulation software Packet Tracer was excellent.	Definitely agree.
The combination of Cisco Academy material with OU's support material and assessments worked very well for me.	Definitely agree.

The responses to statements generated by other students are very insightful as it states what is important to them, rather than asking what is important to the teacher or the institution. Definitely agree is the highest endorsement that

student's can give, and they valued the general approach taken by the UKOU, that is a combination of Cisco and OU teaching and assessment. The success of this approach is further endorsed by support for packet tracer and the day schools. It can be seen that not every student agreed that assessment was essential to keeping on track with their study, but that is not surprising as few really look forward to assessment even if it does enhance their learning overall.

Perhaps the best overall feedback is the type shown in table III, sent unsolicited from a student at the end of 2009.

TABLE III. STUDENT FEEDBACK

By passing the course I managed to pick up a degree and a new job - yipee. No longer am I a technician in the Royal navy but now working in industrial networking for a company called GarrettCom - they manufacture industrial strength switches, routers and media converters. The networking theory is the same but the command lines are different. Hopefully I can find the time to start CCNP next year.

This student's comments neatly summarise the strength of the Cisco CCNA when combined with distance learning. This student has had a full career in the navy and has prepared for the day that he has to leave by studying towards a degree with the UKOU. His success is testimony to the flexibility of this study method, as it cannot be easy for a full time member of the armed forces. His choice to complete his degree with the CCNA has provided him with both the academic qualifications and the skills to start a new career in networking.

Overall, students are successful with their study of UKOU course T216, with most completing all four Exploration modules and passing the final exam. These students generally also go on to take the UKOU examination and gain credit towards their University study. Based upon anecdotal evidence from the student forum, a significant number go on to take the CCNA certification. Some claimed to have passed with a mark of 100%.

Students who study with the UKOU are adults (over 18 years), and more than 75% on T216 are over 25 years. Most are studying to further their careers. These factors give this group a high level of motivation, and they may do better with this type of learning than other groups.

V. CONCLUSION

The decision taken by the UKOU to offer the CCNA curriculum as a blended distance course has been rewarded with high student numbers and good student feedback. This success has shown that this model of delivery is well suited for adult learners, and may well be suited to all learners. It must however be recognized that BDL involves much more that just enrolling students and offering them access to the Cisco curriculum. It is vital to support learners in a way that facilitates learning, and to make maximum use of Packet Tracer and Netlab to develop student's practical skills. Day schools are also essential as they give students the chance to get their hands on real equipment. Overall, it is the management of the students and resources, in a way that facilitates active learning, that lead to successful students. On-line forum, good information systems and tutor support all play an important role in this management. A well planned and managed BDL form of the Cisco Academy Program offers an opportunity for educators to reach new students in established and developing markets.

Some arguments have also been put forward that a course that focuses on teaching practical skills as well as knowledge does engage students in deep learning. Particularly, that the pedagogy that underpins this course is constructionist. A conclusion that may surprise many educators who have tended to draw a line between education and training, and may have neglected to give credit for the situated nature of practical learning and hence its contribution to the overall learning of the student.

The UKOU looks forward to building upon the success of the CCNA as they move forward with a Masters qualification built around the blended delivery of CCNP.

ACKNOWLEDGEMENT

The authors would like to thank the many people who have worked with them in establishing and running this BDL version of CCNA. They are too numerous to mention individually, but include people from Cisco UK, CLI, the UKOU, the UK central academy and our other day school partners.

REFERENCES

[1] Moss, N. and Smith, A. (2010). Large Scale Delivery of Cisco Networking Academy Program by Blended Distance Learning. LMPCNA 2010, March 7-13, 2010 – Cancun Mexico.

[2] The Open University. http://www3.open.ac.uk/about/ Accessed on 22/12/10.

[3] Cisco Networking Academy. http://www.netaced.net Accessed on 22/12/10.

[4] Hernandez-Ramos, P et al. (2000). Changing the Way We Learn: How Cisco Systems is Doing It. International Workshop on Advanced Learning Technologies, December 4 – 6 Palmerston North, New Zealand

[5] Bissell, C.C. and Endean, M. (2007). Meeting the growing demand for engineers and their educators: the potential for open and distance learning. Meeting the Growing Demnds for Engineers and Their Educators 2010-2020, Munich, Germany, 9-11 November 2007.

[6] NDG. http://www.netdevgroup.com/home.htm Accessed on 22/12/10.

[7] Prieto-Blázquez, J. et al. (2008). An Integrated Structure for a Virtual Networking Laboratory. *In IEEE transactions on Industrial Electronics*, Vol 55, no 6, pp 2334-2342

[8] Papert, S. (1991). Situating constuctivism. In Harel I & Papel S (Eds.), Constuctivism: research reports and essays, 1985-1990 (pp 1-11). Norwood, N.J: Ablex Publishing Corporation.

[9] Heap, N.W., Kear. K.L. and Bissell. C.C. (2004) An overview of ICT-based assessment for engineering education. European Journal of Engineering Education, Vol 29, no 2, pp 241-250.

[10] Laurillard, D. (2002) Rethinking University Teaching: a framework for the effective use of educational technology (2nd edition) London, Routledge Falmer.

[11] Piaget, J. (1978). Success and Understanding. London: Routledge & Kegan.

[12] Brophy, J. (2002). Social Constructivist Teaching: Affordances and Constraints. Oxford: Elsevier Science.

[13] Laurillard, D. (2009). The pedagogical challenges to collaborative technologies. Computer-Supported Collaborative Learning (2009) 4:5–20.

A Novel 3D-Based Network Simulation Platform for Wireless Indoor Networks

Mikko Asikainen, Mauno Rönkkö, Keijo Haataja, Pekka Toivanen
School of Computing, Kuopio Campus
University of Eastern Finland
P.O. Box 1627, FI-70211 Kuopio, Finland
E-mail: {mikko.p.asikainen, mauno.ronkko, keijo.haataja, pekka.toivanen}@uef.fi

***Abstract*—In this paper, a novel 3D-based network simulation platform for wireless indoor networks is proposed. The purpose of the platform is to improve the accuracy of radio propagation modeling and to offer designers a virtual workspace for testing their systems. Radio propagation models are investigated by performing path-loss calculations and Fresnel zone geometry estimation in a research laboratory environment.**

***Keywords*-3D Locationing; Sensor Networks; Simulation; TOSSIM; ZigBee.**

I. Introduction

This paper is an extended version of our conference article [1] containing additional backround information, extended testing and a more comprehensive explanation of the proposed simulation environment.

Sensor networks are comprised of small wireless devices capable of sensing their environment and dynamically communicating with their neighbors [2][3][4]. These networks are characterized by short link distances, limited system resources and low battery consumption [5][6]. Popular applications include intelligent home automation, monitoring and short range wireless communication [7][8][9].

Sensor network technologies are deeply rooted to their hardware platforms [10][11], limiting the things a developer can do before getting access to the actual devices. Testing programs designed for sensor networks without simulation becomes difficult due to the need of deploying actual hardware to the target environment during testing [12]. A versatile and efficient simulator would definitely benefit the research and development community. The simulation of sensor network programs is also important for the purposes of determining the requirements for hardware before committing funds for prototyping on actual devices, which might normally be unsuitable for the task or the amount of nodes in the prototype might not be sufficient.

In our previous research work [13][14], we examined the TOSSIM simulator [15][16] packaged with the TinyOS system [17][18]. We found out that it was a useful tool for sensor network developers, but lacked simulation of radio propagation, path-loss and small-scale indoor fading. Moreover, the simulation of ad-hoc networking was not supported at that time [19]. As a continuation of our previous research work, we started to investigate different methods of modeling path-loss in an indoor environment and to design a simulation environment that would best serve the needs of sensor network design and administration.

Modeling radio wave propagation is a mature field and closely related to acoustics. For example, one model that will be discussed in this paper, Rayleigh fading, dates back to 1880's when Lord Rayleigh observed the behavior of sound propagation of an orchestra [20]. Even though the principles of radio wave modeling are a mature field, finding novel applications for old models is relevant to today's science. In the field of simulation, a more complex combination of suitable propagation models can be used, since computing capabilities are increasing rapidly.

The rest of the paper is organized as follows. Section II gives a brief overview of ZigBee and wireless indoor communication. Path-loss modeling in sensor networks is covered in Section III. The section also contains results of some experiments we performed on a simple log-distance path-loss model to investigate the relation between the model and in-lab reality using ZigBee-enabled Wireless Personal Area Network (WPAN) devices. Section IV describes the Fresnel zone equation, which can be used to determine Line-Of-Sight (LOS) between two transmitting radios. Tests were performed based on this equation. The potential of using the Fresnel zone equation to improve path-loss calculations is discussed and the need for a 3D simulation environment to achieve these benefits is explained. Different problems related to 3D simulation is discussed in Section V. A novel 3D network simulator, which can act as a sandbox for network developers and administrators, is proposed in Section VI and concrete results of the experiments that are relevant to the proposed simulator are also discussed. This proposal concerns a prototype in an early design stage. Moreover, essential functions and possible technologies are discussed in the section. Finally, Section VII proposes some new ideas that will be used in our future research work and concludes the paper.

II. An Overview of ZigBee Communication

ZigBee [21] operates in the 2.4 GHz band with maximum transmission speed of 250 kb/s. It uses 16 channels ranging from 11 to 26. Each channel uses 5 MHz of bandwidth. The center frequency of each channel is $F_C = (2405 + 5 \times (k - 11))$ MHz, where $k = 11, 12, \ldots, 26$. ZigBee is based on the Direct Sequence Spread Spectrum (DSSS) technique and Offset Quadrature Phase Shift Keying (O-QPSK) modulation.

The ZigBee node listens to a chosen channel before transmitting. If the channel is occupied, the node waits for a random amount of time. After this waiting period, the node listens to the channel again and if it is free, the node can transmit data. This technique is called as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).

When the data has been sent to the destination node, it sends an acknowledgement message to the source node to confirm that the data transmission has been successful. If the source node does not receive the confirmation in a certain interval, it resends the data to the destination node. Moreover, in ZigBee mesh topology, the node can send message through an alternative route if the first route fails.

In our research laboratory experiments we used Sensinode Nanorouters and Nanosensors [22] as the first hardware set and Texas Instruments CC2530 evaluation boards [23] as the second hardware set. Figure 1 illustrates Texas Instruments' devices used in the second hardware set.

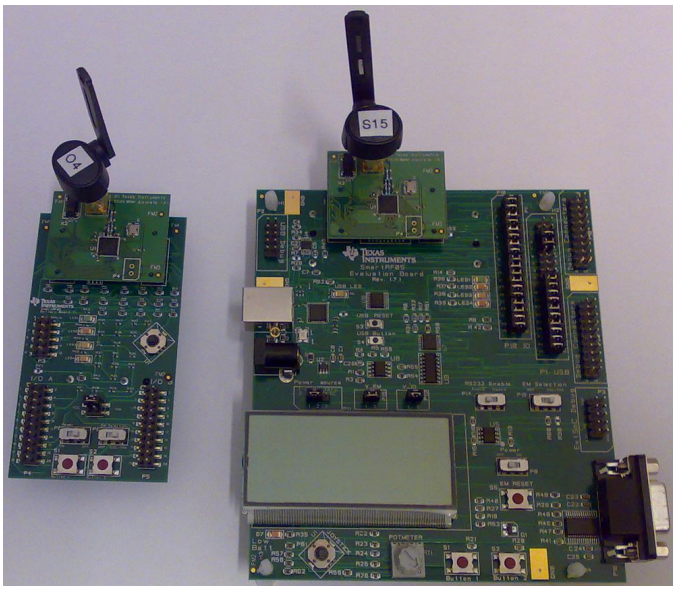


Fig. 1: Texas Instruments CC2530 evaluation boards [23] were used as the second hardware set in our research laboratory experiments.

ZigBee was chosen for our testing technology, because it is a good representative for modern wireless personal networks and it includes a lot of features to test in our research work.

III. Path-Loss Modeling

Radio transmissions traveling through space disperse and attenuate in a manner that can be, to some extent, predicted through path-loss modeling. Under ideal conditions, such as a LOS vacuum with no interference, radio propagation can be modeled by using the Friis free space equation [24]:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \quad (1)$$

In the equation, $P_r(d)$ is the strength of the received signal at distance d from the transmitter, P_t is the transmitted signal strength, G_t and G_r are the antenna gains at the transmitter and receiver, λ is the transmission wavelength, and L is an artificial system loss factor accounting for loss caused by, for example, hardware. However, these ideal conditions are extremely rare on our planet. Radio waves travel through matter with different properties and are reflected by electromagnetic fields of objects creating a multipath signal. The equation can be regarded useful for sensor network modeling only as a starting point. In the log-distance path-loss equation, the Friis equation can be used to help determining a priori loss values for an experimental system [24]:

$$\overline{PL}(dB) = \overline{PL}(d_0) + 10n \log(\frac{d}{d_0}) \quad (2)$$

In the equation, $\overline{PL}(dB)$ is the predicted signal loss in decibels at distance d, $\overline{PL}(d_0)$ is the path-loss at a reference distance determined by the user, and n is a path-loss exponent determined through experimentation. In a small scale indoor environment, a reference distance of one meter is perhaps the most practical.

Our focus in this paper is on the practical applications for path-loss models. In our research work, we are interested in determining the path-loss at the reference distance as well as the path-loss exponent. These variables have a significant role in the reliability of the model.

It has been suggested in [24] that the Friis equation can be used to determine the path-loss at the reference distance, if the user has no access to better reference values. Values for the path-loss exponent can be found from various sources [24] and the user can use a value that best seems to reflect the surroundings she attempts to simulate.

In our experiments, the main focus was to determine the accuracy of the model. A single link was tested in our research laboratory to see how close the values of the model would come to the measurements. A transmitted signal of 100 mW (0 dBm) was used at the 2.4 GHz frequency. Two sets of ZigBee-enabled hardware were used in our experiments. The first set was comprised of devices with a small ceramic antenna integrated to the chip [22], while the other set used larger detachable antennas [23].

Our research laboratory is a typical office environment with desks, computers and electronics (see Figure 2). The surroundings are sure to cause multipath propagation and several wireless networks were present to cause interference. Over 10 Wireless Local Area Networks (WLANs) were in the vicinity with one WLAN-router close to the test setup.

Tests with a spectrum analyzer revealed interference values with an average of -65 dBm and with spikes up to -40 dBm caused by WLAN transmissions. Interference was therefore significant enough to impact the results. Figure 3 illustrates a spectrum image taken in the research laboratory. The spectrum image shows the range between 2400 MHz and 2485 MHz, spanning the entire ZigBee range.

In the measurements, we used a Rohde & Schwarz FSH6 (model .26) Handheld Spectrum Analyzer [25] (100 kHz – 6 GHz) with an Empfänger receiver, made for measuring signals between 500–3000 MHz band. The center frequency in the measurements was 2.4425 GHz and the channel bandwidth was 85 MHz. Figures of power characteristics were captured

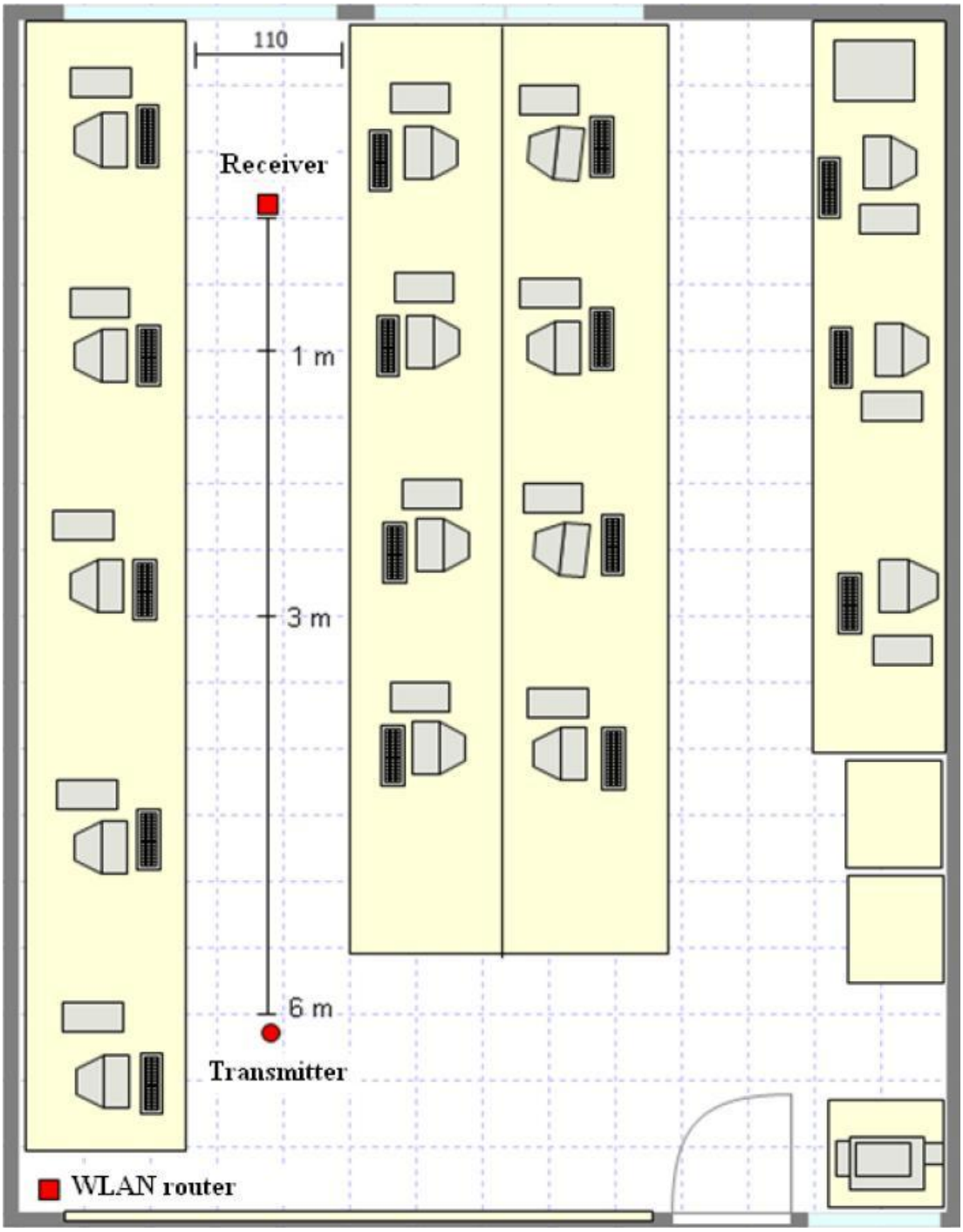


Fig. 2: The layout of our research laboratory used in the experiments, drawn using Room Arranger. [26]

with FSHRemote Program [25]. The same spectrum analyzer was successfully used also in our previous research work on ZigBee and Bluetooth interference measurements [27]. The equipment used is illustrated in Figure 4.

Path-loss was tested at distances of 1, 3 and 6 meters. The received signal strength and average packet loss was measured at these distances with clear LOS.

Tests revealed that on a 1 meter distance, using the Friis equation as an estimate for the reference distance, will not give a realistic value compared to the measurements. On the first hardware setup [22], an estimated received signal strength of $-44\ dBm$ versus an actual received signal of $-77\ dBm$ was observed. On the second hardware [23], the difference between the estimate and the measurements at the reference distance was smaller: an estimated received signal strength of $-44\ dBm$ versus an actual received signal of $-48\ dBm$ was observed.

However, when the measured path-loss at the reference distance was used instead of the Friis equation value, the estimates became accurate to ± 2 dBm at the distance of 3

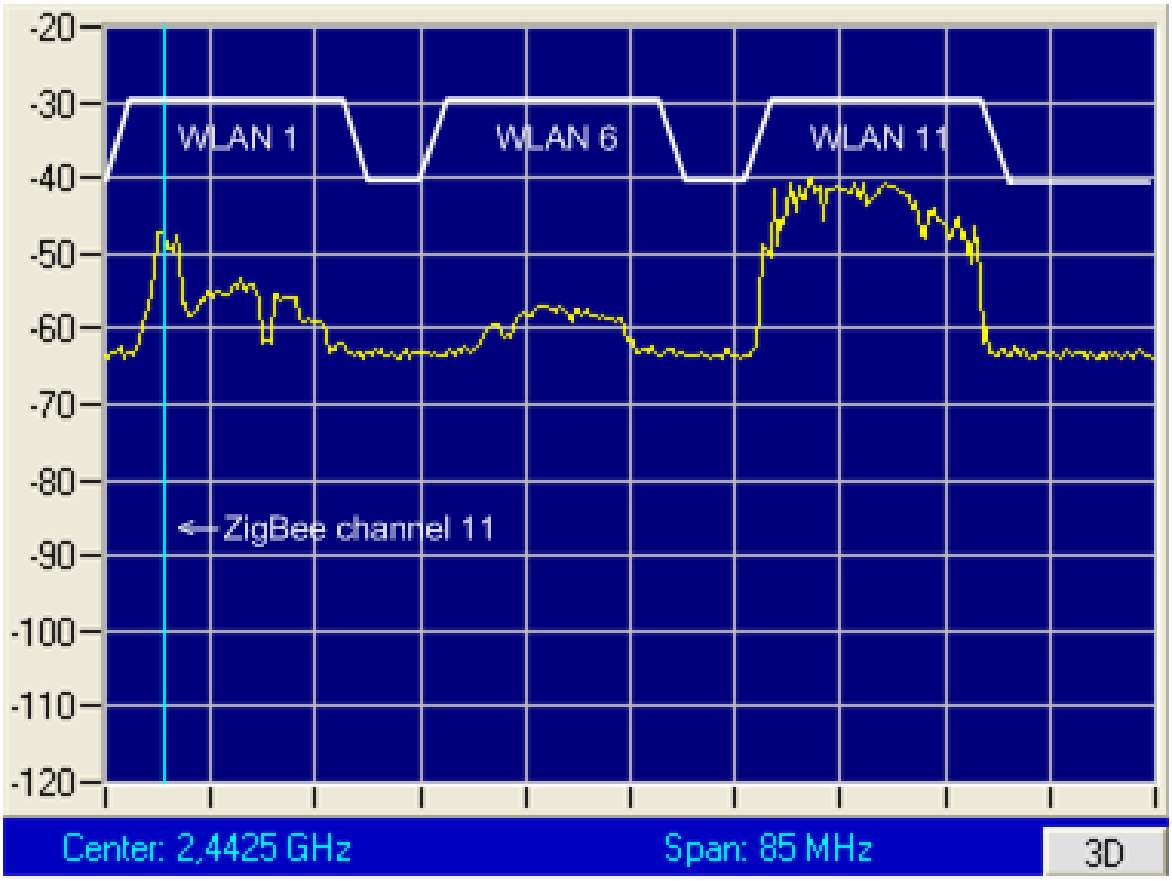


Fig. 3: Spectrum analyzer output from the laboratory: Three non-overlapping channels used by WLAN networks can be seen as pronounced humps. The spike marked with the vertical line is a ZigBee device transmitting on channel 11. The spectrum range of this image is from 2400 MHz to 2485 MHz encompassing the entire ZigBee range.

and 6 meters. An average received signal strength at 3 meters was $-86\ dBm$ on the first setup and $-58\ dBm$ on the second. At 6 meters, a signal strength of $-92\ dBm$ on the first set and $-62\ dBm$ on the second set was observed. This suggests that the use of the path-loss exponent values available (for example in [24]) work adequately. However, the determination of the loss at the reference distance leaves something to be desired. Using the equation for free space loss does not account additional loss caused already on 1 meter distance, or by the hardware used. Therefore, the reference distance must be modeled more accurately even for simulations.

It is worth noting that the signal strength at 6 meters on the first setup was actually below the announced receiver sensitivity limit of the ZigBee-enabled device used, although the manufacturer's specification promises a link distance of 10 – 30 meters for indoor environments. Due to the use of the small integrated antenna, the overall link distance was not nearly as good as when using the devices equipped with larger antennas in the second setup.

Tests conclude that the log-distance path-loss model requires either actual reference distance measurements or a more accurate estimate than what the Friis equation can provide in a short range indoor environment. Using the model to get more realistic link estimates in TOSSIM simulations has been tested with some success in [28]. However, if a path-loss model is based solely on the distance between the nodes with an average modifier for additional loss caused by the environment, the model cannot function in a changing environment. A path-loss model, used in an indoor network simulation with mobile nodes, needs information whether the LOS between the nodes is clear or not.

Other, more sophisticated attenuation models are also available. Our focus in this paper is on small-scale indoor networks. Therefore, models accounting for multipath reception are especially relevant. Two models are prominent in especially this regard: Rayleigh fading and Rician fading. Rayleigh fading assumes that the majority of the signal is composed of Non Line of Sight (NLOS) multipath signals without a dominant LOS signal available. In Rayleigh fading, the signal is assumed to contain a degree of scattering and jitter caused by the multitudes of reflected signals, and it is modeled using Gaussian distribution. Rician fading is otherwise similar, but a dominant LOS signal complemented by multipath signals is assumed. Rayleigh fading is suitable especially in heavily built urban areas, while Rician fading might be more suitable to our focus area. [24]

IV. Fresnel Zone Modeling

Whether two communicating devices have a clear LOS to each other or not is significant in predicting the signal loss

Fig. 4: Rohde & Schwarz FSH6 (model .26) Handheld Spectrum Analyzer [25] with an Empfänger receiver was used in our research laboratory experiments.

between them. For this, the Fresnel zone equation can be useful [24]:

$$F_n = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}} \quad (3)$$

In the equation, F_n is the radius of the n:th Fresnel zone at the point where the distance from the transmitter is d_1 and the distance to the receiver is d_2, while λ is the transmission wavelength. Our interest is in the first Fresnel zone, as it can be used to determine if a clear LOS exists between the two devices.

The first Fresnel zone forms an ellipsoid between the two antennas with the high point being in the middle, where $d_1 = d_2$ (see Figure 5). If the zone is at least 55 % free of obstruction, the LOS can be considered free. If the LOS is obstructed, the path-loss calculations should be adjusted accordingly. We experimented on the validity of the model in our research laboratory by placing a partially blocking metal barrier between the devices at 3 and 6 meter link distances. Initially the results seemed valid with attenuation between the devices increasing as the barrier blocked a greater portion of the Fresnel zone. These experiments were conducted with the Nanorouter devices. However, further experiments using the CC2530 radios with more efficient antennas revealed that multipath propagation is a significant factor in the results. This time, when we used the partition to block the line of sight between the devices, we failed to notice an appreciable change in the received signal strength. This is likely due to multipath propagation occurring in the laboratory. Tests with the Fresnel Zone equation conclude that multipath propagation makes reliable experimenting on the model in an indoor environment difficult, if a signal traveling by the shortest path between the devices cannot be distinguished from a multipath signal. However, it does not mean that using the model as a simulation tool is pointless in an indoor network.

A wireless network simulation using the Fresnel zone in an outdoor environment has been implemented in [29] by using the ns-2 simulator. However, the required calculations were processor intensive and required a long time to complete. This is an issue that needs to be addressed as sensor networks, even in indoor link distances, transmit a great volume of packets with a large number of non-static nodes. However, the system resources of modern computers continue to increase rapidly. Thus, even the most processor intensive calculations become possible over time. Unless a method distinguishing direct signals from indirect multipath signals is available, multipath propagation must be taken into account in the modeling. In practice, it means performing the Fresnel calculations not only to the direct signal, but also to reflected signals.

Using the Fresnel zone in a simulation is not effective unless a 3D simulation environment is available. In a 2D simulation, it is not possible to apply the equation in a realistic manner. Therefore, we need a 3D sensor network simulation environment.

V. Simulation Problems

Our proposal, a novel 3D-based network simulation platform for wireless indoor networks, is intended to address many problems found in the field of WPANs.

First and foremost, the physical properties of radio wave propagation in WPANs are not simulated in a way that a developer would find very useful. Simulators with 2D-based simulation and visualization qualities, such as TOSSIM, are available. However, we feel that a simulation limited to only 2D representation is an oversimplification, albeit a convenient one.

2D spatial representation limits the modeling methods that can be used to create more accurate estimations of the behavior of radio waves. The Fresnel zone cannot be modeled in a 2D space nor the effects of many objects interfering with the transmission. Moreover, the visualization and handling of networks in multi-floor buildings becomes difficult in a 2D model.

Furthermore, wave reflection such as ray-tracing, which could be used for the benefit of physical modeling, cannot be effectively used in 2D spaces either. Two dimensional ray-tracing may have been used for the purposes of radio wave propagation prediction, but using two dimensional ray-tracing, although useful in heavily built urban areas and large scale models, is too simplistic to be useful in small scale indoor environments.

3D-based simulation is available in the form of, for example, Actix's Radioplan RPS software [30], which makes some of the proposed features available. For example, using ray-tracing and designing 3D-spaces is possible. It provides a good

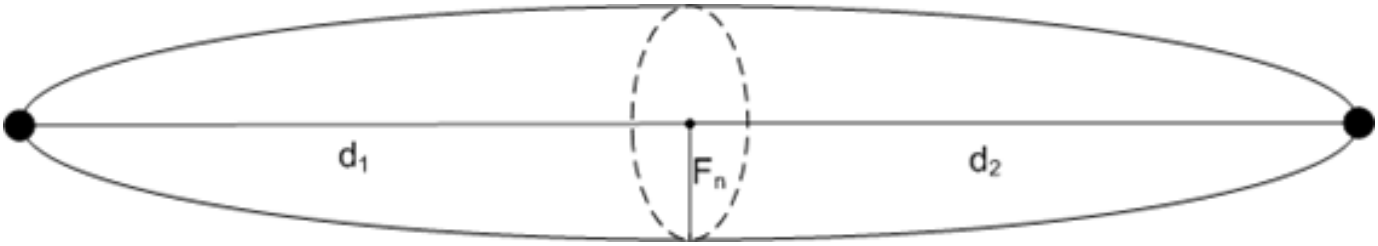


Fig. 5: The Fresnel Zone.

starting point, on which to take example of when designing a more full-bodied simulator. [30]

VI. Novel 3D Simulation Environment

To make 3D sensor network simulations plausible for development and administration, a common platform should be established. Our platform is independent from sensor network technologies themselves: it can be configured for a wide variety of devices on the market. Next we will go through the properties of the proposed system and present an outline of its structure.

The simulation platform is a base component running the simulation of the properties of radio communication and phenomena of the physical space. This platform component is connected to separate components running virtual devices representing wireless network nodes and other applicable devices. The user interface of the platform is also a separate component interfacing with the platform so that maximum flexibility in modifying the system is achieved. The component based model along with the central functions of each component is illustrated in Figure 6.

Section VI-A describes the simulation platform component. The user interface component is explained in Section VI-B. Section VI-C describes the virtual device component. Finally, Section VI-D explains how to use the simulation environment for studying and modeling the propagation of radio waves.

A. Simulation Platform Component

The simulation platform component contains the simulation of the physical world. It contains the three dimensional space, in which all objects and devices are placed. This space contains the physical phenomena and characteristics the real world does: temperature, moisture, lighting, and the passage of time. Gaussian white noise is generated to account for background noise, which may negatively impact on connectivity. Everything that affects on the radio communication or sensor data of devices within the simulation should be implemented as realistically and feasible as possible. Aspects, such as lighting, may not affect on radio communication, but they are useful for simulating sensor stimulae with the device components.

The environment must be able to simulate different physical mediums. A variety of materials will all be implemented with appropriate effects on radio waves, and the simulation of radio waves is the most important aspect of the proposed environment.

A radio wave traveling through an indoor environment is subject to many effects, which should be accounted for in an accurate three dimensional simulation. A signal will attenuate over distance. A wave coming into contact with an object will, depending on the situation, experience reflection, diffraction or scattering.

Reflection is caused by the wave coming into contact with objects, which have much larger dimensions compared to the wavelength of the transmitted wave. For the 2.4 GHz ISM (Industrial, Scientific, and Medical) frequency band, the wavelength is approximately 12.5 centimeters, so an object need not be large to cause reflection. When a reflection occurs, the wave will change its direction much in the same way as a wave of light would do, and it possibly lose a part of its energy depending on the object.

Diffraction occurs when the transmitted wave comes into contact with obstructions with sharp edges, such as corners. In indoor environments these features are understandably common and thus diffraction must be accounted for in the most accurate systems. Diffraction causes the transmitted wave to bend around the edge and form secondary waves.

Wave scattering is caused by objects smaller than the wavelength of the frequency of the wave. Objects like this are also common in indoor environments. A scattered signal multiplies and reflects in a fashion that is very difficult to model accurately, and it is usually represented by a gaussian multipath effect.

Reflection, diffraction and scattering all affect on how far the signal of a radio device will carry, how long it travels and how strong the signal will be in the point of reception. Several different methods of modeling the behavior of radio signals are available with a variety of accuracy, some of which have been presented in this paper.

The behavior of signals can be modeled by using, for example, ray-tracing in a three dimensional space. Ray-tracing is a method, in which the path of the ray is plotted in three dimensional space by utilizing vectors. The ray is traced through the medium and its direction and strength is modified. In three dimensional indoor radio wave propagation, ray-tracing can be utilized by using ray tubes emanating from a transmitter. This tube expands over distance in a way a signal would spread, and the ray tube can be reflected, multiplied and passed through obstacles appropriately. This has been tested with success, for example, in [31]. The receiver is surrounded by a sphere having the contact with a ray tube and receiving the transmitted signal, which strength should be strong enough for decoding. As such, accounting for reflected

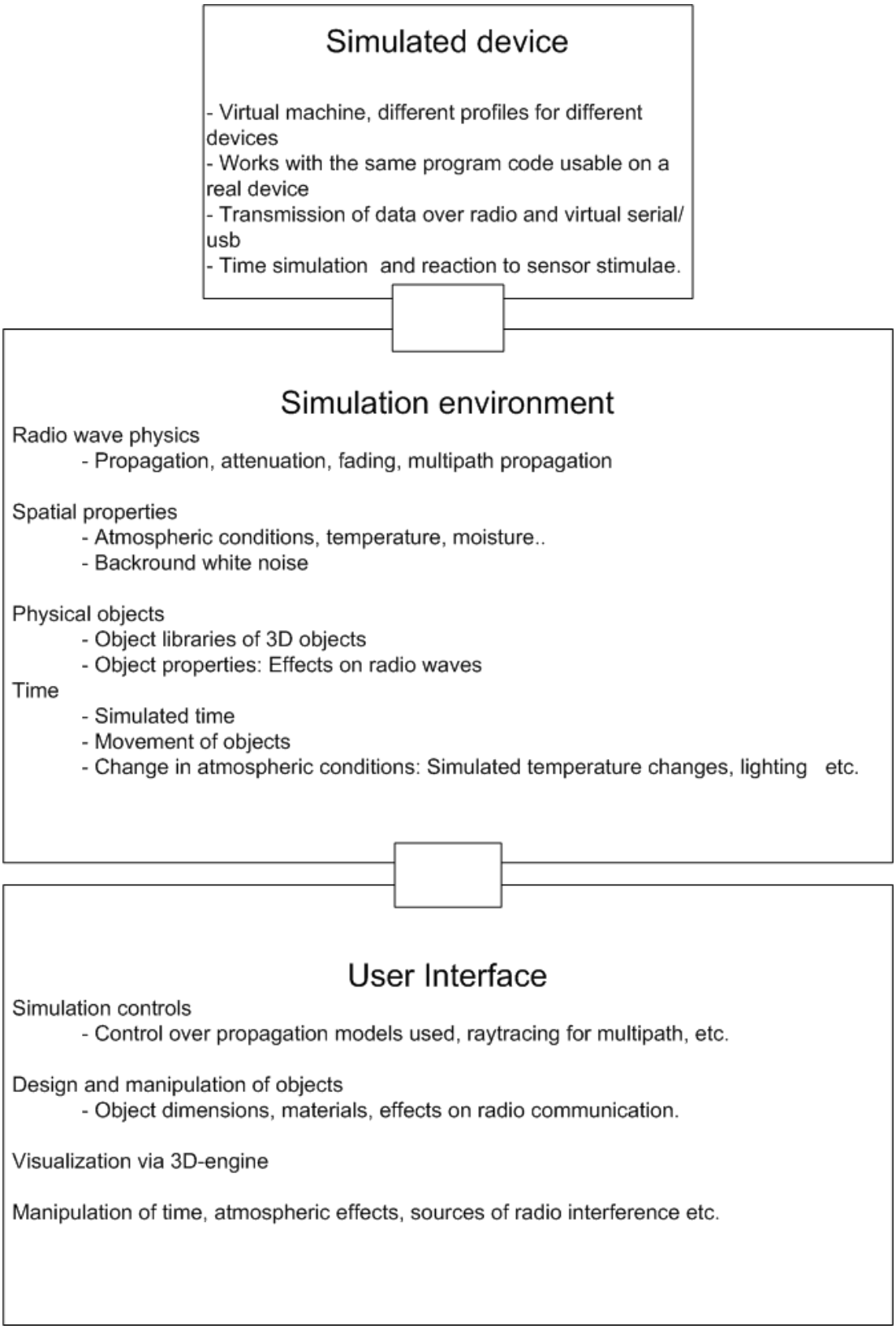


Fig. 6: A component model outline of the proposed platform with a list of its central features.

signals is possible with ray-tracing and has also been tested, for example, in [32]. A ray-tracing model should be optimized so that a minimum of computation is required for sufficiently accurate simulation. For accurate ray-tracing, a large amount of ray tubes need to be sent to every direction from the transmitter, but rays becoming irrelevant to the simulation can be eliminated. For example, the simulation system can keep track on the distance the ray has traveled and apply path-loss calculations on when the transmitted signal strength weakens beyond the receiver sensitivity of the receiving devices, and eliminate such rays. Rays traveling beyond the simulated spacial domain may also be removed. If the computation still proves to be too demanding, rays which have reflected for a certain amount of times may be eliminated. [33]

B. User Interface Component

The user interface component should contain the tools for viewing and manipulating the simulation. The user can design an indoor environment, select wall materials, and place objects, such as furnitures, from a library of premade generic items. The tools make it possible to model an entire building with objects inside. Alternatively, it should be possible to use a simplified attenuation model instead of placing objects by defining properties to a room. For example, a room could be defined to be ”a room with office furniture” and the model could adjust attenuation properties accordingly so that users content with a simpler modeling solution can save time in layout design.

The user can then select the modeling algorithms used for the simulation or optimally, import one manually. For example, the log normal path-loss model presented earlier could be used for path-loss calculation and ray-tracing and Fresnel zone determination to bring additional accuracy to account for multipath signals and partially blocked line of sight. The simple path-loss model could be easily substituted with Rayleigh or Rician fading. Signals traveling through walls and large objects will suffer additional attenuation with their own specific attenuation factors based on the material of the medium. The user can also manipulate the passing of time, as fast forwarding or slowing down time to observe the simulation in action, which can be beneficial. A suitable 3D-engine will be used to present the final model in operation.

C. Virtual Device Component

The device components will be virtual devices interfacing with the platform. The device component will be responsible for the inner workings of a network device. Program code working on an actual device should be made to work on the virtual device with as little modifications as possible. The virtual device will communicate with the platform via its interface by sending and receiving radio signals and by receiving sensor stimulae from the simulation environment. The device could, for example, sample its temperature sensor and receive a temperature value present in the simulation platform in the coordinates of the device. As part of the interface, the device will have to be able to provide for the simulation platform not only the messages it sends but also the frequency used, the signal strength of the transmitted signal, and antenna characteristics, such as antenna gain and receiver sensitivity. Transmission methods, such as DSSS, will have to be considered and functions, such as Clear Channel Assessment (CCA), supported. The creation of the virtual devices for different types of wireless devices is left to third party developers and interested parties wishing to add support for a certain device. Support for virtual serial or Universal Serial Bus (USB) communication for further levels of simulation can be considered.

D. Platform in Operation

Once the simulation environment is set with required structural elements and objects in place, and the radio wave models chosen, the device components can be started up much in the same way as in the real world. The user controls the speed, in which the time progresses and the devices should resume same functions they would do in reality. During the simulation, the user has access to debug tools displaying the traffic of transmitted signals and a graphical representation of the flow of radio messages. A message packet transmitted from a device is stored and the chosen attenuation/ray-tracing model determines the devices that are reached. The packet is then transferred to those devices and the device component will determine independently how to react on it. Messages arriving below a device’s receiver sensitivity level are not transferred. This enables debugging of a network on a level that is not available in most simulators.

Although the creation of a simulation with all the objects and properties of 3D space can be time consuming, the advantages of using such a system can outweigh the additional work required. Figure 7 illustrates a mock-up of what the simulated space would look like on a platform.

An important aspect of the simulator is the possibility to test the code during the development. As in TOSSIM, developers can test their programs before installing them on real devices. This saves time and effort since it is much easier to test a new piece of code virtually than to deploy a real life network to see the results. However, TOSSIM does not support ad-hoc network connections. All connections are predetermined manually at a certain received signal strength and as such the simulation is lacking the ability to determine whether the tested program operates as required or not.

By using the proposed simulation platform, the designing of network deployment becomes easier. It would help those responsible for the deployment of the network, if they can see how densely the devices should be deployed to achieve optimal network coverage. By testing ad-hoc networking and getting a sufficiently realistic estimate on required amount of nodes, the optimal density can be achieved.

Developers can also stress test the network to make sure it works when a lot of activity is present. Great amounts of network traffic in the same frequency can cause transmission problems, which are difficult to predetermine. Foreseeing traffic problems in the design and development phase can save money, as it is done before committing funds to sensor devices because a fairly accurate amount of required nodes can be determined.

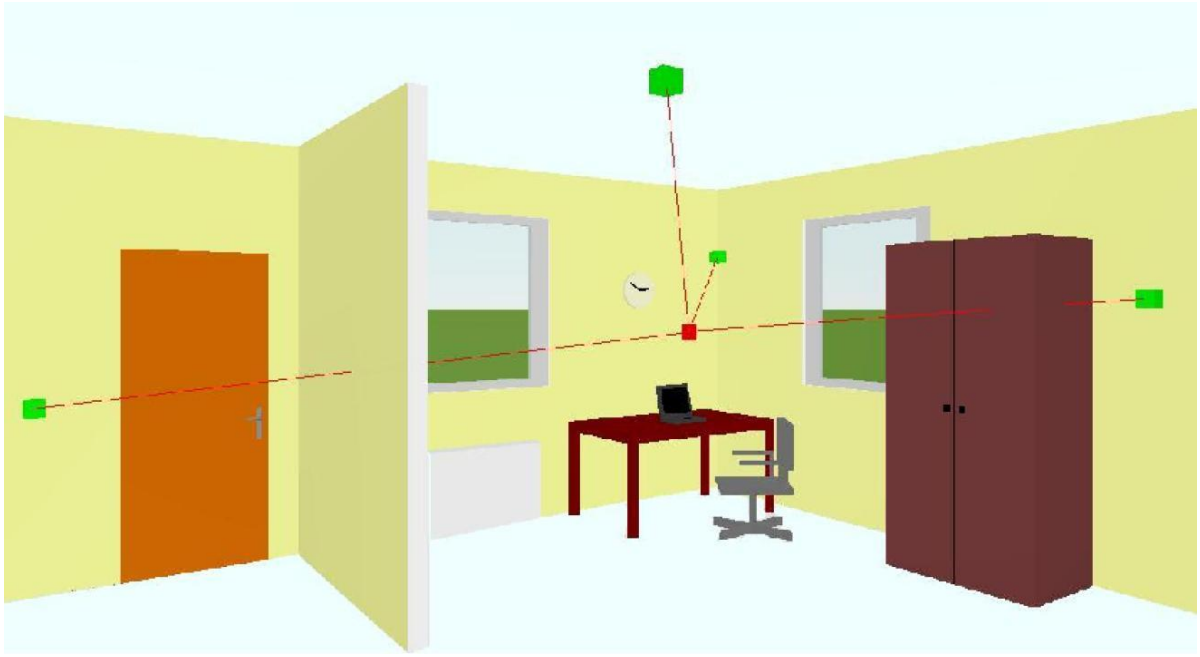


Fig. 7: Mock-up of a 3D simulation space, drawn by using Room Arranger. [26]

The simulation platform could also be used in conjunction with real world data. By replacing device components with the message flow of actual devices from, for example, packet sniffer data and with the radio model removed in the favor of actual on-site information of received signal strengths, one can create a network visualization tool of an actual network for administration purposes. A network consisting of hundreds or even thousands of nodes, perhaps in a large building complex, is difficult to monitor. Administrators can use the simulator tools in conjunction with actual data flow of the deployed network to oversee connectivity and diagnose problems more easily. The visualization would also help in tracking mobile nodes, but would require accurate positioning methods.

Visualizations of radio nodes in two dimensional networks are commonly shown as circles describing the maximum link distance available. This method of visualization is often observed, for example, in wireless locationing, since the locationing algorithms are often based on calculations using circle geometry. This is an oversimplification especially indoors, since the range of the available radio link changes dramatically based on the physical medium the wave is in contact with.

By using the simulation environment to study and model the propagation of radio waves, it might be possible to design better positioning algorithms by using methods such as ray-tracing. Moreover, since a 3D space is used, modeling of the Fresnel zone becomes possible in the determination of LOS between two devices, and this can potentially be used to improve the accuracy of path-loss calculations.

VII. Conclusion and Future Work

A novel 3D-based network simulation platform for wireless indoor networks was proposed based on a project under design and development. Moreover, radio propagation models were investigated by performing path-loss calculations and Fresnel zone geometry estimation in our research laboratory environment.

In our experiments, we tested the feasibility of using the path-loss equation and Fresnel zone calculations in small scale indoor environments. The usage of the Friis free space loss equation proved inaccurate in determining path-loss at the reference distance of one meter due to the hardware used and the environmental factors unaccounted for by the equation as discussed in Section III.

By using measured reference loss values in a laboratory environment, the inaccuracies were eliminated and the simple path-loss equation became a reliable method in predicting path-loss under the test conditions. Tests with the Fresnel zone equation suggest that it can be used to create more precise path-loss predictions in the proposed environment once the effect of multipath signals can be accounted for, as discussed in Section IV.

The simple path-loss equation can be used to simulate the weakening of a radio signal in indoor environment when it is combined with real world test data for the reference loss. The Fresnel zone equation with multipath signals taken into account can be used to analyze whether obstruction between simulated nodes is enough to warrant further loss to the signal. Therefore, both equations should be implemented in the simulation environment.

We stress that the simulation environment proposed in this paper is the result of a pen and paper feasibility study, and not yet a realized software. We are in the initial stages of development of the prototype for the simulator and as such, there are no system validation results nor is the prototype available for scrutiny. However, we strongly believe that the proposed idea is of technical and scientific relevance and interest already at this point of its development.

During our research work, we found out that a design and

simulation environment, such as our novel 3D-based network simulation platform for wireless indoor networks, is sorely needed for improving effective sensor network design, simulation and implementation. The rapidly increasing capabilities of computing create new possibilities to model and simulate physical radio wave properties more accurately using very intricate and sophisticated methods. A generic, standardized platform would be required for the sake of greater interoperability between different technologies within the platform.

To summarize, the potential key benefits of the simulation platform are:

- Easier testing of code in development.
- More efficient network deployment design and visualization.
- Simulated network stress testing.
- Administrative uses in conjunction with real world data.
- Scientific modeling purposes.

A logical next step in our future research work is to create a prototype of a 3D simulation environment by combining existing simulators and available 3D modeling software. A proper implementation of the simulation framework is necessary to evaluate the validity of the proposed system. We will also look into other models to use in simulating path-loss and assess their validity for small scale indoor use. Our further experiments on radio propagation models and positioning algorithms will ultimately show how much benefit the platform can create for design, simulation and implementation of wireless sensor networks.

REFERENCES

[1] M. Asikainen, M. Rönkkö, K. Haataja, and P. Toivanen, "A Novel 3D-Based Network Simulation Platform for ZigBee Networks," in *International Conference on Networking,* pp. 151-156, 2010 Ninth International Conference on Networks, 2010.

[2] G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori, "Performance Measurements of Motes Sensor Networks," in *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Venice, Italy, Oct. 4–6, 2004, pp. 174–181.

[3] I. Essa, "Ubiquitous Sensing for Smart and Aware Environments," *IEEE Personal Communications*, vol. 7, no. 5, pp. 47–49, Oct. 2000.

[4] L. Jiang, D. Liu, and B. Yang, "Smart Home Research," in *Proceedings of the IEEE International Conference on Machine Learning and Cybernetics*, vol. 2, Aug. 26–29, 2004, pp. 659–663.

[5] C. Chee-Yee and S. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges," in *Proceedings of the IEEE*, vol. 91, no. 8, Aug. 2003, pp. 1247–1256.

[6] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: a Survey," *Computer Networks*, Elsevier, vol. 38, no. 4, pp. 393–422, Mar. 15, 2002.

[7] T. Laavola, *Älytalot*, Master's thesis, University of Kuopio, Department of Computer Science, 2008.

[8] T. Laavola, K. Haataja, J. Mielikäinen, and P. Toivanen, "Designing and Implementing an Intelligent X-10-Enabled Home: Studies in Home Intelligence," in *Proceedings of the IEEE Fourth International Conference in Central Asia on Internet, The Next Generation of Mobile, Wireless and Optical Communications Networks*, Tashkent, Uzbekistan, Sep. 23–25, 2008.

[9] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, Georgia, USA, 2002, pp. 88–97.

[10] F. Randy, *Understanding Smart Sensors*, Boston, MA, Artech House, 2000.

[11] Y. Choi, M. Gouda, M. Kim, and A. Arora, "The Mote Connectivity Protocol," in *Proceedings of the 12th IEEE International Conference on Computer Communications and Networks*, Dallas, Texas, USA, Oct. 20–22, 2003, pp. 533–538.

[12] E. Tapia, S. Intille, and K. Larson, "Activity Recognition in the Home Using Simple and Ubiquitous Sensors," *Lecture Notes in Computer Science*, vol. 3001, pp. 158–175, 2004.

[13] M. Asikainen, K. Haataja, R. Honkanen, and P. Toivanen, "Designing and Simulating a Sensor Network of a Virtual Intelligent Home Using TOSSIM Simulator," in *Proceedings of the Fifth IEEE International Conference on Wireless and Mobile Communications*, Cannes, La Bocca, France, August 23–29, 2009, pp. 58–63.

[14] M. Asikainen, *Häiriömallinnus ja kenttätestit IEEE 802.15.4 -verkoissa*, Master's thesis, University of Kuopio, Department of Computer Science, 2009.

[15] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," in *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems*, Los Angeles, California, USA, 2003, pp. 126–137.

[16] L. Perrone and M. Nicol, *A Scalable Simulator for TinyOS Applications*, Institute for Security Technology Studies, 2002.

[17] J. Thomsen and D. Husemann, "Evaluating the Use of Motes and TinyOS for a Mobile Sensor Platform," in *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks*, Innsbruck, Austria, Feb. 14–16, 2006.

[18] UC Berkeley, TinyOS Community Forum. [Online]. Available: http://www.tinyos.net. [Accessed Jan. 12, 2011].

[19] TinyOS Tutorials. [Online]. Available: http://docs.tinyos.net/index.php/TinyOS_Tutorials. [Accessed Jan. 12, 2011].

[20] L. Rayleigh, "On the Resultant of a Large Number of Vibrations of the Same Pitch and of Arbitrary Phase," *Phil. Mag.*, vol. 10, pp. 73–78, Aug. 1880 and vol. 27, pp. 460–469, Jun. 1889.

[21] ZigBee Alliance, ZigBee specifications. [Online]. Available: http://www.zigbee.org. [Accessed Jan. 12, 2011].

[22] Sensinode, NanoStack Manual Version 1.1.0. [Online]. Available: http://www.sensinode.com/EN/news/nanostack-v1.1.0-release.html. [Accessed Jan. 12, 2011].

[23] Texas Instruments, CC2530 Datasheet–A True System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee Applications. [Online]. Available: http://focus.ti.com/lit/ds/symlink/cc2530.pdf. [Accessed Jan. 12, 2011].

[24] T. Rappaport, *Wireless communications: Principles & Practice*, Prentice Hall, USA, 1996.

[25] Rohde & Schwarz, R&S FSH6 (model .26) Handheld Spectrum Analyzer. [Online]. Available: http://www2.rohde-schwarz.com/en/products/test_and_measurement/spectrum_analysis/frequencies. [Accessed Jan. 12, 2011].

[26] J. Adamec, Room Arranger. Shareware version of the software. [Online]. Available: http://www.roomarranger.com. [Accessed Jan. 12, 2011].

[27] J. Suhonen, K. Haataja, N. Päivinen, and P. Toivanen, "The Effect of Interference in the Operation of ZigBee and Bluetooth Robot Cars," in *Proceedings of the Fourth IEEE International Conference in Central Asia on Internet, The Next Generation of Mobile, Wireless and Optical Communications Networks*, Tashkent, Uzbekistan, Sep. 23–25, 2008.

[28] C. Suh, J-E. Joung, and Y-B. Ko, "New RF Models of the TinyOS Simulator for IEEE 802.15.4 Standard," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, Hong Kong, Mar. 11–15, 2007, pp. 2236–2240.

[29] A. Dzambaski, D. Trajanov, S. Filiposka, and A. Granov, "Ad Hoc Networks Simulations with Real 3D Terrains," in *Proceedings of the 15th Telecommunications Forum*, Serbia, Belgrade, Nov. 20–22, 2007.

[30] Radioplan, RPS Radiowave Propagation Simulator User Manual Version 5.4. [Online]. Available: http://www.actix.com. [Accessed Jan. 12, 2011].

[31] C-F. Yang, B-C. Wu, and C-J. Ko, "A Ray-Tracing Method for Modeling Indoor Wave Propagation and Penetration," *IEEE Transactions on Antennas and Propagation*, vol. 46, no. 6, Jun. 1998.

[32] Z. Ji, B-H. Li, H-X. Wang, H-Y. Chen, and T. Sarkar, "Efficient Ray-Tracing Methods for Propagation Prediction for Indoor Wireless Communications," *IEEE Antennas and Propagation Magazine*, vol. 43, no. 2, Apr. 2001.

[33] H. Kim and H. Ling, "Electromagnetic Scattering from an Inhomogeneous Object by Ray-Tracing," *IEEE Transactions on Antennas and Propagation*, vol. 40, no. 5, May 1992.

Multiple Criteria Routing Approaches in Mesh Overlay Networks

Lada-On Lertsuwanakul
Communication Networks
FernUniversiät in Hagen
Hagen, Germany
lada-on.lertsuwanakul@fernuni-hagen.de

Hauke Coltzau
Communication Networks
FernUniversiät in Hagen
Hagen, Germany
hauke.coltzau@fernuni-hagen.de

Herwig Unger
Communication Networks
FernUniversiät in Hagen
Hagen, Germany
herwig.unger@fernuni-hagen.de

Abstract — **The multi-constrained optimal path problem is one of the main issues of Quality-of-Service (QoS) routing, which consists in finding a route between two nodes that meets a series of QoS requirements such as overall delay time, maximum acceptable packet loss ratio, and others. With the aim to improve the QoS routing by considering buffer stages as well as remaining distance to the target, three adaptive routing algorithms in grid-like P2P overlays are presented in this paper: an adaptive probability function, a weighted decision function and a fuzzy-logic approach. In all proposed algorithms, a thermal field is used to communicate the buffer utilization over the network. By means of simulations it is shown that the weighted decision function as well as the fuzzy-logic approach show very good performance according to message losses and overall routing time in both low and high-congestion traffic scenarios. Additionally, all approaches are able to balance the network load and therefore effectively avoid message losses.**

Keywords- multi-constrained decision making, routing algorithm, overlay networks, buffer utilization.

I. INTRODUCTION

The main advantage of structured Peer-to-Peer overlay networks lies in their ability to distribute arbitrary contents over a dynamically changing number of participants and still provide efficient lookup mechanisms. Additionally, such overlays usually provide robust routing architectures, redundant storage and – though more seldom – distributed implementations of trust and authentication mechanisms that avoid single points of attacks and failures.

Unfortunately, in some overlays as e.g. in CAN and Grid-like structures, the routing process can cause single peers to have a high message load, since each may have a central or otherwise crucial position in the network so that a lot of messages are routed to or through it. This problem is enforced, whenever a peer manages content that is accessed by a lot of users in the whole network. The peers around such hot-spots are inherently exposed to higher routing load, since a lot of messages need to be routed to and from the hot-spot. Whereas all messages that are targeted to a hot-spot or its surrounding nodes necessarily have to be routed into the overloaded region, other messages should be routed around it. This not only avoids additional load and possible resulting message losses for the already stressed region, but also decreases and therefore optimizes the delay time for the redirected message. On the other hand, the alternative routes should still have a minimum number of hops to make sure, no messages are lost due to TTL expiries.

To increase the robustness and provide some load-balancing, we therefore propose a routing algorithm for Peer-to-Peer overlays that is able to dynamically route messages around over- or highly loaded peers and regions. To find the fastest, but not necessarily shortest path to the requested target and avoid message losses at the same time, each peer does not only take the target direction into account, but also the buffer levels of its direct neighbors that may be involved into the routing process. To propagate each peer's buffer levels into its neighborhood, a thermal field approach is used.

Such kind of routing problems are generally referenced to as finding a multiple-constraint optimal path (MCOP). The constraints, as e.g. overall routing delay, the maximum number of hops or transfer rate, usually are entailed by application-specific quality-of-service (QoS) requirements.

A multi-criteria decision function is needed to find an appropriate tradeoff between distance and load. Since this function is crucial for the effectiveness of the routing algorithm, we propose and compare three different approaches. Those are: (i) an adaptive probability function, (ii) a weighted decision function as presented in [1], and (iii) a fuzzy-logic approach, which provides a mathematical model for dealing with imprecision and uncertainty as given in common traffic situations in today's communication networks.

The rest of this article is organized as follows: in section II, a short overview about related work is given. Section III discusses both the thermal field approach as well as the three different decision making mechanisms. Section IV shows the simulation of the different decision functions and discusses the results. Section V concludes this article and gives an outlook on future works.

II. Related Work

A. QoS Routing

The multi-constrained optimal path problem (MCOP) is related to the issue of Quality of Service (QoS) routing, which consists in finding a route between two nodes that meets a series of QoS requirements such as overall delay time, maximum acceptable packet loss ratio, and others. Although the utilization of a routing node's message buffers is an indicator for that node's load, current approaches mainly consider available bandwidth or the remaining hop-count for the decision making algorithm [2-4]. Only few approaches, like [5], take buffer utilization into account.

The *Fuzzy Logic Ant based Routing* (FLAR, [23]) is a routing algorithm based on ants, which was enhanced by fuzzy logic. The messages are forwarded according to information gathered from forward and backward ants which dynamically update the routing tables on each node during message transfer. The link delay and link utilization are also considered in the fuzzy logic decision function.

In [24], an Adaptive route selection policy is proposed. The algorithm is based on back-propagation neural networks, which are used to predict the optimum policy for adapting to dynamically changing network load conditions. The back propagation method is used to train the neural network to learn the relationship between different policies and the resulting effects on the network traffic.

B. Routing in Mesh Topologies

Mesh (Grid-like) topologies have been widely used in communication networks as for example in packet/circuit switching between wireless [6, 7] and wired networks [8, 9]. The functions of routing algorithms in general are the provision of the fastest path, prevention of deadlocks, low latency insurance, network utilization balancing, and fault tolerance. Routed by these classical methods, grid-like structures provide multiple paths which have the same hop count. The mesh structure is reliable and offers redundancy which in turn can be used to improve routing performance [10, 11].

In 2000, John Kleinberg [12] introduced a family of small-world network models based on the work of Watts and Strogatz [13]. His models are built of *k*-dimensional grids with a lateral length of *n*, in which each peer has undirected local links connecting it to its neighbors. Additionally, directed far distant links are generated randomly. Kleinberg showed, that optimal routing performance can be gained, when a long distance link between two nodes *u* and *v* is constructed with a probability proportional to $d(u,v)^{-n}$. Hence, for the two-dimensional case, links are added with a probability proportional to the inverse square of the lattice distance of *u* and *v*. In such structures, a path with an expected length of $O(\log^2 n)$ can be found by using a simple greedy algorithm which relies only on local knowledge.

Martel and Nguyen [14] re-analyzed Kleinberg's Small-World model and deduced an expected path length of $\Theta(\log^2 n)$ and a diameter of $\Theta(\log n)$ for the 2-dimensional case. By making use of some additional knowledge of the graph they show that the expected path length can be reduced to $O(\log^{1+1/k} n)$ for a general *k*-dimensional model ($k \geq 1$)

By taking the neighbors of a node's neighbor into account for decision-making, Naor and Wieder [15] improved the delivery time for greedy algorithms. Finally, Zou et al. [16] claimed that Kleinberg's model needs to use global information to form the structure. Consequently, they proposed to use cached long distance links instead of fixed ones. The structure is refined as more queries are handled by the system.

C. Thermal Field Algorithms

A routing approach in analogy to temperature fields in thermal physics was first introduced by Unger and Wulff [17] in 2004 to locate nodes managing contents of common interest in P2P networks. Each node features a temperature, which is an index for the activity of that node. The heat of each node radiates towards its direct neighbors and therefore influences their temperature as well. Whenever the content of a node is accessed or updated, its temperature is increased, whereas during periods of inactivity, the temperature falls of exponentially to align with the temperatures of the surrounding neighbors.

In 2007, Baumann et al. [18] introduced the *HEAT* routing algorithm for large multi-hop wireless mesh networks to increase routing performance. *HEAT* uses anycasts instead of unicasts to make better use of the underlying wireless network, which uses anycasts by design.

HEAT relies on a temperature field to route data packets towards the Internet gateways. Every node is assigned a temperature value, and packets are routed along increasing temperature values until they reach any of the Internet gateways, which are modeled as heat sources. It is a distributed protocol to establish such temperature fields which does not require flooding of control messages. Rather, every node in the network determines its temperature considering only the temperature of its direct neighbors, which renders our protocol particularly scalable to the network size.

III. Multi-Criteria Routing Algorithm

We present three algorithms for making routing decisions in grid-like structures, where each routing node only has local knowledge. Additionally to the Euclidean distance from the current node to a message's destination, the approach also takes the current buffer stages of a routing node's neighbors into account to find optimal paths around congested areas or nodes.

A thermal field is used to communicate the buffer utilization over the network, rendering every node to memorize its neighbors' temperatures. A lower temperature indicates that the respective neighbor currently has more communication resources available and will therefore be capable of handling new data. On the other hand, a message should still be directed towards its destination. Therefore, in the route selection process, the distance between the origin and target node, the length from the current peer to the

target, and the distance from each neighbor to the target node are measured.

We evaluate the performance of our fuzzy-logic based decision function against both an adaptive probability function as well as a weighted decision function. All three approaches base the routing decision on a combination of neighborhood temperatures and target distances. Before we provide a detailed description of each of the approaches, we show how the temperature values of each node are calculated and distributed to build the thermal field.

In the presentation of the algorithms, we denote a grid-like network by a set of lattice points in $n \times m$, $\{(i,j): i \in \{0,1,\ldots n-1\}, j \in \{0,1,\ldots m-1\}\}$. A node's ID is defined by its coordinate (i,j).

A. Thermal Field

In the discussed algorithms, the temperature θ indicates the usage level of a peer's incoming- and outgoing message buffer. The temperature of a node c is referred as θ_c. The possible values of θ_c are in the range from 0 to 1, where 0 denotes an empty buffer and a value of 1 indicates that the buffer is full.

$$\theta_c = \frac{\text{Messages in Buffer}}{\text{Buffer size}}, \quad 0 \leq \theta_c \leq 1 \qquad (1)$$

To reduce complexity, each node only uses one message buffer, which is organized in a FIFO manner. Hence, the temperature of that buffer is equal to the temperature of the node.

Since the routing decision strongly depends on θ_c being up to date, the temperature is recalculated with every message that enters or leaves a buffer. Additionally, the messages themselves act as temperature-carriers, conveying a node's temperature from one peer to another until they either reach their target or expire. This underlines the analogy to convectional processes in thermal physics, where temperature is conveyed by rapidly moving particles.

Each node keeps account on the temperature of its neighbors. Let $N(c)$ be the set of neighbors of c and let k is the number of neighbors when $1 \leq k \leq 4$ in degree of distribution mesh structure is 4. Let i be the index of each neighbor N_i in $N(c)$ where $1 \leq i \leq k$. Additionally, let φ_i be the number of messages sent from N_i to c. Now, there are two cases to update a neighbor's temperature $\theta(N_i)$ in c's dataset:

(i) Whenever c receives a message from neighbor N_i, containing that node's temperature θ_i, the previously stored value, is overwritten:

$$\theta(N_i) = \theta_i, \quad \text{if } \varphi_i > 0 \qquad (2)$$

(ii) If no message is sent from N_i to c, $\theta(N_i)$ is decreased exponentially over time with a configurable time constant of λ:

$$\theta(N_i) = \theta(N_i) \cdot e^{-\lambda t}, \quad \text{if } \varphi_i = 0 \qquad (3)$$

The thermal field was used for all approaches analyzed in this article to enable decision making with only local knowledge.

B. Adaptive Probability Function

The basic concept of using an adaptive probability function is to base the decision on which path to select on a configurable parameter P_θ, which denotes the probability for selecting low-temperature routes in preference over the shortest path. Each node on the route randomly selects a low-buffer route with a probability of P_θ, or a direct route with the probability of $1 - P_\theta$. Higher values P_θ make each node prefer low buffer routes, which may lead to longer routing times. On the other hand, smaller values for P_θ let peers select a direct route more often, and hence may increase the number of message losses due to overloaded nodes along the shortest path. Thus, the challenge is to find values for P_θ which result in both optimal routes and a load balanced network.

It is clear that suitable values for P_θ depend on the current distance that a message still has to bridge to reach its target. If the message is still close to its source, making a detour is acceptable, whereas if only few hops are left to the destination, direct paths should be preferred. Therefore, we propose to use adaptive probability functions (AP_θ) that provide values for P_θ depending on the relative remaining distance, which we denote as Ω. When a message is sent from a source node σ to a destination node ϕ, the distance between the two nodes is $d(\sigma,\phi)$. The distance from any node c along the path to the target is $d(c,\phi)$.The relative remaining distance Ω is now determined as follows:

$$\Omega = \frac{\text{Distance\_current\_to\_target}}{\text{Distance\_source\_to\_target}} = \frac{d(c,\phi)}{d(\sigma,\phi)} \qquad (4)$$

In previous works [18] it came out that two adaptive probability functions of Ω showed good results for different scenarios, which we denoted as AP_θ^4 and AP_θ^5.

Adaptive Probability4 (AP_θ^4): AP_θ^4 is an exponential cumulative distribution function (cdf). The probability of using a low-temperature route has a co-domain of [0, 1). It results in strongly preferring low-temperature paths at the beginning of the routing process. The closer the message comes to the target, the more the direct route is preferred. When the target is only a few hops away, the thermal field is almost completely ignored.

$$AP_\theta^4(\Omega;\lambda) = 1 - e^{-\lambda\Omega} \qquad (5)$$

Adaptive Probability5 $\left(AP_\theta^5\right)$: Whenever the low-temperature path is preferred over the shortest path, the message could go astray, which results in higher probability of message losses. Therefore, AP_θ^5 is designed to pull back the message onto the shortest path, whenever the current distance to target becomes larger than the overall distance between source and target. In such cases, the probability of using the path with the lowest temperature decreases.

$$AP_\theta^5(\Omega;\lambda) = \begin{cases} e^{-\lambda\left(1+\frac{1}{\Omega(t)}\right)}, & d(c2\phi) \le d(\sigma 2\phi) \\ 1 - e^{\left(\frac{-\lambda}{\Omega(t)}\right)}, & d(c2\phi) > d(\sigma 2\phi) \end{cases} \quad (6)$$

Fig. 2 depicts both AP_θ^4 and AP_θ^5 as functions of the relative remaining distance Ω.

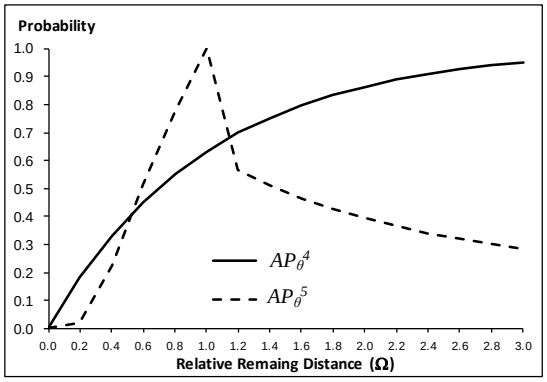


Figure 2. Adaptive Probability Functions AP_θ^4 and AP_θ^5

C. Weighted Decision Funtion

In this approach, a weight is assigned to every neighbor that is based on the neighbor's distance to target and temperature. The message is then routed to the neighbor with the lowest weight. Each weight is calculated as a linear combination, where the coefficients control the influence of each summand on the total weight. Although this approach seems similar to the adaptive probability function presented in the previous section, it is different in one important point: it always selects the path with the minimum weight, whereas the AP approach in some situations might select paths with higher temperature to explore them.

Again, let all N_i with $1 \le i \le 4$ be the neighbors of current node c and let ϕ be the target node of the message that is to be routed. Additionally, let $d(N_i,\phi)$ be the Euclidean distance from a neighbor to the target node and finally, let $\theta(N_i)$ be the temperature of neighbor N_i. The weight of the edge to the neighbor N_i is now calculated as follows:

$$f(N_i,\phi) = \alpha \cdot d(N_i,\phi) + (1-\alpha)\cdot\theta(N_i) \quad \text{with } 0 \le \alpha \le 1 \quad (8)$$

The coefficient α defines the influence of the remaining distance to target and the load of the next hop on the total weight and therefore on the routing decision. Higher values for α let the node select a more direct path while taking the risk to lose the message due to buffer overflows. On the other hand, lower values for α result in selecting a low buffer route, which on the other hand may leading to long routing times.

The pseudo-code of the weighted decision function approach is shown in Fig. 3. This code is executed on every node at each simultion timestep.

```
1   ..
2   while (receiveMsg != null)  {
3     updateNeighborTemperature();
4     if (currentIsTarget())
5       continue;
6     else if (queueBuffer !=MAX)
7       keepInQueue();
8     else
9     lostMessage();
10  }
11  while (queueBuffer != null && outBW != MAX){
12    popMessageFromQueue();
13    for (1 to sizeOfNeighbor)  {
14      d = α * (distanceToTarget);
15      t = (1-α) * (Temperature);
16      WeightNeighbor = d + t;
17    }
18    nextNode = minWeightNeighbor();
19    forwardMsg( nextNode );
20  }
21  spreadTemperature();
22  ..
```

Figure 3. Pseudo-code of weighted decision function approach

The algorithm consists of three parts. In lines 2 to 10, the node receives messages, brings its neighborhood temperature database up to date and decides, if the message needs to be routed or already received its target. If the message is to be routed further, the buffer is checked for remaining free space to handle the message. If no free space is available, the message is dropped.

Lines 11 to 20 describe the forwarding process that is started, when the buffer contains any messages. In line 12, the message is taken from the FIFO buffer, in line 13-17, the weights for each neighbor are calculated. Then, the minimum weight is selected and the message is forwarded to the according neighbor (lines 18-19). The minimum value is found on line 18. Afterwards, line 19 is used to forward the message. Finally, the node recalculates its own buffer's temperature.

D. Fuzzy Logic Approach

Fuzzy Logic was first introduced by Zadeh [21] in 1965. It allows a computer to take decisions the same way as humans do it: not always precise. People think and reason using linguistic terms such as "hot" and "fast", rather than using precise numerical terms as "90 degrees" or "200 km/hours", respectively. The fuzzy set theory models the interpretation of imprecise and incomplete sensory information as perceived by the human brain. Thus, it represents and numerically manipulates such linguistic information in a natural way via membership functions and fuzzy rules. Some advantages of fuzzy logic are that it is

conceptually easy to understand, flexible, and tolerant towards imprecise data. It can model nonlinear functions of high complexity, and it also can be built on top of expert's experience.

A key feature of Fuzzy Logic is to handle uncertainties and non-linearity as they exist in physical systems, similar to reasoning conducted by human beings, which makes it very attractive for decision making systems. A fuzzy logic system comprises basically three elements: (i) Fuzzification, (ii) Knowledge base (rule and function), and (iii) Defuzzification. Fig. 4 shows the generalized block diagram of a fuzzy system.

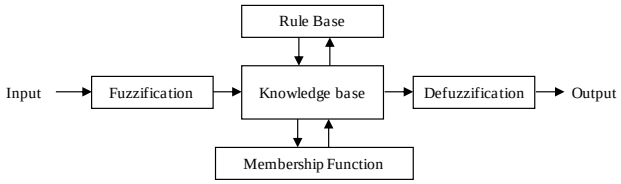


Figure 4. Block diagram of a generalized fuzzy system

The function of fuzzification is to determine an input value's degree of membership to the best corresponding value of a fuzzy set. The fuzzy rule base is used to present the fuzzy relationship between input- and output fuzzy variables. The output of the fuzzy rule base is determined based on the degree of membership specified by the fuzzifier. The defuzzification is used to convert outputs of the rules into so-called "crisp" values, like real numbers.

In terms of temperature field routing, the inputs to the fuzzy controller are: (i) buffer usage status, (ii) current distance to target, and (iii) neighbor type. These three selection parameters make the route reflect the network status, the nodes' ability to reliably deliver packets as well as the direction to the target. The buffer usage is calculated the same way as in (1), the current distance to target is the reciprocal value of Ω as calculated (4). The neighbor type is defined by the difference of the distances from current node to target and from the respective neighbor to target, $d(N_i,\phi)$-$d(c,\phi)$.

Those three input variables are now fuzzified. The neighbor's temperature is now described as either "*Cold*", "*Tepid*", "*Warm*", *Hot*" or "*Torrid*", the neighbor type can either be "*Closer*" or "*Farer*".

Finally, the distance can either be "*VeryFar*", "*Far*", "*StartPoint*", "*Close*" or "*VeryClose*". Fig. 5 shows the respective membership functions to classify the input variables. Five terms are defined to describe the output of the evaluation of each neighbor: Using a neighbor as the next hop can either be rated "*VeryBad*", "*Bad*", "*Fair*", "*Good*", or "*VeryGood*" as shown in Fig. 6.

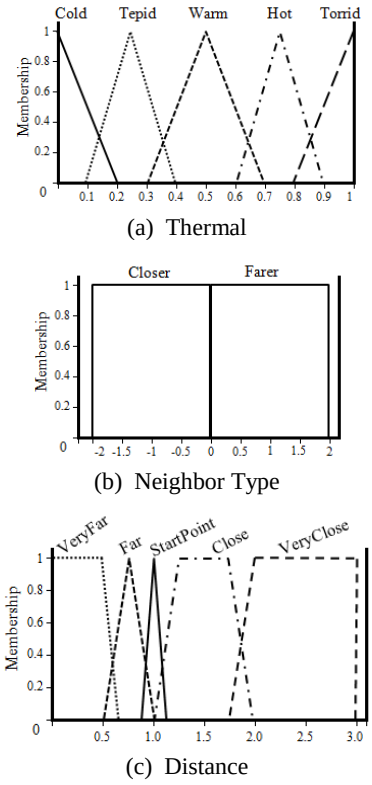


Figure 5. Fuzzy Membership function of input variable

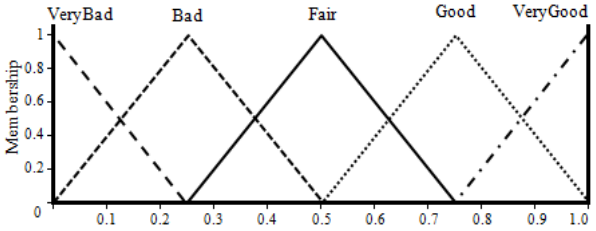


Figure 6. Fuzzy Membership function of Neighbor Rating

A rule set containing 50 atomic rules as shown in table 1 is now used to evaluate the suitability of each neighbor for routing the message next. The rules assign a single rating value to every possible combination of input values. By merging of rules, the number of rules can be reduced to a total of 37.

Table1. Fuzzy Rule Base

Neighbor Rate		Neighbor = Closer					Neighbor = Farer				
		Thermal					*Thermal*				
		Cold	*Tepid*	*Warm*	*Hot*	*Torrid*	*Cold*	*Tepid*	*Warm*	*Hot*	*Torrid*
Distance	VeryClose	VeryGood	VeryGood	Good	Fair	Bad	Good	Good	Fair	Bad	VeryBad
	Close	VeryGood	VeryGood	Good	Fair	Bad	Good	Good	Fair	Bad	VeryBad
	StartPoint	VeryGood	VeryGood	Good	Fair	Bad	Good	Good	Fair	Bad	VeryBad
	Far	VeryGood	Good	Fair	Bad	Bad	Fair	Fair	Bad	VeryBad	VeryBad
	VeryFar	VeryGood	Good	Fair	Bad	Bad	Fair	Bad	VeryBad	VeryBad	VeryBad

Some example rules are:

R1: IF thermal IS Cold AND neighbor IS Closer THEN neighbor_rate IS VeryGood;
R2: IF thermal IS Torrid AND neighbor IS Farer THEN neighbor_rate IS VeryBad;
…
R37: IF thermal IS Hot AND distance IS VeryFar AND neighbor IS Farer THEN neighbor_rate IS VeryBad;

Rules *R1* and *R2* are combined rules that only depend on two input values, because the third value (*distance* in this case) has no influence on the result.

The rating can now give guidance for which neighbor to use as the next hop. If more precision is needed, because several neighbors have the same rating, Defuzzification, i.e. the process of conversion of a fuzzy output set into a single number, can provide more clarity. In our simulations, Mamdani's "Center of Gravity" (COG) method has been used:

$$Neighbor\ Rate = \frac{\sum_{i=1}^{N} x_i \cdot \mu(x_i)}{\sum_{i=1}^{N} \mu(x_i)} \quad (9)$$

So, the center of gravity is calculated by multiplying each input value (x_i) with the output of its corresponding membership function ($\mu(x_i)$), sum up all of those products and divide it by the sum of the membership function's outputs. The COG method is the most widely used defuzzification strategy, which is reminiscent of the calculation of the expected value of probability distributions.

IV. SIMULATIONS

In this section, the simulation results for the three different decision making approaches based on thermal fields for buffer load propagation are discussed.

A. Simulation Setup

1) *Simulation Tools* – The simulation was analyzed using P2PNetSim, a simulation environment for large distributed P2P networks [22]. This flexible tool can be used to simulate, model, and analyze any kind of networks. It has been used for example to analyze distributed RFID-processing as well as the spreading of infectious diseases. Due to the distributed nature of the simulation engine, it is able to handle simulations with millions of individuals. Peers are configured collectively but still individually using an open XML configuration format. for simulation setup. The peer-behavior can be implemented in the Java programming language.

2) *Network* – In the simulations, the networks are organized into two-dimensional grid structures, each composed of 10,000 nodes (100x100). Nodes are connected to their neighbors in all four directions. The coordinate of a node is serves as its ID. The grids overlay a simulated IPv4 network. The buffer sizes and outgoing bandwidths are limited for all the peers, both distributions following a power-law distribution. There are two types of messages: data packet and acknowledgements. The system handles data packet in First-In-First-Out (FIFO) manner, while the acknowledgements are handled with priority.

3) *Traffic pattern* – Traffic is generated randomly by all network nodes. The sending probabilities and intensities are distributed exponentially for both a source node generates, as well as the number of messages that can be sent per simulation time-step. The constants λ_{send} and λ_{number} therefore indicate the load (congestion) of the simulated network. All simulations run until 500,000 messages have been processed,

4) *Performance measurement* – The metrics used to measure the performance using different decision methods are *loss and success ratios*, *average hop-count*, *average delay time (time-steps)*, and *average routing time*. The total routing time includes both the routing steps and waiting times (delay) on busy nodes. Furthermore, load balancing performance was assessed by the number of heated nodes with a buffer usage ration of more than 0.7. For the three decision mechanisms described in section 3, Adaptive Probability Function (AP), Weighted Decision Function (WF), and Fuzzy Logic (FL), the performance is measured. For the weighted decision function, three configurations for *α:1-α* have been used: 0.1:0.9, 0.5:0.5, and 0.1:0.9. So, in the 0.1:0.9-configuration, the decision function considers the distance to target with a weight of 0.1 and the temperature with 0.9, whereas in the 0.9:0.1-configuration, the distance to target is weighted with 0.9 and the temperature with 0.1.

All three approaches are compared to a pure shortest path approach (SP), which does not take the current buffer level of the next hop on the route into account.

B. Simulation Results

The first scenario compares the performance of the decision mechanisms in a low congestion networks. The time constant that defines the probability to generate messages on a specific node is $\lambda_{\text{send}} = -0.1$ and each source node generates only one message per time at maximum. The average number of messages generated per simulation time step is approximately 800 messages.

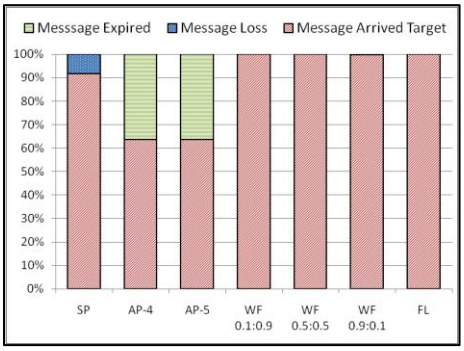


Figure 7. Success delivery ratios in low congestion networks.

Fig. 7 shows the success delivery ratio comparing all decision algorithms in such low congestion networks. The weighted functions as well as the fuzzy logic approach show the best results with 100% of successfully delivered messages, so there was no message expired or lost due to overloaded nodes. On the other hand, the adaptive probability functions show a message expiry ratio of 36%. Those losses occur, when the decision functions strongly prefer low-temperature routes over the shortest path. This way, messages can take remarkably longer routes, inevitably leading to a higher message expiry ratio.

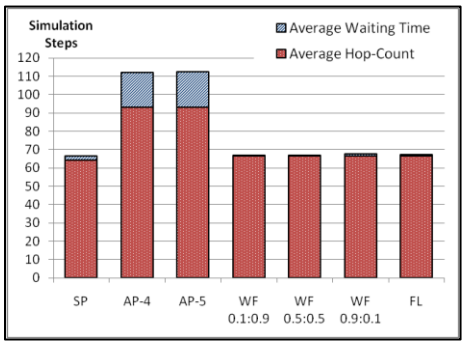


Figure 8. Average routing time is the summary of hop-count and waiting time in low congestion networks.

The average routing time for low congested networks is shown in Fig. 8. Shortest path, weighted decision function and fuzzy logic approach all have similar performance results. In terms of hop-counts and delivery time, the shortest path approach generates the best results with an average of 60.87 hops per message. During 3.83 time-steps, the average message is delayed in congested message buffers, so that the SP approach sums up to a total routing time of 64.70 time-steps. The performance of the weighted decision functions in the configurations 0.1:0.9, 0.5:0.5, and 0.9:0.1, as well as the performance of the fuzzy logic approach are in the same range with only slightly increased values. The hop-counts are 65.12, 63.28, 63.51, and 64.98, whereas the delay times are 3.58, 3.45, 3.81, and 4.08 respectively. So, the average routing times are 68.71, 66.73, 67.32, and 69.06 in order. But the adaptive probability function results show remarkably higher number of hops and delay time.

In Fig. 9, the load balance of networks is presented. The graphs represent number of nodes that have temperature or buffer utilization level higher than 0.7 or 70% of the buffer space. The shortest path method obviously shows many high temperature nodes comparing to others decision algorithms.

The results of the first simulation scenario are that in low congestion networks, the shortest path approach delivers messages in the fastest possible manner and therefore shows the best performance. On the other hand, SP generates the highest amount of heated nodes, even though total load of the network is yet low. The weighted decision function and the fuzzy-logic approach accept short detours, resulting in slightly higher routing times, but utilize the network resources much better and therefore generate remarkably fewer heated nodes. The adaptive probability function approaches show considerably longer routing times.

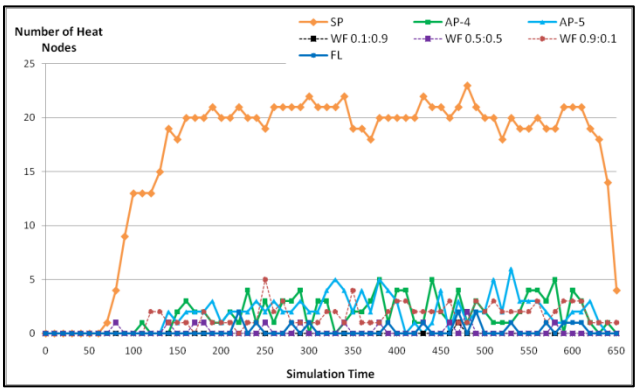


Figure 9. Number of nodes had higher buffer usage level than 70% of buffer size in low congestion networks.

In the second scenario, medium congestion networks are analyzed. The time constants for the probability distribution functions are $\lambda_{send} = 0.1$ for the number of messages per peer and $\lambda_{number} = 0.5$ for the number messages per time-step. The average number of messages that are launched per simulation time is approximately 1,000 messages.

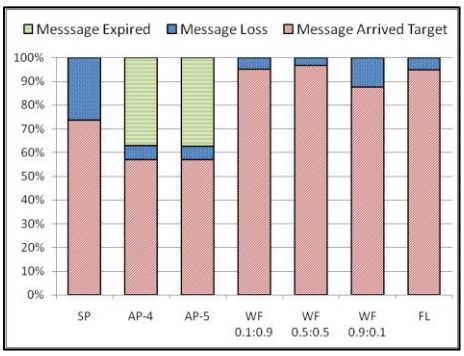


Figure 10. Success delivery ratios in medium congestion networks.

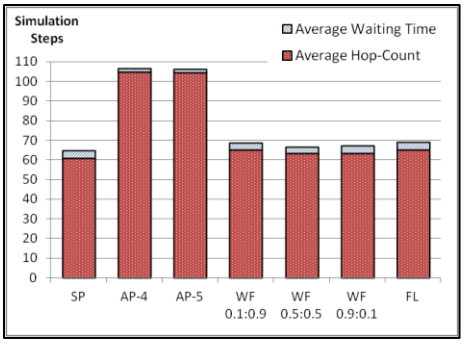


Figure 11. Average routing times in medium congestion networks.

Similar results can be seen when analyzing the routing times as shown in Fig. 11. Again, the 0.5:0.5 configuration

for the weighted decision function shows best performance with 66.73 time-steps of average total routing time, which consists of an average number of 63.28 hops and 3.45 time-steps of delay. Again, the 0.1:0.9 configuration as well as the fuzzy-logic approach show similar performance. In contrast to the message loss ratio in Fig. 10, the 0.9:0.1 also shows good results, when it does not loose messages due to overfull buffers. The adaptive probability function again needs a lot more hops to route the message.

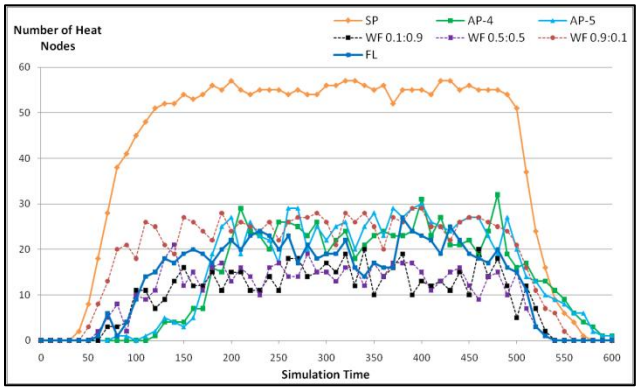


Figure 12. Number of nodes had higher buffer usage level than 70% of buffer size in medium congestion networks.

As expected, the shortest path approach tends to produce overfull buffers a lot more than all multi-constraint approaches, as can be seen in Fig. 12.

In the final scenario, the traffic in a highly loaded network is analyzed. The time constants for the distribution functions now are $\lambda_{send} = 0.1$ and $\lambda_{number} = 0.1$. The average number of messages per time-step is approximately 3,200.

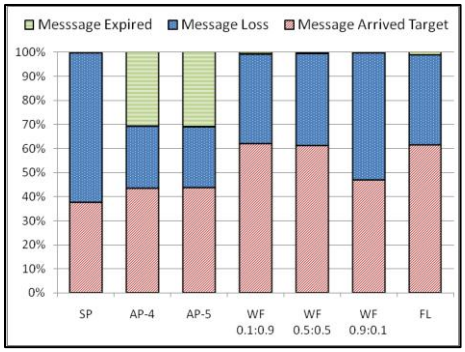


Figure 13. Success delivery ratios in overloaded traffic networks.

In this scenario, the shortest path approach can only deliver a little more than one third of the messages, because now a lot of highly loaded nodes exist and the shortest path approach puts even more load on these nodes. Both adaptive probability functions now perform a little better than SP with 43% of successfully delivered messages (Fig. 13). It is remarkable that the AP approach does loose the most of the messages due to TTL expiries and not because of overloaded buffers. Again this is a direct result of this approaches tendency to make detours into network regions that are far away from the shortest path. So, in highly loaded networks, increasing the TTL could make the AP approach very successful. In terms of successful delivery, the 0.1:0.9 and 0.5:0.5 configurations of the weighted decision function and the fuzzy-logic approach show the best performance with approximately 60%.

If the shortest path algorithm is able to route a message to target, it does so in the fastest possible manner. The weighted decision functions and the fuzzy-logic approach take slightly longer routes, which directly results from avoiding highly loaded nodes. The adaptive probability functions need a lot more hops but on the other hand show the best values for message delays in congested buffers (Fig. 14).

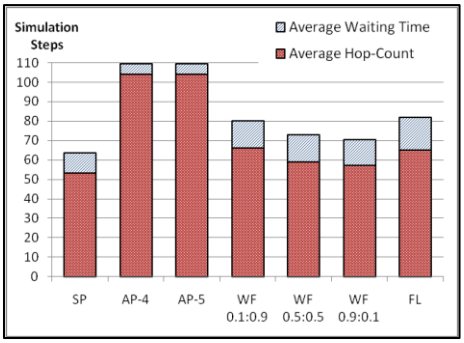


Figure 14. Average routing time is the summary of hop-count and waiting time in overloaded traffic networks.

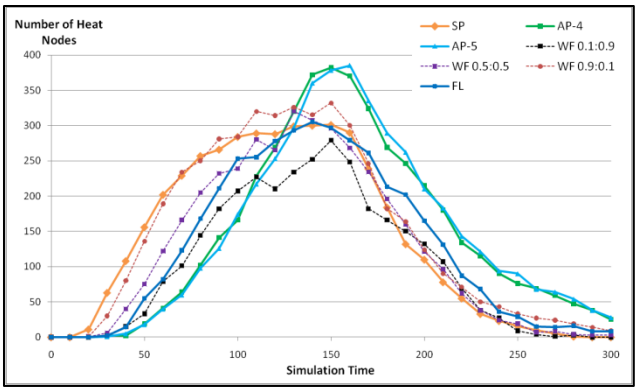
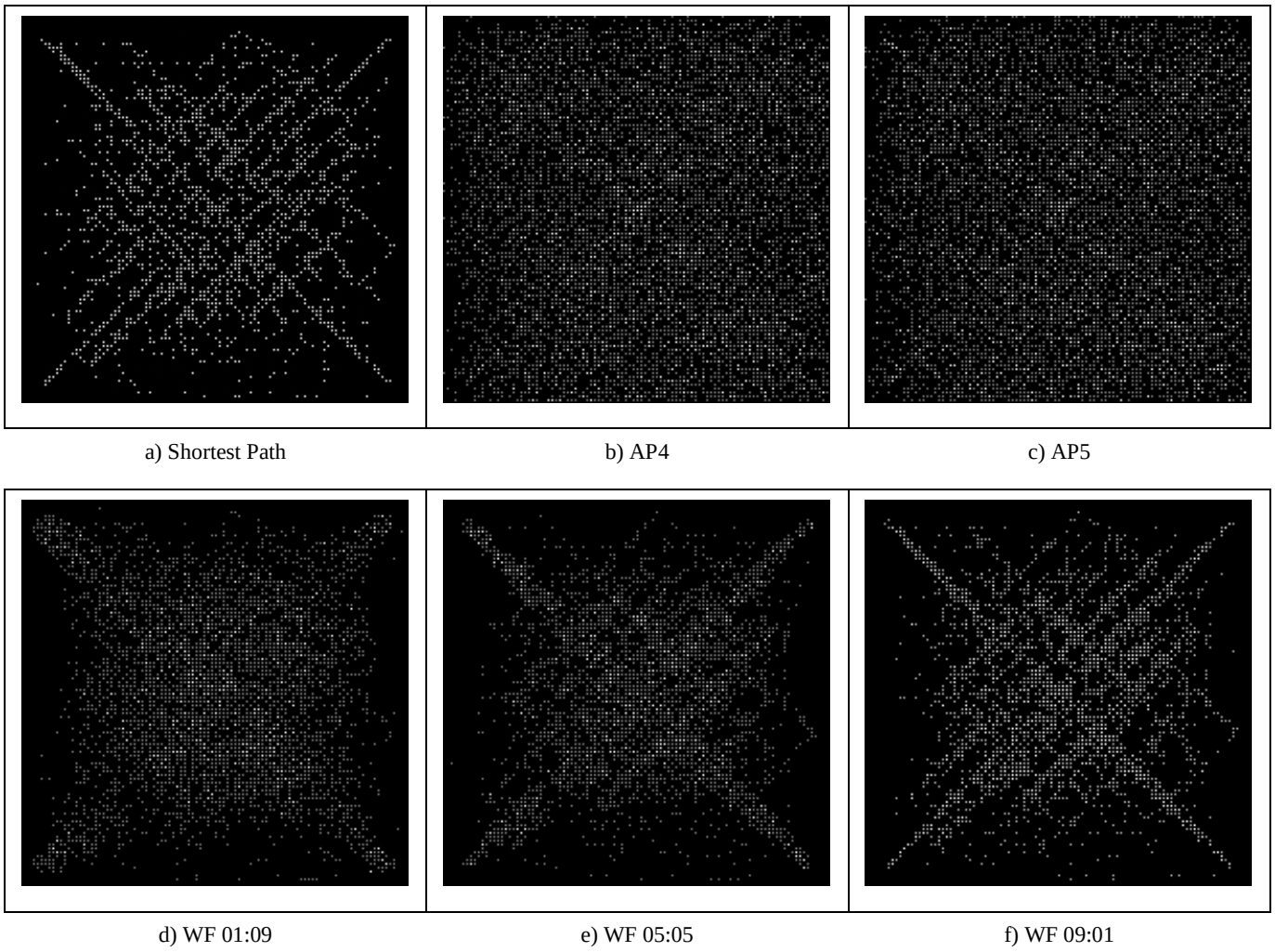


Figure 15. Number of nodes with higher buffer usage level than 70% of buffer size in overloaded traffic networks.

In highly congested networks, the fuzzy-logic approach distributes the load best over time and therefore does produce the lowest amount of heated nodes per time-step (Fig. 15). It is remarkable that the results of shortest path approach are similar to those of the weighted decision functions and even better than the adaptive probability functions. This is, because a lot of messages are dropped long before they are delivered or expire and therefore do no longer add on the network load.



a) Shortest Path b) AP4 c) AP5

d) WF 01:09 e) WF 05:05 f) WF 09:01

Figure16. Heat nodes distribution in highly congested network

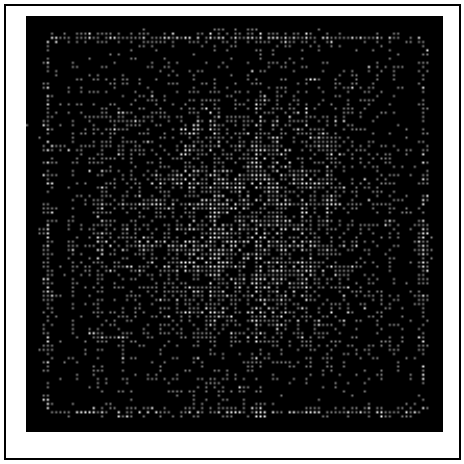
Another part of the simulation results is the distribution of highly loaded nodes over the whole network, which is shown in Fig. 16. The snapshots refer to the high-congestion scenario. Each subfigure represents the 100x100 grid at the simulation time of highest load. Each pixel represents one node. The lighter the node is, the more load it has, i. e. the higher its buffer utilization and therefore its temperature is. To make the source and target nodes clearly visible, in this simulation only four nodes placed at the corners of the grid generate messages to the nodes on the opposite side of the network.

It can be seen that the shortest path algorithm (Fig. 16a) generates a lot of overloaded nodes along the direct path from source to target, but only few load aside from that paths. On the contrary, the adaptive probability functions (Fig. 16 b,c) distribute the load over the whole network. One can see that this approach does not generate fewer loads than SP. This is, because the messages stay a lot longer in the network, until they are dropped due to TTL.

In Fig. 16 d to f, the three configurations for the weighted decision function are shown. In general, the results for this approach show that it uses a balanced path between shortest distance and lowest temperature. So, the 0.1:0.9 configuration allows for some detours from the direct path and therefore generates only few highly congested nodes along the direct path and more mildly congested nodes left and right from that path. In the 0.5:0.5 configuration, the shortest paths are taken stronger into account, so that more highly loaded nodes can be seen on that path. Finally, the 0.9:0.1 configuration looks similar to the shortest path approach although it still balances the load better than SP.

Finally, the fuzzy logic algorithm distributes the message paths over almost all possible routes and so takes most of the load from the highly stressed center of the network in Fig. 16g. In terms of network balancing, this approach therefore generates the best results.

Summarizing the simulation results, one can say that both the weighted decision function and the fuzzy-logic approach are able to handle high-traffic situations remarkably better and with an almost twice the success ratio as the shortest path approach does. While the weighted decision approaches show slightly better performance in terms of overall routing



g) Fuzzy Logic

Figiure16. Heat nodes distribution in highly congested network

time, the fuzzy-logic approach balances the traffic much better over the whole network and therefore avoids creating overloaded regions. Although the adaptive probability functions also distribute the load over the whole network, they show poor performance in low-congestion situations.

Additionally, fuzzy-logic has the advantage that it can take more constraints into account, as e.g. bandwidth, size, load prediction, etc., which makes this approach more flexible than all other approaches analyzed in this article.

V. CONCLUSION AND FUTURE WORK

We have presented and analyzed three general approaches for multi-criteria optimum path decision making in distributed systems. All approaches base their decision on both the distance to target as well as the current load of the possible next hop nodes. The load was distributed using a thermal field approach. Through simulations, we have shown that the weighted decision function and the fuzzy-logic approach show good performance in different network traffic scenarios. The flexible fuzzy-logic approach also is able to balance the network load over the whole network in a very good manner.

Because both the weighted decision function and the fuzzy logic approach showed good performance in high-congestion networks, both concepts shall be merged as part of our future works. The result shall be a dynamically generated weighted decision function that can take more than just two parameters into account and so be truly multi-criteria. Additionally, the thermal filed approach can also be used to build a traffic dependent overlay network structures that can enable and disable links depending on the load of nodes or regions.

Finally the outlook for the project is to implement and deploy the proposed algorithms using real tested data, as well as compare to existing similar approaches.

REFERENCES

[1] L. Lertsuwanakul, S. Tuammee, and H. Unger, "Routing with Temperature Field in Mesh Overlay Network", In Proc. of 9th International Conference on Networks (ICN 2010), France, 2010, pp. 285-290.

[2] B. Peng, A. H. Kemp, and S. Boussakta, "QoS Routing with Bandwidth and Hop-Count Consideration: A Performance Perspective, Journal of Communications, Vol. 1, No. 2, May 2006.

[3] M. Song and S. Sahni, "Approximation Algorithms for Multiconstrained Quality-of-Service Routing," IEEE Transactions on Computers, vol. 55, no. 5, May 2006, pp. 603-617.

[4] X. Lin and N. B. Shroff, "An Optimization Based Approach for QoS Routing in High-Bandwidth Networks," In Proc. of IEEE INFOCOM, 2004.

[5] R. Zhang and X. Zhu, "Fuzzy Routing in QoS Networks", FSKD 2005, LNAI 3614, pp. 880-890.

[6] A. Lee and P. A. S. Ward, "A Study of Routing Algorithms in Wireless Mesh Networks", ATNAC, December 2004.

[7] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," In Proc. Of 10th annual international conference on Mobile computing and networking (MobiCom'04), USA, 2004.

[8] A. Vishwanath and W. Liang, "On-Line Routing in WDM-TDM Switched Optical Mesh Networks,", 6th International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT'05), 2005, pp. 215-219.

[9] D. Seo, A. Ali, W. T. Lim, N. Rafique, and M. Thottethodi, "Near-Optimal Worst-Case Throughput Routing for Two-Dimensional Mesh Networks,", 32nd Annual International Symposium on Computer Architecture (ISCA'05), 2005, pp. 432-443.

[10] A. Lee and P. A. S. Ward, A Study of Routing Algorithms in Wireless Mesh Networks, ATNAC, December 2004.

[11] A. V. Mello, L. C. Ost, F. G. Moraes, and N. L. Calazans, Evaluation of Routing Algorithms on Mesh Based NoCs, Technical Report Series No.040, May 2004.

[12] J. Kleinberg, "The small-world phenomenon: An algorithmic perspective", In Proc. of 32nd ACM Symposium on Theory of Computing, 2000.

[13] D. Watt and S. Strogatz, Collective dynamics of small world network, Nature (393), 1998, pp. 440-442.

[14] C. Martel and V. Nguyen, Analyzing Kleinberg's (and other) Small-world models. In Proc. 23rd ACM Symposium on Principles of distributed computing, Canada, 2004, pp. 179-188.

[15] M. Naor and U. Wieder, Know the Neighbor's Neighbor: Better Routing for Skip-Graphs and Small Worlds. LNCS, Vol. 3279, 2005, pp. 269-277.

[16] F. Zou, Y. Li, L. Zhang, F. Ma, and M. Li, A Novel Approach for Constructing Small-world network in Structure P2P Systems. LNCS, Vol. 3251, 2004, pp. 807-810.

[17] H. Unger and M. Wulff, Search in Communities: An Approach Derived from the Physic Analogue of Thermal Fields, Proc. the ISSADS 2004, LNCS 3061, Mexico, 2004.

[18] R. Baumann, S. Heimlicher, V. Lenders, and M. May, HEAT: Scalable Routing in Wireless Mesh Networks Using Temperature Fields, IEEE Symposium on a World of Wireless, Mobile and Multimedia Networks, 2007.

[19] L. Lertsuwanakul, and H. Unger, "An Adaptive Policy Routing with Thermal Field Approach", In Proc. of 9th Innovative Internet Community Systems, Germany, 2009, pp. 169-179.

[20] L. Lertsuwanakul, "Fuzzy Logic Based Routing in Grid Overlay P2P Network", In Proc. of 10th Internaional Conference on Innovative Internet Community Systems, Thailand, 2010, pp. 140-149.

[21] L. A. Zadeh, "Fuzzy sets: Information and Control", Vol. 8, 1965, pp. 338-353.

[22] H.Coltzau, "Specification and Implementation of Parallel P2P Network Simulation Environment", Diploma Thesis, University of Rostock, 2006.

[23] S. J. Mirabedini1, M. Teshnehlab, and A. M. Rahmani, "FLAR: An Adaptive Fuzzy Routing Algorithm for Communications Networks using Mobile Ants", Proc. Int. Conf. on Convergence Information Technology (ICCIT 2007), 2007, pp. 1308-1315.

[24] F. Jing, R. S. Bhuvaneswaran, Y. Katayama, and N. Takahashi, "Adaptive Route Selection Policy Based on Back Propagation Neural Networks", Journal of Networks, Vol.3, No.3, March 2008, pp. 34-41.

IMS-centric Evaluation of IPv4/IPv6 Transition Methods in 3G UMTS Systems

László Bokor, Zoltán Kanizsai, Gábor Jeney
Budapest University of Technology and Economics-Department of Telecommunications (BME-HT)
Mobile Communication and Computing Laboratory – Mobile Innovation Center
Magyar Tudósok krt. 2, H-1117, Budapest, Hungary
{goodzi, kzoltan, jeneyg}@mcl.hu

Abstract - **The Internet Protocol is facing version change nowadays, IPv4 (the old version of IP) will be replaced by IPv6 (the new version of IP) in the near future. This transition strongly affects also wireless and mobile architectures due to widespread application of IP-based mobile networking architectures, the continuously increasing number of mobile Internet users, and the emerging convergence of different communication services driven by the IP Multimedia Subsystem (IMS). However both IPv6 and IMS are deeply covered in the existing literature as self-possessed researches, the challenge of provisioning IPv6-based IMS services over 3rd Generation (3G) Universal Mobile Telecommunication System (UMTS) networks as well as related problems and performance issues were not considered so far. In this work, we try to fill this gap and raise attention on the current questions and challenges of the transition from IPv4 to IPv6 in all-IP 3G and beyond multimedia systems. We analyze eight state of the art methods providing IPv6 support in existing mobile telecommunication architectures and evaluate their impacts on the network and service/application performance. In order to achieve this, we designed and implemented a real-life 3G UMTS-IMS testbed, and compared the characteristics of the selected transition techniques with native IPv6 and IPv4 scenarios from an IMS-centric point of view. Our results expose the main benefits and drawbacks of the evaluated technologies and their actually available implementations.**

Keywords - IPv4, IPv6, L2TP, OpenVPN, 6to4, ISATAP, Teredo, NAT-PT, IMS, all-IP, 3G UMTS, performance evaluation, real-life testbed, measurements

I. INTRODUCTION

IPv6 is the new version of the Internet Protocol and expected to be introduced for the wide audience in the next few years. IPv6 comes with a huge amount of improvements compared to IPv4; however it keeps the conceptual basics. For instance, IPv6 has built-in functionality for mobility management, while IPv4 has only an extension for this purpose (and it is usually not implemented). Thus, for mobile networks we believe that the appearance of IPv6 will extend provisioning systems, therefore evaluation of novel and advanced services over IPv6 is essential [1].

IPv6 was built on the same fundaments as IPv4: both represent a best effort service over a packet switched network [2]. Since IPv6 cannot be a global replacement of IPv4 (they are not compatible), it is expected that IPv6 and IPv4 will live together for approximately twenty years. In the short run, devices and networks will be dual stack, having both IPv4 and IPv6 supported. Later, some terminals and network segments might appear to support IPv6 only, and finally IPv4 will be regarded obsolete. Obviously, it must be a very long process. Thus, it is very important to see how IPv6 behaves compared to IPv4 in mobile networks. This article wants to discover some performance metrics of IPv6 in a mobile environment.

As the world tends to apply IP as the sole networking protocol, the role of mobile operators may turn simply into internet service providers. There are three facts, which should not be forgotten: 1) mobile services yields more income than Internet services, 2) mobile networks/services are centralized compared to some distributed internet services (e.g., P2P), and finally 3) distributed services are difficult to charge. Mobile service providers do not want to take a loser position in the next version of mobile networks, so a new centralized entity has been defined: the IP Multimedia Subsystem (IMS) [3]. IMS plays a central role in the network: it provides multimedia services to users, so users must use the IMS to have these services available/operational. Thus, IMS keeps being the centralized entity of mobile networks, where charging can be solved easily. IMS assures the future of mobile operators: using the IMS as an efficient instrument in the work of combining the new all-IP multimedia features with the benefits of IPv6, mobility support and multihoming, it becomes possible to provide an almost unlimited range of advanced, interactive multimedia services even for future scenarios.

One of the core aspects of the IP Multimedia Subsystem is the convergence on Internet protocols such becoming the main delivery platform for multimedia services throughout every kind of possible access networks. The technical background of this convergence is built upon two protocols, namely IP (v4 and v6) for data transport and Session Initiation Protocol (SIP) 0 for the negotiation and management of sessions. Since all users in an IMS enabled network must experience the performance metrics of key IMS operations, in this paper all the measurements are connected with basic IMS signaling and media delivery parameters.

This paper is organized as follows. Section II is the introduction part and this is the longest section of this paper. First an overview of 3G UMTS and IMS is given in Section II-A. Then, Section II-B details the specifics of IPv6 UMTS access: eight possible access methods (native IPv6, L2TP, OpenVPN over UDP/TCP, 6to4, ISATAP, Teredo and NAT-PT) are described in separate subsections. Section III introduces the performance metrics, which are used for comparison in the measurements. Section IV shows

the testbed where all the experiments have been done. There has not been any simulation, only real measurements with physical hardware have been applied. Section V describes the measured results. Finally, Section VI concludes the paper and shows some possible future work.

II. BACKGROUND

In this section we first introduce the basics of IMS and 3G UMTS architectures, then we present the existing most well-known and most-widespread protocols to set up and maintain IPv6 connection for end users in all-IP 3G (and beyond) systems. Performance characteristic of IMS over IPv6-capable 3G networks will be analyzed using these methods as they provide IPv6-based connection for IMS applications in next generation mobile telecommunication systems.

A. Overview of 3G UMTS and IMS

The major innovation presented by the 3rd generation mobile networks during the pending evolution of mobile telecommunication architectures was the introduction of the Wideband Code Division Multiple Access (WCDMA) technology on the air interface and the all-IP paradigm of the core. As a result, significantly higher bandwidth became available compared to 2nd Generation (2G) Global System for Mobile telecommunications (GSM) and 2G+ Global Packet Radio Subsystem (GPRS) and Enhanced Data rates for GSM Evolution (EDGE) networks and also converged service provision became possible. The 3rd Generation Partnership Project (3GPP) 3G UMTS architecture can be divided into three main domains: Circuit Switched (CS), Packet Switched (PS) and Registration domain. In the next generation converged IP services, the most important one of the above listed items is the PS domain. The Packet Switched domain relies on the basics that were set in the GPRS principles but it uses the IP protocol in a more sophisticated way. In the core network the most important entities for the PS access are the RNC, SGSN and the GGSN [5]. The RNC (Radio Network Controller) manages the available radio resources by assigning appropriate radio bearer to user to maintain optimum performance. The SGSN (Serving GPRS Support Node) is responsible for routing and mobility management while also taking part in the authentication process. The GGSN (Gateway GPRS Support Node) provides the connections towards any exterior IPv4 and/or IPv6 network as seen in Fig. 1.

When a subscriber wants to access PS services it needs to request a PDP (Packet Data Protocol) context that enables the subscriber to access the service based on the information stored in the HSS (Home Subscriber Service). The PDP context defines the APN (Access Point Name) where the user belongs to, which determines the IP address and QoS properties for that PDP context. In case the connection is successfully set up the traffic between the SGSN and the GGSN is transmitted in GTP (GPRS Tunneling Protocol) tunnels. Theses tunnels are used to differentiate the user traffic belonging to a PDP context until it reaches the GGSN.

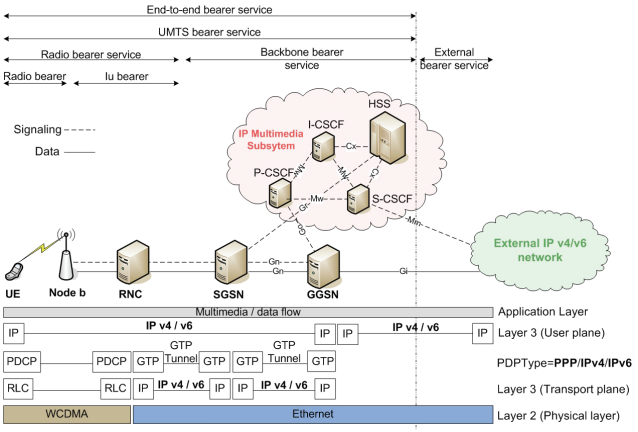


Figure 1. Overview of the all-IP 3G UMTS/IMS architecture

As the UMTS penetration has reached a critical level, research on advanced service provisioning standards was emerged to overcome the shortcomings of the already existing solutions. The core concept of IP Multimedia Subsystem (IMS) is to provide a comprehensive service provisioning framework for delivering IP multimedia to mobile users. As Fig. 1 shows, the IMS – one of the most important structural elements of all-IP systems in 3G and beyond – is an organic and integrated part of the 3G UMTS core network and also depends on the PS domain [6]. Initially designed for mobile networks by 3GPP, IMS has since evolved to also incorporate Next Generation Networks and the associated Fixed/Mobile Convergence vision: the Evolved Packet System (EPS). IMS enhances the basic IP connectivity of UMTS, provides flexible multimedia session management, media processing and control, and generally defines overlay architecture on the top of the packet switched core network incorporating the key converged, service and application oriented networks of the future [3]. All the above functions to control the multimedia sessions are implemented via different types of Call Session Control Functions (CSCF) using the Session Initiation Protocol (SIP) as a basis:

- Serving-CSCF: the session controller of the UE in the home network (like a GGSN).
- Proxy-CSCF: the local contact point of the UE in the visited network (like an SGSN).
- Interrogating-CSCF: the router of sessions in case of multiple S-CSCFs.

The flexibility of the above main architectural elements made IMS to be the common standard for next generation fixed (ETSI/TISPAN), cable (PacketCable) and mobile (3GPP, 3GPP2) networks, supporting the efficient delivery of multimedia data over any kind of access technologies (wired, wireless and mobile), and of course allowing operators and service providers to control the deployment, management and charging procedures of such convergent services.

B. IPv6 deployment and transition in 3G networks

The integration of the next generation Internet Protocol in today telecommunication architectures is a work in progress: IPv6 is still at its early stage of deployment. While rock-solid implementations are available for the most of the core network entities, solutions for mobile users to connect to IPv6 networks are sparse in the access (or the "last mile") segments of the networks. Besides 3G and beyond cellular networks these access architectures also include xDSL connection, cable connection and different heterogeneous access environments. Nowadays such systems usually provide the user with an IPv4-only connectivity, implying several drawbacks and incorporating different and often restrictive policies like limitation of possible mobility scenarios, restriction in the number of simultaneously active users, the application of private IPv4 addresses and NAT technologies, different firewall rules, etc. These disadvantages and drawbacks should be eliminated by serving the users with native IPv6 connection over existing access network technologies or by applying special solutions in order to provide IPv6 connectivity over IPv4. Only with doing this can IPv6 play its roles as the basis of new peer-to-peer services requiring advanced IP reachability and as the enabler of future innovation in converged mobile and wireless systems.

However, the transition from IPv4 to IPv6 and the regarding deployment questions are quite complex and affect many layers in the 3G UMTS/IMS architecture.

Considering the networking layer it is obvious that the IPv6 reachability is one of the fundamental needs in this context. The IPv6 network connectivity of a 3G user equipment (UE) can be provided either natively (i.e., by introducing native IPv6 in the user plane) or by applying one of the existing transition technologies on IPv4 (i.e., dual stack IPv4/IPv6 support in the affected network elements, tunneling, or IPv4-IPv6 protocol translators). In the first phase of the v4-v6 network transition several IPv6 islands will be interconnected by the IPv4 Internet using tunneling mechanisms. IPv4 only or dual stack UEs will use mainly IPv4 services and the rare IPv6 services provided to the users in this phase will be reached by tunneling (e.g., 6to4 [7] and ISATAP [8]) or protocol translation (e.g., Network Address Translation – Protocol Translation, NAT-PT [9] and Transport Relay Translator [10]). In the second phase we presume that IPv6 will be widely deployed over the Internet and numerous services will be based on the next generation IP protocol. However the deployment of IPv6 will be global in this phase, the IPv4 reachability will still be needed as the IPv6 Internet will not have full connectivity: several services will still exist only on IPv4 requiring dual stack implementations for efficient networking support. In the last phase IPv6 will achieve the dominant position. Due to global IPv6 connectivity all services will work on the IPv6 platform thus no dual stack functionality or other transition technique will be needed in the 3G and beyond architectures: native IPv6 will simplify the network architecture and will make possible to assign a unique, globally routable address to each and every user equipment in the network.

In the signaling layer two main aspects can be classified from the UE's point of view: IMS (i.e., SIP) signaling and Domain Name System (DNS) resolution. No issues of IMS signaling emerges in cases when the IMS callee (in the IMS context, callee refers to the called party) and caller are communicating over the same version of IP and the IMS itself supports the same version of Internet Protocol in the user plane. However, when an IPv6 user tries to call an IPv4 user (or reverse), translation in SIP and session negotiation (SDP) is to be applied by using application-aware translators as NAT-PT interworking with IMS Application Layer Gateways (ALGs) and special proxy servers acting as Back-To-Back User Agent (B2BUA). Regarding the DNS resolution the root of the problem lies on the fact that IMS procedures (e.g., registration, call-setup) strongly rely on the DNS database, as corresponding A/AAAA records provide the mappings of domain names of IMS entities to their IP addresses. In order to support this, a DNS ALG must be applied [11] or the DNS database has to be extended and modified with the appropriate AAAA entries and IPv6 mechanisms [12].

In the media layer questions similar to the signaling layer are to be answered as the main challenge here is also to handle v4/v6 heterogeneous situations (i.e., when an IPv6 user calls an IPv4 user or vice versa).

In this paper we focus on the state of the art user plane transition techniques (L2TP, OpenVPN UDP, OpenVPN TCP, 6to4, ISATAP, Teredo, NAT-PT) both able to efficiently deal with IPv6 provision over existing IPv4 3G UMTS architectures. The performance characteristics of IMS signaling and media transport will be compared over the above techniques using Native IPv6 and IPv4 scenarios as the basis of our comparison.

1) Native IPv4 / IPv6 3G UMTS

When a mobile user wishes to use packet switched (e.g., Internet) services in a 3G UMTS architecture, it must first attach to the network and then activate a PDP context. The UE receives its IP address during the activation of the PDP context and then it will be able to start the packet switched data communication. After the UE has been attached to the SGSN and it has been successfully authorized (i.e., the UE's identity has been checked and granted to access PS services), it must activate a PDP context (with appropriate IPv4 or IPv6 address) for commencing packet data communication. This is usually performed on application request (e.g., by starting a web-browser on the SmartPhone), but in some cases users may choose to be on-line for the whole time thus the packet data connection is established during or right after the boot sequence (e.g., registration into IMS). Users must specify on the UE the network service access point (i.e., the APN) of the Packed Data Network (PDN) they want to connect to and the PDP type (i.e., IPv4, IPv6, etc.) of the PDN they want to use. At the beginning of the PDP context activation, the UE puts the above parameters in an *Activate PDP Context Request* message and sends it to the SGSN. The SGSN uses the APN parameter to identify the corresponding GGSN for the requested PDN and makes it aware of the UE by the exchange of the *Create PDP Context Request* and *Create PDP Context Response* messages. As a

result, a two way point-to-point tunnel is established between the SGSN and the GGSN: activating a PDP address sets up a GTP association between the UE's current SGSN and the GGSN that anchors the PDP address. A special data record is created regarding the associations maintained between the SGSN and GGSN. This record is called as PDP context and describes the main parameters of the connection (e.g., network type, and address type, APN, Quality of Service, billing information, etc.). After creating or updating the PDP context, the SGSN sends an *Activate PDP Context Accept* message to the UE in order to inform the mobile about the assigned PDP address and other context-related information. After finishing the PDP context activation procedure, the UE starts the appropriate v4/v6 address setup or allocation mechanism (e.g., DHCPv4/v6, IPv6 stateless autoconfiguration, etc.) depending on the requested PDP type and the received PDP address value. As a result, a native IPv6 or IPv4 connectivity will be produced where the GGSN plays the role of the default gateway for the UE. (More details on the IPv4/IPv6 address allocation mechanisms in 3G UMTS architectures can be found in [2], [13].)

2) Layer Two Tunneling Protocol

Layer Two Tunneling Protocol (L2TP) of RFC3931 [14] was designed to provide a dynamic and effective mechanism for tunneling Layer 2 "circuits" across datagram-oriented communication systems (like IP networks). L2TP was originally defined in RFC 2661 as a standard scheme for tunneling Point-to-Point Protocol (PPP) [15] sessions over IP. It was also designed to terminate these PPP sessions in a defined concentration point (i.e., L2TP Access Concentrator) of the network. Since the release of the first version of the protocol, L2TP has been adopted for tunneling a number of other layer two protocols like Ethernet, Frame Relay and Asynchronous Transfer Mode (ATM). L2TP merges the functionality of two former proprietary tunneling methods for PPP, which are Cisco's L2F (Layer 2 Forwarding) and Microsoft's PPTP (Point to Point Tunneling Protocol) and operates in the Session Layer of the OSI reference model. The latest version of the protocol also incorporates advanced security features (L2TP/IPSec VPN protection), improved encapsulation and the possibility to carry extended circuit status attributes (to communicate finer-grained error states).

L2TP operates in two sublayers namely the control sublayer and the data sublayer. The control sublayer provides the reliability through packet numbering and acknowledgment system, while the data sublayer ensures the data transmission and detects any message loss using a sequence number. During its operation, L2TP simulates a specific data link layer and inserts every single data packet into a PPP frame before adding the L2TP encapsulation. Then the entire L2TP packet (including the payload and the L2TP header) is sent in a simple IP or in a UDP datagram. When L2TP operates directly over IP, L2TP packets cannot take advantage of the UDP checksum for checking packet integrity, which is important especially in case of L2TP control messages. Therefore L2TP usually applies UDP, in which messages will be transmitted using any IP network based on any data link connection between the two endpoints of the L2TP tunnel. These two endpoints are called the LAC (L2TP Access Concentrator) and the LNS (L2TP Network Server). The LAC plays the role of the initiator of the tunnel establishment while the LNS is the server continuously waiting for new tunnel requests. Every established L2TP tunnel between the peers is bidirectional and transmits higher-level protocols. In order to support this, an L2TP session (i.e., call) is established within the tunnel for each higher-level protocol such as PPP. Sessions inside an existing tunnel can be initiated either by the LAC or the LNS, and the traffic of each session is isolated by the L2TP. Note that this feature makes it possible to set up multiple virtual networks across a single tunnel.

L2TP is often used as a tunneling mechanism in xDSL and Cable architectures as a solution for selling/reselling endpoint connectivity: an L2TP tunnel sits between the user and the ISP the connection is to be sold/resold to, so the selling/reselling ISP will not appear as dealing with the transport functionalities.

In 3G UMTS architectures L2TP could be an effective way to provide IPv6 access over existing IPv4 technologies: the IPv4 PDP type traffic containing encapsulated IPv6 packets from the UE is processed at the GGSN, where the IPv4 sessions are terminated, then the GGSN transports this traffic over L2TP and then routes over Gi to their IPv6 destination.

3) Virtual Private Networks

Virtual Private Networking (VPN) is another method to provide IPv6 connection on an existing IPv4 only 3G UMTS architecture. In general, a Virtual Private Network is a special computer network that is implemented as supplemental software layer (i.e., overlay) on the top of an existing network infrastructure aiming to create an exclusive interconnection of communicating nodes or to provide a secure access to a private network by extending it into an insecure or shared/public architecture (like the Internet). Such overlay structures can be built by using different tunneling methods and by encrypting, decrypting and authenticating traffic inside the tunnels. OpenVPN is a well known and widespread VPN implementation also based on tunneling [16]. OpenVPN creates the secure tunnels using SSL (Secure Sockets Layer), which is a commonly-used protocol for securing Internet transactions in the application layer (HTTPS protocol also uses SSL for securing Internet transactions on the web). This protocol is one of the industrial standards for establishing VPNs, robust, quite easy to implement/manage by administrators and learn/understand by users.

The implementation of OpenVPN is based on the OpenSSL library, which realizes encryption, authentication and certification features for the secure tunnel and manages the SSL connection over TLS (Transport Layer Security) protocol to transmit data [17]. OpenVPN tunnels can be established between a client and a server and can run both over UDP and TCP. During the operation IP packets that need to be sent in the tunnel are encrypted and encapsulated in a UDP or TCP message. Then this packet can be transmitted using any IP network based on any layer 2 connection. The fact that OpenVPN is implemented as a

user-space daemon rather than a kernel module or a complex extension to the IP layer makes the method portable, easily deployable and configurable.

As OpenVPN is a cost-effective and lightweight alternative to other VPN technologies, it is commonly applied at Small and Medium Enterprises (SMEs) and also well targeted in the enterprise markets.

In order to provide IPv6 connectivity for UEs in an IPv4 3G UMTS network, a secure OpenVPN tunnel can be used, which encapsulates the IPv6 traffic and relays it to the UE through the IPv4 networking segment. In such a scenario the dual stack UE operates in IPv4 mode (it opens an IPv4 type PDP context and receives an IPv4 address from the GGSN), but also an OpenVPN tunnel is spanned over this IPv4-only connection between the UE and the tunnel server. This tunnel is the gate to a VPN, which basically extends the IPv6 connectivity into the IPv4 3G UMTS network.

4) 6to4

In RFC3056 [7] authors specify a scheme for IPv6 sites to communicate with each other over an existing IPv4 network without explicitly given tunnel endpoint information. 6to4 does not use IPv4-compatible IPv6 addresses (where the prefix ::/96 is separated for IPv4-compatible addresses, and the rightmost 32 bits of the IPv6 address stand for the IPv4 address of the destination) but it has a proper IPv6 address format that includes the IPv4 address of the tunnel endpoint in the prefix such allowing automatic tunnel setup. In 6to4 the transport IPv4 network behaves as a unicast point-to-point link, and the 6to4 domain segments communicate via 6to4 routers (i.e., 6to4 gateways): IPv6 packets are encapsulated and decapsulated here requiring at least one globally unique IPv4 unicast address. Only the gateways need to be 6to4 compatible, therefore no other changes have to be made to the IPv6 nodes inside the 6to4 network. The prefix for the 6to4 protocol assigned by the IANA organization is `2002::/16` providing 6to4 addresses in the `2002:IPv4Adrr::/48` structure. It is important to notice, that if a host in a 6to4 network wants to exchange packets with a host in another 6to4 network, no tunnel configuration is needed: the tunnel entry point can take the IPv4 address of the tunnel exit point from the IPv6 address of the destination. Besides the above, a 6to4 relay router is needed for a successful communication with an IPv6 node in a remote IPv6 network. The relay router is a router configured for 6to4 operation and also IPv6 connection. The relay router connects 6to4 networks to the native IPv6 network as the `2002::/16` prefix is announced into the native IPv6 network by such relays.

As an extension to the basic standard, RFC3068 [18] specifies a 6to4 relay router anycast address in order to optimize the configuration of 6to4 gateways, which require a default route towards a 6to4 relay router on the IPv6 Internet.

The application of the 6to4 technique in mobile telecommunication architectures is twofold. On one hand sites offering IPv6 mobile access can be connected with each other and with the IPv6 world through IPv4 using 6to4. Here the operation of the transition technique is transparent to the IPv6 mobile UEs: they only have to be configured with at least one 6to4 IPv6 address in the `2002:IPv4Adrr:SubnetID::/64` format. On the other hand 6to4 tunnels can be spanned right between a 6to4-compatible dual-stack UE and the 6to4 relay router over the IPv4-only 3G UMTS network, such providing encapsulation-based IPv6 support while still using IPv4 PDP contexts. In this case the 6to4 relay router resides inside the operator network on the v4/v6 domain boundary.

5) ISATAP

The Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) is specified in RFC4214 [8] aiming to provide IPv6 connectivity for dual-stack hosts over an IPv4-based networking infrastructure. This technique also uses the existing IPv4 network as one large link-layer architecture and allows the dual-stack hosts to automatically create tunnels and exchange data between themselves. ISATAP can be used regardless of whether the hosts have global or private IPv4 addresses. Addresses of this automatic tunneling mechanism embed an IPv4 address in the EUI-64 interface identifier in the following format:

`64bitPrefix:16bitControl:5EFE:IPv4address.`

ISATAP interfaces form ISATAP interface identifiers using their IPv4 addresses and apply them to produce the ISATAP link-local addresses in order to make the technique able to perform standard IPv6 neighbor discovery mechanisms. Using this method, IPv6 nodes inside an IPv4 intranet can communicate with each other. If hosts want to communicate with IPv6 hosts outside the intranet (e.g., 6Bone hosts), a border router must be configured, which can be an ISATAP router or even a 6to4 gateway. An important issue of this method is that all hosts in an ISATAP network need to support the ISATAP protocol.

ISATAP (together with 6to4) are considered as two really promising and already popular transition technologies evaluated and assessed within several real-life testbed experiments and projects like [19], [20] and [21]. In IPv4-only 3G UMTS architectures ISATAP can be used as an automatic tunneling solution for dual-stack UEs that are multiple IPv4 hops away from the IPv6 network. Mobile terminals can build tunnels between each other and exchange IPv6 traffic using their link-local addresses: the packets are transmitted via ISATAP tunnels with endpoints that are derived from the interface ID segment of the link-local addresses. For outside (or offlink) IPv6 traffic UEs have a default route, pointing to the ISATAP address of the ISATAP router.

6) Teredo

Teredo is specified in RFC4380 [22] as an IPv6 transition technology providing address assignment and automatic host-to-host tunneling for unicast IPv6 traffic in cases when IPv6/IPv4 nodes are placed behind IPv4 network address translators (NATs). Comparing Teredo with 6to4 and ISATAP we can summary that 6to4 makes IPv6 available over an IPv4 network using public IPv4 addresses, ISATAP helps deployment of IPv6 nodes within a site regardless of whether it applies private or public IPv4 addresses, and Teredo makes IPv6 available to nodes through any number of NAT layers using UDP-based tunneling. The Teredo

architecture consists of a Teredo server (a well-known host, which is used for initial configuration of a Teredo tunnel helping clients to access IPv6 networks), several Teredo clients (running on an IPv4/IPv6 dual-stack terminal in an IPv4 network behind a NAT) and Teredo relays (the remote end of a Teredo tunnel forwarding IPv6 traffic between a Teredo client and a host in the IPv6 network). The technique introduces a special prefix called Teredo Service Prefix (`2001:0000::/32`), which is announced by the Teredo relays to the outside world using conventional IPv6 routing mechanisms. Based on this prefix each Teredo client assigns a public IPv6 address that is constructed as follows:

```
2001:0000:ServerIPv4:Flags:UDPport:ClientIPv4.
```

A significant part of RFC4380 deals with how Teredo identifies the specific type of NAT deployed in the actual network and defines mechanisms for handling these various NAT types.

During the protocol's basic communication procedure first the Teredo client inside the IPv4-only domain starts the determination of the Teredo relay serving the IPv6-only host by sending out an *IPv6 Echo Request* message via the Teredo server. This request is forwarded to the IPv6-only host, which answers it with an *IPv6 Echo Reply* message destined to the Teredo client's address and routed to the to the nearest Teredo relay. The Teredo relay tunnels the reply message to the client that now determines the relay IPv4 address and starts sending packets to the IPv6-only host via the relay. The Teredo relay decapsulates the IPv6 packet and forwards it to the IPv6-only Host.

Based on the above operation Teredo solves numerous problems of IPv4-IPv6 transition. However, the current version of the standard does not work with symmetric NATs. In order to support Teredo for symmetric NAT traversal, authors of [23] proposed SymTeredo, which imposes minor modifications on the Teredo relay and the Teredo client components but also keeps compatibility with the standard protocol.

3G operators can rely on Teredo's efficient and NAT friendly IPv4-IPv6 transition toolset by introducing the components of the Teredo architecture in the UMTS network. However, as Teredo can only provide a single IPv6 address per tunnel endpoint, it is not possible to use a single Teredo tunnel to make connection with multiple nodes (contrary to 6to4), such creating significant tunneling overhead on the air interface in several common scenarios. The application of Teredo –similarly to the majority of the above schemes– still not transparent: it requires additional UE configuration and installation of supplementary software modules (i.e., Teredo implementation) on the UE. Nevertheless, the big number of Teredo implementations that are already available for the widest scale of operating systems (Linux, *BSD, Mac OS X, Windows XP SP2/Server 2003/Vista and Windows 7) may assume that popular UE platforms will introduce Teredo functionality.

7) NAT-PT

RFC2766 [24] introduces the Network Address Translation - Protocol Translation (NAT-PT) transition scheme, which uses a pool of public IPv4 addresses for dynamic assignment to IPv6 hosts, and employs a stateful IPv4/IPv6 header translation on a special network device located at the boundary of the IPv4 and the IPv6 networks. This NAT device translates IPv6 packets into analogous IPv4 packets and vice versa, and such routes between an IPv6 network and an IPv4 network. NAT-PT reserves the pool of IPv4 addresses and translates the fields for IP Source addresses, IP, TCP, UDP, and ICMP header checksums. Note that in order to achieve this behavior, NAT-based v4/v6 transition schemes usually apply IPv4/IPv6 header translation rules specified in RFC2765 (Stateless IP/ICMP Translation) [25].

An extension of NAT-PT is Network Address Port Translation - Protocol Translation (NAPT-PT), which further extends the original idea: in order to allow numerous IPv6 hosts to share one single IPv4 address for multiplexing multiple sessions on one address, transport identifiers (such as TCP and UDP port numbers) are also translated in this technique.

The main benefit of NAT-PT and NAPT-PT is that no changes are required to end hosts because all the translation procedures are executed at the separate NAT device in the network. However the mechanisms defined in RFC2766 seem to be convenient in several transition scenarios, serious issues exist with the standard. For example, NAT-based schemes cannot take full advantage of the enhancements offered by IPv6, and it is really hard to maintain the big number of Application Level Gateways (ALG) needed in NAT devices to keep the widest scale of applications working correctly through the gateway. The raised problems are summarized in RFC4966 [26] together with the conclusion that technical and operational difficulties resulting from these issues make it undesirable to recommend the usage of RFC2766 as a general purpose transition mechanism. However, the transparent nature of NAT-PT/NAPT-PT (i.e., the fact that clients don't need to be modified for benefitting from the method's IPv4-IPv6 transition services) makes suitable the technique for application in mobile telecommunication systems.

In 3G UMTS networks NAT-PT or NAPT-PT can be deployed by installing a NAT device and the appropriate ALGs at the boundary of the IPv4/IPv6 network segments. Configuration and modification on UEs is not required, only the suitable DNS server settings must be provided for the terminals.

III. PERFORMANCE METRICS

The main motivation of our work was to compare native IPv4 3G UMTS network performance with different IPv4-IPv6 transition methods (including the native IPv6 communication itself), using essential parameters of IMS operations as performance metrics. These measured parameters, which substantially affect the network performance in IMS based multimedia-centric user scenarios are the following: the round-trip time, the IMS registration time, the call setup time, and the downlink RTP delay.

A. Round-trip Time

The round-trip time (RTT) is the time elapsed while a transmitted packet arrives back from the recipient, if the packet is forwarded back immediately. This parameter is useful to examine the minimum response delay between two communicating nodes.

We used the `ping` application with 64byte packets to measure the round trip delay between the UE (sender) in the 3G network and the CN (recipient) in the outside PDN. The results of RTT measurements are corresponding as the main performance metrics of the examined architectures in the four scenarios.

B. IMS Registration Time

Registration is one of the most important procedures in next generation IP multimedia systems since this mechanism makes possible to initiate sessions between users in the network and to receive data from media and application servers.

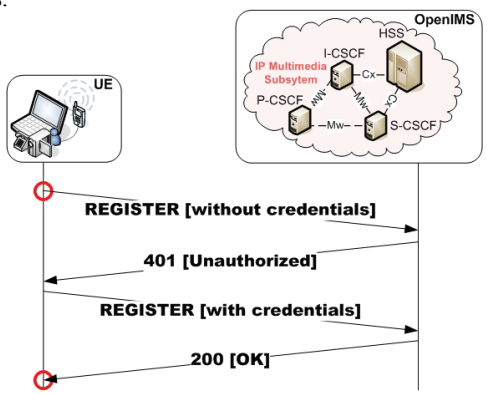


Figure 2. IMS Register Flow

To measure the time required to register a user inside the IMS in a 3G UMTS architecture we used `SIPp`, which is a traffic generator tool for the Session Initiation Protocol (SIP) [27]. The simplified schematics of the message exchange of an IMS registration procedure is shown in Fig. 2. The registration starts with a REGISTER message sent by the UE. A 401 UNAUTHORIZED message reaches the UE after the IMS processed the initial REGISTER in order to challenge the UE to send the required authentication information. After that an extended REGISTER message is transmitted on the same path as the first one. This message now contains all the required data to authenticate the UE. The IMS indicates the successful registration with a 200 OK message. (Further details on the IMS registration procedure can be found in [28].)

Appropriate `SIPp` scripts were executed on the UE in order to manage the REGISTER procedure and to control the flow of synthetically produced SIP packets between the UE and the IMS system. IPv4 or IPv6 addresses of IMS entities (e.g., P-CSCF) were provided by the DNS server.

In this context we considered the registration time as the elapsed time between sending the first REGISTER message and receiving the 200 OK message in the UE side (see the red markings in Fig. 2).

C. Call Setup Time

Right after a successful registration, IMS subscribers of a 3G UMTS system can initiate IMS calls to other subscribers or media providers. An outline of the IMS call setup flow is depicted in Fig. 3. (The detailed flowchart can be found in [28].)

The caller UE starts the call setup procedure by sending an INVITE message to the P-CSCF with the CN's user name and the SDP descriptors in it. This message is forwarded to the CN by several IMS mechanisms leading the CN to reply by sending a 183 SESSION IN PROGRESS message containing SDP descriptors. Also some informal messages are exchanged (100 TRYING, 180 RINGING) during the procedure, and finally a 200 OK arrives back to UE, which means that the callee (i.e., the CN) accepted the call. This fact is acknowledged by an ACK message, which is sent by the UE to the CN (and the S-CSCF) through the P-CSCF. When the CN receives the ACK message, the call setup is finished and the Real-time Protocol (RTP) [29] datagram exchange starts between the communicating peers. This metric can also be measured using `SIPp` on UE and CN entities in order to generate and manage IMS signaling in the context.

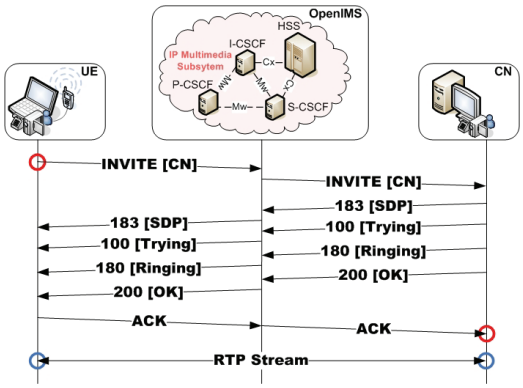


Figure 3. IMS Call Setup and RTP delivery

As because the call setup time is the elapsed time between the first INVITE message (sent by the UE) and the ACK message (arrived at the CN) (see red markings in Fig. 3), there is a strong need to synchronize the clocks of the two nodes to get precise results. This time synchronization can be achieved by NTP (Network Time Protocol). In order to avoid NTP inaccuracy and undesirable drifting, we introduced a dedicated "shadow" network for the NTP signaling between a local NTP Server and the UE/CN nodes. Based on this scheme we achieved an approximated accuracy of ±30µs, which offers sufficient error margin for the measurements presented in the article.

D. One-way RTP Delay

When the call setup is finished, RTP packets are starting to be exchanged between the two communicating peers. Because of the nature of services, the RTP data flow is often unidirectional, usually in downlink direction (e.g., in case of a video or audio streaming). Therefore we measured the

downlink, one-way RTP delay as the most significant performance metric of the media plane.

In our scenarios the CN played the role of the media server and the UE was the subscriber to an audio streaming service, which provided a 192kbps Constant Bit Rate (CBR) audio source.

As in the case of the previous metric, here also a time interval between events occurring on two different nodes (see the blue markings in Fig. 3) have to be examined, so the dedicated NTP network must be introduced here too for accurate measurements. RTP packets can be captured by packet analyzers (e.g., `tshark` [30]), and the time stamps of the sent and received packets can be used for calculating the RTP delay.

IV. MEASUREMENT ARCHITECTURE AND SCENARIOS

In this section we introduce our testbed and the scenarios used to compare the main performance metrics of IMS operations over different IPv6 provision techniques in 3G UMTS networks. In the first subsection our native IPv4/IPv6 3G UMTS network is described in details, followed by the eight measurement scenarios: native IPv4, native IPv6, L2TP IPv6, OpenVPN IPv6, 6to4, ISATAP, Teredo, and NAT-PT respectively.

A. Overview of the Testbed

In order to provide a testbed for advanced IPv6 mobility and multihoming researches and analyzing IPv6 deployment and v4-v6 cohabitation/transition issues in next generation multimedia-centric communication systems, we designed and implemented a native IPv6 UMTS/IMS architecture based on the existing hardware elements of Mobile Innovation Centre (MIK) located in Budapest, Hungary [31]. However almost all the relating hardware and software components were presented in MIK, one important item was missing: the laboratory did not possess any dedicated Gateway GPRS Support Node (GGSN) device for supporting native IPv6 UMTS access. Thus one of the main tasks during the implementation of our UMTS/IMS testbed was to design and develop a GGSN, prepared to be integratable with the other UMTS elements and adequate to handle also IPv6 type PDP (Packet Data Protocol) contexts besides IPv4. In order to achieve this, we used a software GGSN implementation called OpenGGSN [32] as a basis of our work. Our GPL licensed and publicly available OpenGGSN modification (OpenGGSN 0.84_v6_05 [33]) uses the same GTP library and the main architecture as version 0.84, but extends the original edition with the missing IPv6 routines and some other related components for setting up, maintain and tear down contexts of native IPv6 UMTS communication.

The integration of our IPv4/IPv6-compatible (i.e., dual-stack) GGSN software into the UMTS/IMS testbed architecture for providing also native IPv6 packet exchange was a six-step procedure. First, we had to create a new, IPv6-compatible APN in the SGSN, than we had to enable also IPv6 PDP contexts for the SIM cards of our devices in the Home Subscriber Server (HUAWEI HSS 9820). After that we compiled, configured and started all the required OpenGGSN 0.84_v6_05 components on a SunFire X4200 (powered by AMD Opteron™ processors, 4GB RAM, and running Ubuntu 7.04 Feisty Fawn operating system with kernel 2.6.23). As the 4th step we deployed an open-source software IMS implementation called Fraunhofer OpenIMS [34], which realizes all the functional entities (HSS and all CSCFs) of IMS architecture and supports both IPv4 and IPv6. We used version 604 of OpenIMS with a Debian 5 (Lenny) operating system and kernel 2.6.26 on a SunFire X4150 server comprising 2.83GHz Intel™ Dual Quad-Core Xeon E5440 processors and 8GB RAM. Step No. 5 was the configuration of end terminals, while the last step was setting up the appropriate IPv6 routing entries in the routers of the testbed in order to provide outside IPv6 PDN (i.e., GEANT) connection to the mobiles. Fig. 5 shows all the details of the native IPv6 UMTS/IMS architecture we used for our native IPv6 experiences, while Fig. 4 presents the details of the native IPv4 3G UMTS testbed. Note, that these two figures represent one, integrated, dual-stack tested system basically under the same architecture (with the same OpenGGSN 0.84_v6_05): using our OpenGGSN modification both IPv4 and IPv6 PDP contexts can be handled such creating a highly configurable all-IP 3G testing environment making possible to observe, measure and even modify every kind of IP-level function, traffic or operation.

The core UMTS infrastructure in our laboratory consists of one Node B and one RNC linked to the SGSN, which is connected to the GGSN and the HSS using standard interfaces. As Fig. 4 and 5 show, the SGSN and the GGSN are still communicating over IPv4 (i.e., the GTP tunnels are set up on IPv4), but this fact has no effect on the UE's context: either native IPv6 or native IPv4 UMTS connection can be provided, the mode of communication between the GSN nodes (i.e., the transport plane) does not have any impact on the type of user plane communication. The GGSN is connected to the outside (v4 or v6) network through its Gi interface.

For accessing this UMTS/IMS architecture, a dual-stack UE has been constructed from conventional hardware building blocks and equipped with the appropriate software components. UE's hardware is based on an ASUS V6800VA notebook with a Nokia N95 8Gb SmartPhone as an IPv6-compatible, dual-stack 3G modem for UMTS connectivity. The UE's operating system is a Ubuntu 8.04 LTS equipped with IPv6-capable Point-to-Point Protocol daemon (pppd v2.4.4) and SIPp v3.1 for managing the synthetic IMS signaling and media traffic. The CN is a Fujitsu Siemens Scienic SE PC with 3GHz Intel™ Pentium 4 processor, 2GB RAM, double Ethernet LAN adapter and the same software components as on the UE.

B. Measurement scenarios

In the previous subsection we presented the general structure of our dual-stack 3G UMTS/IMS architecture. In order to implement different measurement scenarios we applied several modifications and added some new entities for dealing with scenario-specific functions. These modifications and architectural changes are described in the following paragraphs.

1) Native IPv4

The testbed setup for the native IPv4 scenario is shown in Fig. 4. The UE uses the Nokia N95 8Gb smart phone as 3G wireless interface and connects through the 3G PS/IMS domain to the wired Correspondent Node (CN), which will be the communication partner of the UE during the measurements. An important node is not presented by Fig. 4 although it has a significant task not only here but also in the further scenarios: the Network Time Protocol (NTP) Server providing time synchronization for nodes under measurement is connected to the UE and the CN by a wired "shadow" network. The NTP server itself is a desktop PC running Ubuntu 8.04 LTS with NTP v4.2.4p4.

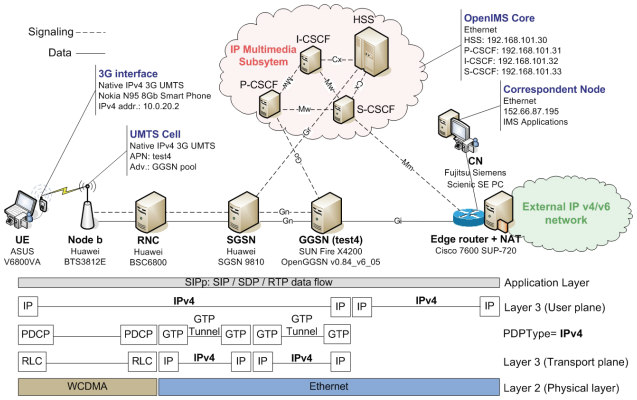


Figure 4. Native IPv4 3G UMTS/IMS testbed architecture

As introduced in the previous sections, the Packet Data Protocol (PDP) context offers a packet data connection over, which the UE and the network can exchange IP packets. In this scenario an IPv4 PDP context is used to build up native IPv4 user plane communication sessions between the UE and the PDN (i.e., the IPv4 Internet). The GGSN provides `10.0.20.2` address from its pool to the UE for IPv4 PS communication. Due to this and the limited number of available IPv4 addresses we also turn on NAT functions in our edge router for assuring outside communication of User Equipments.

The OpenIMS and the related DNS entries for the HSS and the CSCF sub-entities were configured to be reachable with IPv4 addresses. The used APN was *test4*, which identifies the IPv4 PDN in our testbed and the OpenGGSN software is responsible to implement its functions.

2) Native IPv6

The native IPv6 3G UMTS network is basically the same as the IPv4 version. The main difference is the usage of IPv6 PDP contexts for the UE in order to establish and maintain native IPv6 user plane communication (Fig. 5). It can be achieved by specifying IPv6 for the type of PDP context to be created. The UE's IPv6 compatible 3G modem interface can easily be instructed to do this using an appropriate AT command (that is `AT+CGDCONT=1,"IPV6","test6",,0,0` in our testbed setup). As it can be seen, the requisited APN was also modified from *test4* to *test6* (belonging to the IPv6 PDN). Thanks to this, the UE is aware of that an IPv6 PDP context is to be created and will send and *Activate PDP Context Request* message with `PDP type=IPv6` towards the SGSN. The SGSN sends a *Create PDP Context Request* message to the GGSN, which answers it with a *Create PDP Context Reply* containing an IPv6 address in the `PDP address` field of the message. This address will be passed to the UE in an *Activate PDP Context Reply* by the SGSN. The UE extracts the interface identifier part from the received IPv6 address, creates its IPv6 link-local address (`fe80::1234:1234:1234:1234`) and sends an IPv6 *Router Solicitation* message to the GGSN. The GGSN replies with a *Router Advertisement* containing an appropriate IPv6 networking prefix (`2001:738:2001:20a9::/64`). Using this advertisement and the previously get link-local identifier, the UE is able to generate its global IPv6 unicast address for the IPv6 communication. Note, that our native IPv6 3G UMTS testbed only supports the above mechanism (i.e., the IPv6 stateless address autoconfiguration) and no DHCPv6 is supported at the moment.

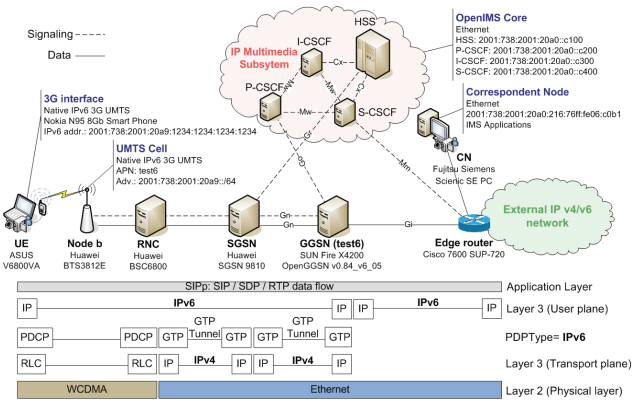


Figure 5. Native IPv6 3G UMTS/IMS testbed architecture

All the procedures shortly introduced above are taking part from the standard operations of a native IPv6 UMTS system, thus the implementation of these functions was mandatory for our OpenGGSN 0.84_v6_05 [33] implementation. However, we took advantages of some simplification possibilities during the design of our dual-stack GGSN software in order to reduce the development time and the requested human resources. These simplifications are mainly connected to the address allocation procedures and the QoS-related functions. More details on our OpenGGSN development and on IPv6 PDP context management in 3G and beyond architectures can be found in [33] and [13], respectively.

After the successful IPv6 context activation and address configuration, the UE is able to natively communicate with the IPv6 IMS domain, with other network entities or nodes in the IPv6 Internet (e.g., the IPv6 CN). Thanks to the tremendous number of available addresses and the nature of IPv6 in general, there is no need to apply NAT for outside communication in this scenario.

The OpenIMS and the DNS entries for the HSS and CSCFs must be configured to use IPv6 addresses. It is not shown but the NTP server still provides time synchronization service over the dedicated "shadow" network for UE and CN nodes.

3) L2TP IPv6

The Layer-2 Tunneling Protocol (L2TP) [14] scenario is built upon the native IPv4 scenario (Fig. 6). After initializing the native IPv4 3G UMTS user plane communication, the UE – configured as an L2TP Access Concentrator (LAC) in the `l2tp.conf` – searches for an L2TP Network Server (LNS) and sets up an unsecured L2TP tunnel over IPv4 in order to transport IPv6 packets on it. It means that on our Linux-based UE a novel Point-to-Point interface (`ppp1`) will be created besides the PPP interface used by the 3G UMTS connection (i.e., `ppp0`). The Router Advertisement Daemon (radvd-1.6) [35] running on the LNS will send periodic Router Advertisements through the PPP tunnel towards the UE, which will be able to configure its global address (`2001:738:2001:20a9:2c4b:931f:144e:1478/64`) with stateless autoconfiguration. The LNS in this scenario is the SunFire X4200 server, which also acts as the *test4* GGSN for the IPv4 PDN and runs Roaring Penguin v0.4 user-space implementation of L2TP such as the UE [36].

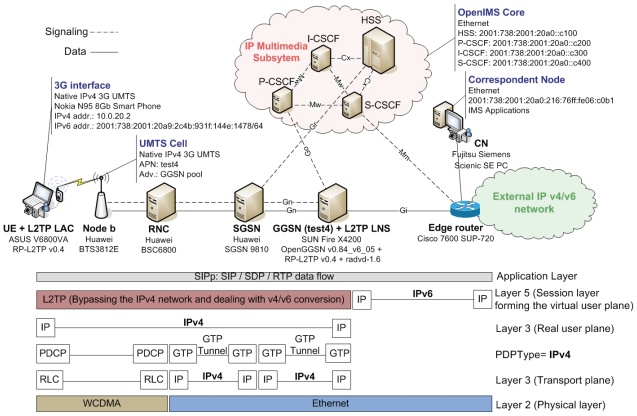


Figure 6. L2TP IPv6 3G UMTS/IMS testbed architecture

As shown in Fig. 6 the L2TP tunnel spanned in the session layer forms a virtual user plane where L2 data frames (Ethernet in our case) are accepted and forwarded. The L2TP tunnel uses UDP datagram to send the L2TP header and the payload to the two endpoints (LAC, LNS). The IPv6 packets are encased into this type of UDP packets and sent through the tunnel as IPv4 packets. Accordingly, the CN, the IMS and the DNS need to be reachable on IPv6 for the measurements.

The "shadow" network for NTP is used again in this scenario in order to synchronize the UE/CN nodes.

4) OpenVPN IPv6

This scenario uses OpenVPN [16] to create an encrypted point-to-point tunnel between the UE and the gateway towards the IPv6 PDN and supports IPv6 communication over a built IPv4 3G UMTS user plane based on both TCP and UDP transport protocols. The scenario topology is almost the same as the previous one, but here OpenVPN (v2.1_rc11 both on the UE and the GGSN) is used to create an application level IPv6 on IPv4 tunnel (Fig. 7).

After setting up our own Certificate Authority (CA) and generating certificates and keys for the OpenVPN server running on the same host as the GGSN and for the OpenVPN client of the UE, we created both the server and client configuration files (`openvpn.conf`). Here we specified the transport protocol (UDP or TCP) and the device (`tun0`) to be used, and edited the *ca*, *cert*, and *key* parameters. The upcoming step of constructing this measurement scenario was the startup of the VPN over the built IPv4 3G UMTS connection by running `openvpn` both on the UE and GGSN nodes. The assembled VPN connectivity makes possible to send `radvd` *Router Advertisements* from the GGSN to the UE over the `tun0` interface. Eventually this enables the UE to configure its IPv6 address for global communication (`2001:738:2001:20a9:d42d:d4ff:fe28:cb6b/64`).

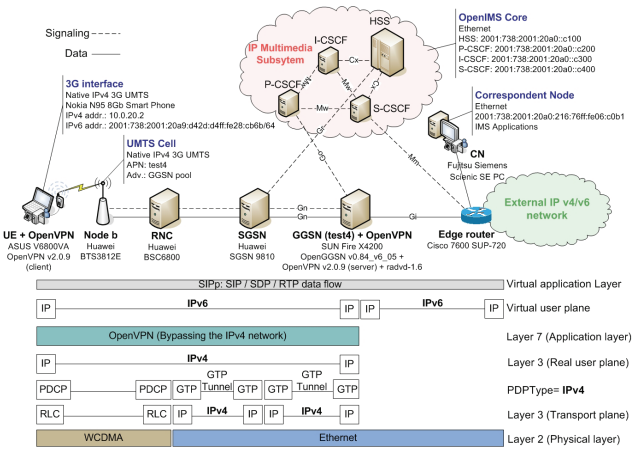


Figure 7. OpenVPN IPv6 3G UMTS/IMS testbed architecture

We measured the performance of the key IMS operations over both TCP and UDP based OpenVPN tunnels. The measurements were supported by NTP using the same dedicated time synchronizer network as in the above scenarios. Since it is also an IPv6 scenario, the CN, the DNS and the IMS needs to comprise IPv6 reachability for the measurements.

5) 6to4

In this v4-v6 transition scenario the dual-stack UE was also a 6to4 router: it was configured to support the use of a 6to4 tunnel interface and to forward 6to4-addressed traffic between itself and a 6to4 relay over the IPv4 3G UMTS connection. Since 6to4 routers require additional configuration and processing logic for encapsulation and decapsulation, the operation of such 6to4 compatible UE cannot be transparent. We used the Linux kernel implementation of the 6to4 protocol and our setup was based on the descriptions and guidance of [37].

The UE's 6to4 prefix was `2002:0A00:1402::/48` derived from the `2002::/16` IPv6 prefix and the IPv4 address `10.0.20.2` acquired during its pure IPv4-type PDP context activation. We assigned the suffix `::1` to this entity, such creating the IPv6 address of the UE, which equals with the IPv6 address of the 6to4 tunnel spanned between the UE's and the GGSN's IPv4 address.

As the GGSN is the IPv6/IPv4 entity that must forward 6to4-addressed traffic between 6to4 routers (i.e., UEs) inside the 3G UMTS network and IPv6 hosts on the IPv6 Internet, it also applies 6to4 relay functions.

The created 6to4 tunnel maintained by the 6to4 router and relay (i.e., the UE and the GGSN respectively) provides the virtual user plane making able the UE to perform IPv6 communication with the IMS core and the CN (Fig. 8).

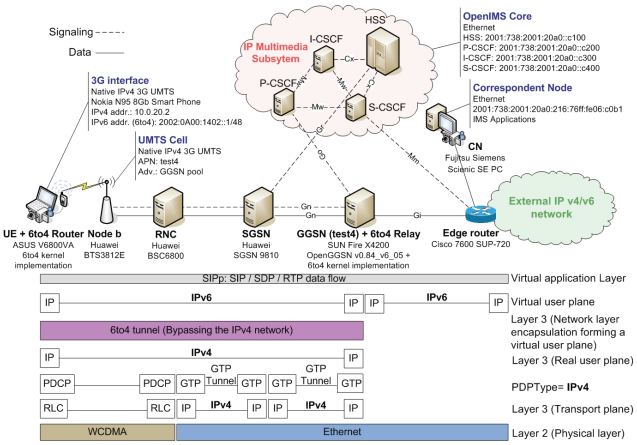


Figure 8. Application of 6to4 in our 3G UMTS/IMS testbed

The dedicated "shadow" NTP network for UE and CN time synchronization was implicitly applied also in this measurement scenario.

6) ISATAP

The ISATAP-based v4-v6 transition scheme was built upon the Linux in-kernel ISATAP support firstly introduced in kernel version 2.6.25. In order to make this implementation work, the GPLv2 licensed isatapd-0.9.6 [38] was installed on the client side (UE) and a static ISATAP tunnel device with `radvd` [35] support was configured on the ISATAP router.

The `isatapd` module on UE creates and maintains ISATAP tunnels by taking care of the following tasks:

- Constructing ISATAP tunnel device(s) based on IPv4 interface(s)
- Periodically querying and adding router addresses to the potential ISATAP router list
- Periodically sending router solicitation messages to potential ISATAP routers to get on-demand router advertisements for maintaining IPv6 connectivity
- Receiving and parsing incoming router advertisements in order to adjust the router solicitation interval
- Detecting link changes and maintaining ISATAP tunnel(s)

The configuration of the ISATAP router was performed on the GGSN with Linux command line tools: we had to statically set up an ISATAP tunnel device, then configure an ISATAP compatible address for it and start `radvd` on it for on demand advertisements.

Applying the above steps in our testbed the dual-stack, ISATAP compatible UE with only IPv4 PDP context in the 3G UMTS network was able to construct its ISATAP address (`2001:738:2001:20a9::5efe:0a00:1402/64`) and to bypass the IPv4-only segment by connecting to the ISATAP router using the `isatapd` module and mechanisms introduced above.

The prepared measurement architecture for the ISATAP scenario can be seen on Fig. 9 (note that the separated NTP network is not shown here).

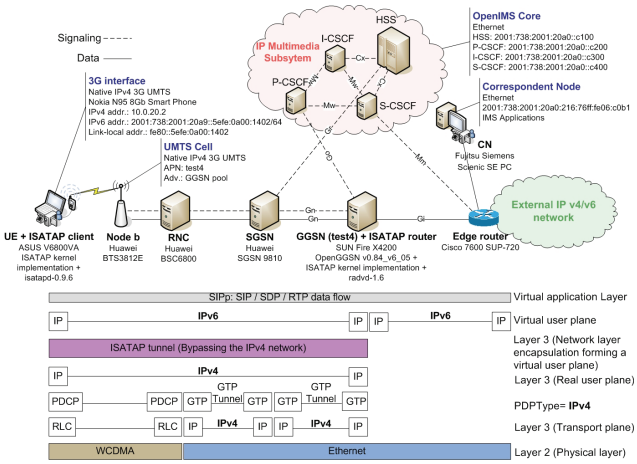


Figure 9. ISATAP-based v4-v6 transition in our 3G UMTS/IMS testbed

7) Teredo

This scenario uses Miredo [39] to provide Teredo client/server/relay functions in our 3G UMTS testbed. Miredo is an open-source Teredo IPv6 tunneling software for Linux and *BSD operating systems. It requires TUNTAP driver (`CONFIG_TUN`) and IPv6 stack support in the kernel, and realizes functional implementations of all components of the Teredo standard (client, relay and server). We installed miredo-1.1.3 on both the UE (with Teredo client functions) and the GGSN (for Teredo server and relay operations). See Fig. 10 for the detailed scheme of our Teredo-extended 3G UMTS testbed architecture.

The installation of Miredo on the UE was performed from binary package. As client mode is the default Miredo behavior, we added only the `ServerAddress` directive in the UE's `miredo.conf`. According to the Miredo implementation the UE first authenticates with the Teredo server (using the information given in `ServerAddress`), and if successful, it sets up the Teredo tunneling interface with the public Teredo address (`2001:0000:9842:578d:100e:598a:0a00:1402`) and the default IPv6 route constructed/calculated by the implementation. Hereafter, this virtual networking interface will be used to reach the IPv6 Internet and other Teredo clients.

As the Teredo server needs two subsequent IPv4 addresses for operation (it waits for UDP IPv4 packets on port 3544 on both addresses), we set up an additional public IPv4 address on the GGSN's Gi interface besides the "normal" IPv4 address and the IPv6 connectivity. The `miredo-server.conf` was used to specify the primary and the secondary IPv4 addresses of the Teredo server while on the IPv6 side no special setting was needed.

Miredo makes possible to run Teredo server (i.e., `miredo-server`) and Teredo relay (i.e., `miredo`) instances on the same host. Therefore the relay role was also played by

the GGSN and `miredo.conf` was used for specifying the relay type. We applied `RelayType restricted` for our measurements. The relay took care of adding required Teredo IPv6 routing and addressing on the host. However, "non-Teredo" IPv6 addressing/routing requires manual configuration or usage of dynamic routing.

As in all of our measurement setups, a separated NTP network for UE and CN time synchronization was also applied here.

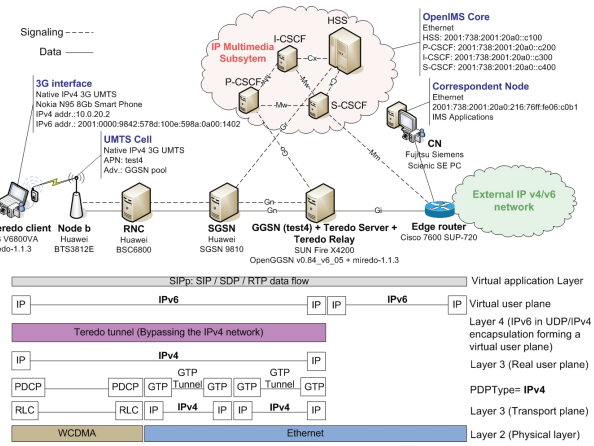


Figure 10. Architecture of Teredo tunneling in our 3G UMTS/IMS testbed

8) *NAT-PT*

This measurement scenario is to evaluate NAT-PT, which is the most widespread translation-based v4-v6 transition scheme transparently applicable for User Equipments. Our analysis and testbed setup was built upon the NAT-PT implementation called `naptd` [40]. The `naptd` software loosely implements RFC2766 [9] in user space, runs on GNU/Linux operating systems and makes possible to easily setup and configure Network Address Translation - Protocol Translation between IPv6 (as internal) and IPv4 (as external) networks. It was designed to effectively utilize available system resources such to run even on low-end hardware with only one network interface card installed. According to the recommendations, `naptd` uses Address Resolution Protocol (ARP) on the IPv4 and Neighbor Discovery (ND) on the IPv6 network segments while also participates in dynamic routing for both IPv4 and IPv6 if needed.

Usually, NAT-PT implementations cannot translate IP address and subsidiary information carried inside packet payloads. However, some protocols (e.g., DNS, FTP or SIP) require such intervention for proper translation between IP versions. This issue is also solved in `naptd` as different Application Level Gateways (ALGs) are implemented by loadable plugins of the main module.

We applied `naptd` version 0.4 (`naptd-0.4`) in our testbed with some minor modifications to the software's default usage scenario and ALG support: we made it possible to measure the translation use-case between internal IPv4 and external IPv6 networks, and introduced a simple way to provide SIP ALG operation for supporting IMS applications and services embodied by our `SIPp` scripts. This slightly modified naptd-0.4 architecture was installed and configured in our 3G UMTS testing environment (Fig. 11) by giving the roles of the NAT device and ALG functions to the GGSN (i.e., the boundary router situated between the IPv4 and IPv6 network segments). Besides the setup of the dedicated "shadow" NTP network, UEs and CNs did not require additional configuration or modification of their basic software environment in this scenario.

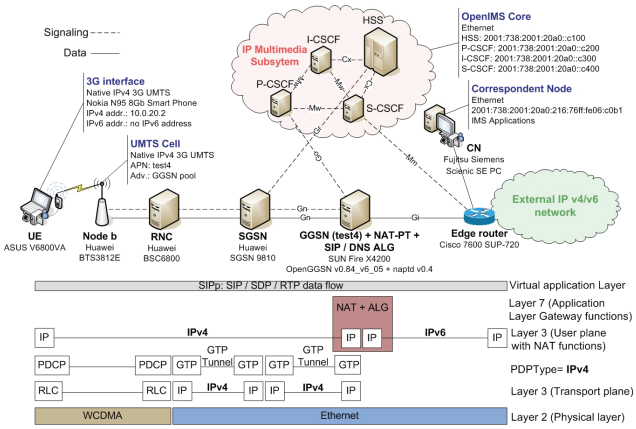


Figure 11. NAT-PT-based v4-v6 transition in our 3G UMTS/IMS testbed

V. Performance Results

This section presents the results of our efforts to evaluate the performance of key IMS operations over the above introduced eight scenarios of 3G UMTS access: native IPv4, native IPv6, L2TP, OpenVPN UDP/TCP, 6to4, ISATAP, Teredo, and NAT-PT. The outcomes are presented in boxplots (i.e., box-and-whisker diagrams) to depict the collected numerical data groups through their six-number summaries. The used six representatives are as follows: the lowest sample value (lower line), the lower quartile called Q1 (the lower edge of the box), the mid-quartile or median called Q2 (the delimiter of the two distinctive colors of the box), the upper or third-quartile called Q3 (the upper edge of the box), the largest sample value (the upper line), and the mean of the collected data (red colored rhombus). In our diagrams the Q1-Q2 interval is indicated by grey color and the Q2-Q3 interval is colored with light blue. The height of boxes (i.e., the interquartile range) represents the middle fifty percent of the measured data.

The test cycles for every performance parameter comprised a total of 1000 measured events in every scenario: 1000 RTTs, 1000 IMS Registrations, 1000 Call Setups, and 1000 RTP transmissions, respectively.

The main motivation behind our measurements was to compare native IPv6 3G UMTS network performance with native IPv4, tunneled IPv6 solutions and the most widespread translation-based solution, using key parameters of IMS operations as performance metrics. The comparison based on the access modes (native IPv4, native IPv6, L2TP, OpenVPN UDP/TCP, 6to4, ISATAP, Teredo, NAT-PT) of all the examined v4-v6 transition scenarios revealed an explicit order, which is noticeable almost in all cases. The analyzed key IMS performance metrics show that the fastest

solutions are the native IPv4 and native IPv6 access modes as expected, and these are followed by the L2TP, 6to4, ISATAP, Teredo, OpenVPN UDP, NAT-PT and the OpenVPN TCP-based IPv6 solutions.

The measurements regarding the native scenarios (IPv4 and IPv6) reveal a slight advantage of IPv4, especially obvious if considering the IMS Registration Time. However, in some cases IPv6 outperforms IPv4 according to the mean values (e.g., Call Setup Time). Although this was sparse it must be mentioned that IPv6 was always very close to IPv4, and particularly observing the deviation we can say that IPv6 showed quite a well balanced performance. The advantage of native IPv4 at the most of the evaluated IMS metrics could be explained with the smaller address space, and it is also a significant fact in this matter, that IPv4 is a full-fledged protocol – it has been developed for nearly forty years – while the IPv6 protocol stack implemented in the present devices is yet likely to face with some performance issues due to its immature nature.

As also expected the tunneling-based access methods have the worst performance compared to the native solutions in all the scenarios and at all key IMS metrics. The explanation can be found in the general nature of tunneling mechanisms, in the characteristics of the used transport protocols and in the implementations. RP-L2TP uses only packet encapsulation over UDP without any encryption to transmit packets between the two end points of the tunnel, and that simplicity added to the session layer operation made possible to get close to the network (6to4, ISATAP) and transport-level (Teredo) tunneling schemes and to beat application-level (OpenVPN) tunneling solutions together with the also evaluated application-level translation-based method (NAT-PT). The operation of 6to4 and ISATAP requires a lower encapsulation overhead compared to L2TP, Teredo and OpenVPN that both apply UDP/IP or TCP/IP encapsulation. OpenVPN builds up tunnels in an encrypted way using the OpenSSL/TLS library by default thus the tunnel endpoints require more time and resources to process the packet encapsulations and decapsulations. In addition, if the solution uses TCP instead of UDP, the OpenVPN implementation expects acknowledgements after sent packets causing more delay and significant deviation among measurement data. It is generally noticeable that choosing more and more complex mechanisms will cause larger response time and thus worse performance. That is also the main reason of the outcomes of our NAT-PT measurements, which show that translation between different IP versions with application-layer gateway support can provide results only barely better than the most resource consuming OpenVPN TCP solution. However, NAT-PT does not require intervention in UE softwares, which could make the deployment of this transition scheme really fast.

Depending on the observations two main conclusions can be stated. The first one is that nowadays native IPv6 is almost as fast as native IPv4 and in some circumstances it can even outperform its predecessor, although yet IPv6 is an immature protocol and further improvements are expected in the near future. However, no serious deducible difference can be observed between the analyzed IPv4 and IPv6 protocol stacks.

According to the second statement we can say that it is highly recommended to use native IPv6 instead of tunneling protocols in 3G UMTS and beyond, because currently available tunneling methods are much slower and worst balanced than their native counterpart. However we cannot determine significant differences between L2TP, 6to4 and ISATAP, we can say that these above tunneling methods outperform Teredo, OpenVPN UDP/TCP and even NAT-PT in most of the measurement scenarios.

IPv6 will provide enough IP addresses for every piece of device also in an “Internet of Things” era, and the native accommodation of the next generation Internet Protocol also will remarkably simplify the network architecture of mobile and wireless communication systems. However, IPv6 in mobile and wireless networks can not appear in one night, IPv6 will not suddenly provide global coverage. This implies that tunneling-based, translation-centric or other kind of transition techniques need special care and explicit attention, despite the fact that they perform worse and show significant overhead compared to the native cases.

VI. Conclusion and Future Work

The research presented in this paper mainly concerned the questions and challenges of the transition from IPv4 to IPv6 in all-IP 3G and beyond multimedia systems, and their impacts on the performance of IMS services and applications. In order to quantify the effects of different methods providing IPv6 support/transition techniques in existing mobile telecommunication architectures, we designed and implemented a 3G UMTS testbed (including the IMS core) and compared the performance characteristics of several selected transition techniques (L2TP, OpenVPN UDP, OpenVPN TCP, 6to4, ISATAP, Teredo, NAT-PT) with native IPv4 and IPv6 scenarios using key IMS operations as performance metrics. Our results exposed the main benefits and drawbacks of the examined technologies based on their actually available implementations, and highlighted some strict limitations concerning the non-native IPv6 support so we must stress the need for further studies aiming to help and urge the process towards the global native IPv6 coverage.

As a part of our future work we are planning to extend the evaluation of heterogeneous scenarios (i.e., v4 caller communicates with a v6 callee and vice versa) using other translation-based transition mechanisms (e.g., BIS, BIA, TRT, SOCKS64), application layer gateways and proxies. We are also devoted to analyze some yet missed tunneling mechanisms (6over4, DSTM, Proto41, AYIYA/AICCU etc.). We also would like to extend our experimental approach with extensive and detailed overhead measurements of different IPv4/IPv6 transition techniques.

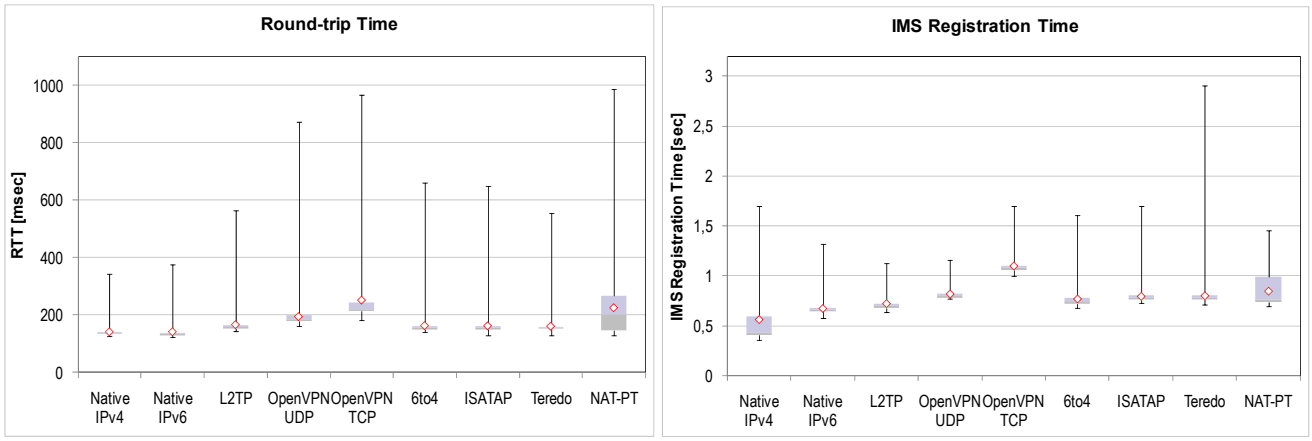


Figure 12. Round-trip Time and IMS Registration Time

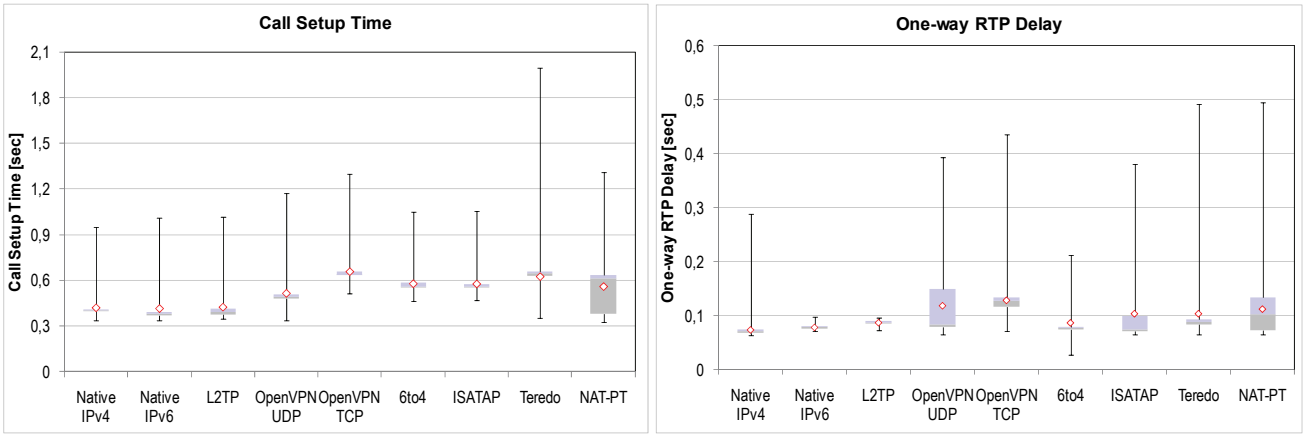


Figure 13. Call Setup Time and One-way RTP Delay

ACKNOWLEDGMENT

This work was supported by the Hungarian Government through the TÁMOP-4.2.1/B-09/1/KMR-2010-0002 project at the Budapest University of Technology and Economics together with the help of Mobile Innovation Centre Hungary and the IST-ANEMONE project, which was partly funded by the Sixth Framework Programme of the European Commission's Information Society Technology. The authors also would like to express their appreciation to László Madarassy and Szabolcs Kustos for their essential work on this research.

REFERENCES

[1] L. Bokor, Z. Kanizsai, and G. Jeney, "Performance Evaluation of Key IMS Operations over IPv6-capable 3G UMTS Networks", In proc. of the Ninth International Conference on Networks (ICN 2010), pp. 1-10, ISBN: 978-0-7695-3979-9, DOI: DOI:10.1109/ICN.2010.49, Les Menuires, France, April 2010.

[2] S. Deering and R. Hinden, "Internet Protocol version 6 (IPv6): Specifications," IETF RFC 2460, Dec. 1998.

[3] A. Cuevas, J.I. Moreno, P. Vidales, and H. Einsiedler, "The IMS Service Platform: A Solution for Next-Generation Network Operators to Be More than Bit Pipes", *IEEE Com.M.* V.44, N.8, pp.75-81, 2006.

[4] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol", IETF RFC 3261, June 2002.

[5] 3GPP TS 23.060 : "General Packet Radio Service (GPRS); Service description", Stage 2, (Release 9), V9.2.0, September 2009.

[6] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS)", Stage 2, (Release 9), V9.2.0, December 2009.

[7] B. Carpenter and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", IETF RFC 3056, February 2001.

[8] F. Templin, T. Gleeson, M. Talwar, and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", IETF RFC 4214, October 2005.

[9] G. Tsirtsis and P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)", IETF RFC 2766, February 2000.

[10] J. Hagino and K. Yamamoto, "An IPv6-to-IPv4 Transport Relay Translator", IETF RFC 3142, June 2001.

[11] P. Srisuresh, G. Tsirtsis, P. Akkiraju, and A. Heffernan, "DNS extensions to Network Address Translators (DNS_ALG)", IETF RFC 2694, September 1999.

[12] S. Thomson, C. Huitema, V. Ksinant, and M. Souissi, "DNS Extensions to Support IP Version 6", IETF RFC 3596, October 2003.

[13] 3GPP TS 29.061: "Interworking between the Public Land Mobile Network (PLMN) supporting packet based services and Packet Data Networks (PDN)", (Release 9), V9.4.0, September 2010.

[14] J. Lau, Ed., M. Townsley, Ed., and I. Goyret, Ed., "Layer Two Tunneling Protocol - Version 3 ", IETF RFC 3931, March 2005

[15] W. Simpson, Ed., "The Point-to-Point Protocol (PPP)", IETF RFC 1661, July 1994

[16] OpenVPN Technologies, http://www.openvpn.net [Accessed: Jan. 13, 2011]

[17] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", IETF RFC 5246, August 2008.

[18] C. Huitema, "An Anycast Prefix for 6to4 Relay Routers", IETF RFC 3068, June 2001.

[19] Z. Xiaodong, M. Yan, and Z. Yumei, "Research on the Next-Generation Internet Transition Technology", In proc. of the Second International Symposium on Computational Intelligence and Design (ISCID '09), pp. 380-382, Changsha, China, 2009.

[20] S.D. Lee, M.K. Shin, and H.J. Kim, "The implementation of ISATAP router", In proc of the 8th International Conference on Advanced Communication Technology (ICACT'06), pp. 1163, Phoenix Park, Republic of Korea, 2006.

[21] Y. Hei and K. Yamazaki, "Traffic analysis and worldwide operation of open 6to4 relays for IPv6 deployment", In proc. of International Symposium on Applications and the Internet, pp. 265-268, 2004.

[22] C. Huitema, Teredo: "Tunneling IPv6 over UDP hrough Network Address Translations (NATs)", IETF RFC 4380, February 2006.

[23] S.M. Huang, Q. Wu, and Y.B. Lin, "Enhancing Teredo IPv6 tunneling to traverse the symmetric NAT", IEEE Communication Letters , 10 (5), 408-410, 2006.

[24] G. Tsirtsis and P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)", IETF RFC 2766, February 2000.

[25] E. Nordmark, "Stateless IP/ICMP Translation Algorithm (SIIT)", IETF RFC 2765, February 2000.

[26] C. Aoun and E. Davies, "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status", IETF RFC 4966, July 2007.

[27] SIPp test tool and traffic generator, http://sipp.sourceforge.net [Accessed: Jan. 13, 2011]

[28] 3GPP TS 24.228: "Signalling flows for the IP multimedia call control based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP)", Stage 3, (Release 5), V5.15.0, September 2006

[29] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", IETF RFC 3550, July 2003

[30] Tshark: The terminal oriented version of Wireshark network protocol analyzer, http://www.wireshark.org [Accessed: Jan. 13, 2011]

[31] Budapest University of Technology and Economics - Mobile Innovation Centre (MIK), http://www.mik.bme.hu [Accessed: Jan. 13, 2011]

[32] OpenGGSN on SourceForge: http://sourceforge.net/projects/ggsn [Accessed: Jan. 13, 2011]

[33] L. Bokor, Z. Kanizsai, and G. Jeney, "Setting up native IPv6 UMTS access with open-source GGSN implementation" [Online] Available: http://www.ist-anemone.eu/index.php/Setting_up_native_IPv6_UMTS_access_with_open-source_GGSN_implementation [Accessed: Jan. 13, 2011]

[34] Fraunhofer FOKUS NGNI, OpenIMSCore Project, Official website: http://www.openimscore.org [Accessed: Jan. 13, 2011]

[35] Linux IPv6 Router Advertisement Daemon (radvd), http://www.litech.org/radvd [Accessed: Jan. 13, 2011]

[36] Roaring Penguin L2TP, http://rp-l2tp.sourceforge.net [Accessed: Jan. 13, 2011]

[37] P. Bieringer, "Configuring 6to4 tunnels", Linux IPv6 HOWTO on TLDP, http://tldp.org/HOWTO/Linux+IPv6-HOWTO/configuring-ipv6to4-tunnels.html [Accessed: Jan. 13, 2011]

[38] S. Hlusiak, "*isatapd*–ISATAP client for Linux", http://www.saschahlusiak.de/linux/isatap.htm [Accessed: Jan. 13, 2011]

[39] R. Denis-Courmont, "Miredo : Teredo IPv6 tunneling for Linux and BSD", http://www.remlab.net/miredo [Accessed: Jan. 13, 2011]

[40] L. Tomicki, "Network Address Translation, Protocol Translation IPv4/IPv6", http://tomicki.net/naptd.php [Accessed: Jan. 13, 2011]

Simulation and Analysis of a QoS multipath Routing Protocol for Smart Electricity Networks

Agustin Zaballos, Alex Vallejo, Guillermo Ravera and Josep Maria Selga
Computer Science Department
Enginyeria i Arquitectura La Salle – URL (University Ramon Llull)
Barcelona, Spain
{zaballos, avallejo, gjravera, jmselga}@salle.url.edu

Abstract— **This paper presents several considerations that must be taken into account in the design of a QoS multipath routing protocol suitable for Smart Electricity Networks (SENs). The main goal is to analyze the routing requirements that will facilitate the future formal specification of a QoS-aware multipath routing algorithm for the coming SEN's data networks in the high voltage segment where long-distance mesh data networks will be a future challenge for engineering and research. In this paper, the study is focused on requirement specification regarding distance-vector routing algorithms that reduce the overall overhead of routing information needed in the network, thus preserving the scalability feature. In order to obtain a detailed study of the performance of these protocols, the proposal "Distributed path computation with Intermediate Variables" and several proposed improvements have been modeled using OPNET modeler in an aim to evaluate the performance of this protocol in different SEN domain-related situations.**

Keywords- Multipath routing; Quality of Service; Smart Electricity Networks.

I. INTRODUCTION

The future of the utilities walks hand in hand with Smart Electricity Networks (SENs) and its advantages [1][2]. SENs saves energy and can cope better with the unpredictable supply from renewable energies [3][4]. Actually, utilities require to be prepared to face the upgraded needs of its telecommunications infrastructure. Although it is known that this is a long term process and the prediction is that smart grids will be implanted in 2030.

One of the main challenges of SENs is to redesign its network architecture. Nowadays, a utility grid has been deployed according to a centralized scheme in which the different elements of the grid are logically and geographically located. This fact is due to the one-way power flow from the generation to dispersed loads. Current grid scheme is easy to operate but the SEN has an opposite point of view. The architecture is based on a decentralized scheme with elements logically identified but not geographically located. For example, the increment in renewable generators in the customer's premises will change the way the energy is generated. All customers will be able to generate energy, to consume it and finally to give the remaining part to the SEN. Because of this, communications must be upgraded and meet new goals [3][4] such as the necessity of several application protocols, the specification of new data models needed by the applications, the exploitation of the currently deployed communications infrastructure and the adoption of robust and QoS-aware routing protocols to satisfy the requirements of the network.

This paper presents and analyses a thorough study of the characteristics that must be considered in a QoS multipath distance-vector routing protocol, in the way to get a protocol that could be implemented in a SEN. In [1], authors have studied the issues on the design of a multipath protocol that could be implemented in a real smart grid. The first task of a QoS multipath routing protocol is to find several suitable loop-free paths from the source to the destination with the necessary available resources to meet the QoS requirements of the desired service. It has to take advantage of the topology of the utility network (partial mesh) by making the network resilient to failures. Moreover, the few requirements of bandwidth are another advantage of these protocols which allow to use the available bandwidth efficiently. There are a lot of implementations of multipath routing, for example: DASM [5], MDVA [6], MPDA [7], MPATH [8] or DIV [9]. These kind of protocols obtain the maximum redundancy of the network finding more than one path to a destination. All these protocols define the behavior of the algorithm but let undefined important aspects such as the routing metric or the load balancing method. These aspects must be defined in the final implementation of a routing protocol to operate properly.

The remainder of this paper is organized as follows: Section II describes the fundamental topics involved with the domain of Smart Electricity Networks. Characteristics and QoS requirements of SENs are presented and the need for a better communication architecture is also explained. Section III briefly details the proposed network model suitable for further formal specification of a QoS-aware multipath routing algorithm for SENs. Section IV introduces the fundamental topics involved in our analysis needed to understand the principles of multipath routing. Section V discusses the routing considerations and covers all the important design issues. Section VI presents the second part of the study with the implementation of a routing protocol based on the theoretical study done in [1]. This implementation has been done over the OPNET MODELER 12.0 simulator [10] and it allows to compare the results of some aspects defined in the theoretical study. Finally, in section VII, conclusions are outlined.

II. SMART ELECTRICITY NETWORK'S COMMUNICATION REQUIREMENTS

Current power grid is defined as a system made up of electrical generators, transformers, transmission and distribution lines used for delivering electricity power to final users. Monitoring and smart grid network control are very important features in order to provide continuity, QoS and security. Nevertheless, at the time, most of these functions are only carried out in high voltage and, sometimes, in the medium voltage grid. The future SEN must be distinguished by self-healing and automation taking into account that should support thousands of clients and all the energy providers. Actually, international organizations, governments, utilities and standardization organisms are becoming aware that the grid needs a modernization.

Many companies which belong to different sectors have seen a great business opportunity and are currently working to make themselves room in the SEN's market. The change towards the so-called Smart Grid or SEN promises to be a change in the whole business model involving utilities, regulation entities, service providers, technology suppliers and electricity consumers. In fact, this transformation towards an intelligent network is possible by importing the philosophy, concepts and technologies from the Internet ambit.

Nowadays, utility grid is used to transport energy from generators to end users. Currently, in most countries the grid is old and has several problems of inefficiencies and low robustness due to the lack of automation [11]. The grid could be improved to overcome these deficiencies by coordinating processes between Intelligent Electric Devices (IEDs). Thus, well-known problems such as those described in [11] could be avoided. SENs will manage lots of real-time information through a data network and they will collect information from established IEDs for the purpose of control. This kind of data networks is not exempt from the growing need of Quality of Service (QoS). SENs are expected to meet a drastic increasing demand of information, communication and miscellaneous data such as voice, data, image, video and multimedia communications, which can be accessed anywhere at any time.

SENs need to communicate many different types of devices, with different needs for QoS over different physical media. Availability is also crucial for the correct operation of the network. The elements of a SEN, the so-called IEDs, can have very different QoS necessities. For example, real-time communications are required in the case of fault detection, service restoration or quality monitoring; periodic communications are used in Automatic Meter Reading systems (AMR); bulk data transfers are useful to read logs and energy quality information. SENs would not be possible without the existence of the IEDs which can play as sensors and/or actuators. There exist many types of IEDs depending on the function carried out. The IEDs involved in these functions can be situated in different locations due to the pursued decentralized architecture. For example, electrical substation elements are connected to the substation's Ethernet network; sensors can be installed along electrical cables communicated through wireless standards. Communication from the control center to energy meters and between substations can be carried out via a high variety of technologies such as Power Line Communication (PLC) or WiMAX.

Due to these circumstances, SEN will be supported by data network with strict constraints of QoS. Therefore, one of the most important needed specifications for the SEN are those regarding its communications. A framework for management of end-to-end QoS for all communications in the grid will be a must in the future [12][13]. In fact, a suitable communications infrastructure allows increasing the efficiency of the electric system further than what is possible with automation without communication capacities.

Furthermore, automation of distributed generation requires new protections and supersedes the actual one way generation flow. The control and monitoring of these new flows could not be done with the same scheme used in the past, but can be done with the aid of a new and more flexible communication network.

Some of the new communication goals faced by SENs are listed below:

- New application protocols will be needed to meet the new network requirements.
- New data models will be required by the applications.
- It is essential to take full advantage of the communication infrastructure deployed.
- To adopt robust and quality aware protocols to satisfy the restrictive QoS requirements of the network.

III. NETWORK MODEL AND NOTATION

In this section, the network model and the used notation for the routing algebra and policies are described. This notation is used to formally define the routing protocol behavior and it is based on Sobrinho's routing algebra [14][15]. An algebraic approach is very useful for both understanding existing protocols and for exploring the design space of future Internet routing protocols.

A. Routing policy

The routing policy defines the elements used by the routing protocol to carry out the routing process. The routing policy is formed by [14][15]:

$$A = < \Sigma, \oplus, L, \preccurlyeq > \quad (1)$$

Each element of this array (1) is defined in Table I. In addition to this, two logical operators are necessary: AND ($\wedge$) and OR ($\vee$) operators. In this paper, the following model of a cost computation based on [15] is used. It is outlined in Fig.1, where node *j* is the destination of the routing information and node *v* is the origin.

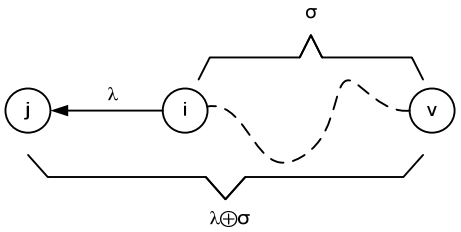


Figure 1. Example of the routing algebra where $(\lambda \in L) \wedge (\sigma \in \Sigma)$

TABLE I. ELEMENTS OF THE ROUTING POLICY

Element	Description
Σ	It is the cost associated to a path and it is known as the signature.
$\oplus$	It defines the way to add the cost of a link to a path and to calculate the total cost. It is known as the operator.
L	It represents the cost associated to a link and it is known as the label of the link.
$\preccurlyeq$	It is the precedence relationship and it is used to decide which path is the best one.

B. Representation of the routing information

In this section, it is described the proposed notation in order to define the information used and stored by the routing protocol. The objective is to avoid any confusion when different routing schemes and metrics will be explained. Actually, a routing protocol is an algebra together with a distribution mechanism (such as link-state or vector-distance) for computing routing solutions.

- i: It represents the origin node.
- k: It represents a neighbor of node i, which has sent a routing advertisement to node i ($k \in N_i$).
- j: It represents the advertised destination of the routing information received.
- λ_{ik}: It is the cost of the link from node i to node k.
- σ_{kj}: It is the cost of the path from node k to node j advertised by k.
- $\hat{\sigma}^i_{kj}$: It stands for the estimated cost of the path from node k to node j stored on the routing table of node i.
- $\hat{\sigma}^i_{ikj}$: It stands for the estimated cost of the path from node i to node j through the neighbor k stored in the routing table of node i.
- $\hat{\sigma}^k_{ij}$: It is the cost estimated of the path from node i to node j that node i guesses that is known by node k.
- S_{ij}: It is a set of all the neighbor nodes of the node i that are feasible successors to node j.
- N_i: It is a set of all the neighbor nodes of the node i.

A node can store two cost values from its neighbors: the actual values and the estimated values. Estimated values are the information received from the neighbors that can be potentially outdated due to network changes which have not been notified yet as the routing protocol has not converged.

IV. MULTIPATH ROUTING IN SENS

The number of nodes in a SEN can range from a few hundred to thousands depending on the deployment. SENs with thousands of nodes can become common in the future. Therefore, the routing algorithm needs to be distributed, decentralized and scalable.

Multipath routing can use disjoint paths or non disjoint paths. The problem with disjoint path is the dependence on the physical topology of the network and the difficulty in being allocated by the routing algorithm because any of the paths to a certain destination node can share any link [16]. For this reason, our study in this paper focuses on non disjoint path distance-vector multipath routing protocols.

Multipath routing let obtain some benefits for the network performance, such as: reduction of the average delay [7], more security against network attacks [8] or reduction in the overall convergence time when a link fails. The latter benefit introduces two extra advantages, the reduction of the communication overhead in the network and a convergence time close to 0 s. in some cases. On the other hand, the routing loops can exist with higher probability than in shortest-path routing.

A. Mechanisms for successors selection

Multipath routing algorithms maintain a group of vectors to the same destination called successors (S_{ij}). These neighbor nodes are used to route the packets to a certain destination. The most difficult problem that must be solved is the selection of these successors avoiding loops in the network. This problem has different solutions:

1) *Using loop-free paths with the same cost:*

This is the simplest multipath mechanism based on selecting the paths with the same cost than the shortest path to a certain destination. An example of a protocol that uses this type of multipath is OSPF.

$$S_{ij} = \left\{ k \middle| \hat{\sigma}^i_{ikj} = \min \sigma_{ij} \ , \forall \ k \in N_i \right\} \qquad (2)$$

The main advantage is the simplicity because it does not require modifying the baseline protocol behavior but it is pretty unlikely to find paths with the same cost.

2) *Using loop-free paths with a variation over the minimum:*

This mechanism is used, for example, by EIGRP. It is based on selecting the paths with a variance of γ in the shortest path.

$$S_{ij} = \left\{ k \middle| \hat{\sigma}^i_{ikj} < \left(\min \sigma_{ij} \cdot \gamma\right), \forall \ k \in N_i \right\} \mid \gamma > 1 \qquad (3)$$

The main disadvantages are that the probability of using a path with a loop depends on γ parameter and the inability to avoid bouncing effects due to a bad configuration of this parameter.

3) *Using loop-free paths in a pseudostationary network:*

This mechanism is based on the Loop Free Invariance (LFI) conditions [17] with the difference that multipath routing pursues a group of successors greater than one. The condition that must be satisfied in order to avoid loops is the following:

$$S_{ij} = \{k | \hat{\sigma}^i_{kj} < \sigma_{ij}, \forall\, k \in N_i\} \mid \sigma_{ij} = \min(\lambda_{iv} \oplus \hat{\sigma}^i_{vj}), \forall\, v \in N_i \quad (4)$$

If this condition is satisfied, it can be affirmed that there are no loops in the network when the routing protocol has converged. But during the convergence of the network it is possible that a loop is originated. To satisfy this condition even when the network is converging, it is necessary a synchronization mechanism.

4) *Using loop-free paths with second-to-last-hop information:*

This mechanism shares the information of the penultimate hop. With this information, the routing protocol knows all hops in the path and can calculate whether a new link added to a path is originating a loop or not. MPATH [8] is an example of this type of multipath routing protocol. Although this mechanism is robust, it also needs a synchronization mechanism to maintain the information updated on all nodes in the network.

B. Synchronization mechanism

LFIs can be used to avoid loops in a converged network but not during convergence time. The reason is the distributed nature of the distance-vector algorithms. This nature does not assure that the LFIs are accomplished by all the nodes in the network because each one maintains his own routing information and can make wrong decisions based on outdated routing information. For this reason, it is necessary a mechanism to state that the information is correct in all nodes of the network. This mechanism must update the routing information within a finite period of time and must have a beginning and an end. Moreover, it is necessary to bear in mind that only the metric's increasing needs to be synchronized because it is critical. A decrease in the metric cannot create a loop.

There are different methods of synchronization but most of them are based on the diffusing computations studied by Dijkstra [19]. The first multipath algorithm incorporating the diffusing computations was DASM [5], which uses a variant of the LFI used in the EIGRP's Diffusing-Update Algorithm (DUAL). This variant only initiates the synchronization when the Loop-free Routing Condition (LRC) is violated. LRC is a relaxed condition of the LFI and it reduces convergence time of DUAL and also the number of routing messages needed. The evolution of DASM is MDVA [6] which accelerates the diffusing computations.

The main problem of diffusing computations is the overhead and the delay needed in end to end synchronization. That is the reason why the synchronization proposed in MPDA [7] and MPATH [8] is only for a one hop scope. The advantage of this synchronization is that the originator of the update can receive an acknowledgement faster. DIV routing protocol [9] has been recently defined. This LFI-based protocol provides normal and alternate synchronization. Normal synchronization is similar to MDVA synchronization and alternate is similar to MPATH or MPDA synchronization. A difference in DIV is the treatment of a decrease in the cost metric. DIV produces a message that supersedes any increment process to the same destination and, thus, it can stop a diffusing computation that is taking place in the network. This difference, combined with the two possible synchronization mechanisms, gives more flexibility to the protocol. For these reasons, this protocol could be chosen for its use in SEN data networks.

C. Constraint- based protocols

The routing protocols based on constraints find one or more paths that satisfy a subset of QoS conditions imposed by the user. It is also necessary to define the QoS constraints scope. They can be applied to a single link or to the whole path [18]. The first problem solved was finding a path using two constraints, which is a NP-complete problem [18][20]. The problem with two constraints can be generalized to z constraints. MPOR routing protocol [20] is able to solve the Multiple Constraint Problem (MCP) over a group of z restrictions, furthermore this protocol uses an optimization function to select the path. The method applied to solve the problem is the limited path heuristic proposed in [21]. In order to avoid the loops, the algorithm uses LFIs with the synchronization mechanism proposed in DUAL. The disadvantage of this type of algorithms is the dependency on global parameters to solve the routing problems, thus limiting the possibilities of the algorithm. Moreover, the heuristic solution can fail to find the path even when this path exists.

V. CONSIDERATIONS IN THE DESIGN FOR A QOS MULTIPATH ROUTING PROTOCOL FOR SENS

The study from the different multipath algorithms, such as DASM, MDVA, MPDA, MPATH, MPOR or DIV among others has concluded that the most robust way to solve the multipath routing problem is applying a scheme based on the diffusing computations and use LFIs to avoid loops. Even though our initial studies were focused on solving the routing loops and bouncing problems using techniques such as split horizon, split horizon with poison reverse, triggered updates, hold down timers, the use of the second-to-last-hop information or legacy synchronization methods.

On the other hand, there are other aspects that must be defined to obtain an algorithm that could be implemented in a real SEN. Some of these aspects are: the detection of neighbors, the hierarchy of the network, the definition of which synchronization mechanism is used, the addressable elements in the network or the address scheme used by the protocol to identify the nodes in the network. Furthermore, if the protocol is oriented to provide QoS, additional aspects have to be defined, such as the QoS metric, the specification of the protocol to minimize the amount of bandwidth needed and the load balancing scheme. These aspects are treated in each of the following parts.

A. Neighbor detection

The neighbor detection has two primary objectives in a routing protocol: establish neighbor adjacencies and detect the failure/recovery of a direct connected link.

The distance-vector routing protocols do not need to establish adjacencies between neighbors, but the failure/recovery of a link is critical because it can force the synchronization of the routing algorithm which is crucial to

maintain the stability of the network. In case of synchronization need, the most critical event is the increment in the metric because it can create a loop in the network. If the failure of a link is not detected quickly, all the packets routed across the failed link will be discarded decreasing the performance of a SEN. A failure in the detection could compromise the detection of the end of a diffusing computation and it is known as Stuck In Active process or SIA. The cause of a failure in the neighbor detection could be the physical medium. For example, Ethernet or PLC is a shared medium and it is not always possible to detect a neighbor failure if a switch or bridge is in the middle. This behavior is likely to be usual in the future SENs' PLC and wireless hybrid networks.

The solution to that problem is to implement a keep-alive mechanism controlled by the routing protocol, consequently incrementing the message overhead of the routing protocol. On the other hand, the possibility of using BFD protocol [22] could be very interesting for our purposes. BFD, which is a much more efficient solution, establishes a connection with neighbors through a three way-handshake and it monitors them with hello messages. In case of failure, it is immediately notified to the routing protocol.

B. Hierarchy of the network

The network hierarchy is the distribution in areas of the nodes in the network. Some examples of hierarchic protocols are OSPF or IS-IS which are link-state protocols. Hierarchy mechanism is applied to solve the scalability problem of the link-state protocols, although it can be used on distance-vector to reduce the amount of traffic inside an area.

The information maintained by a node in the network is updated depending on the number of changes in the network. The routing table size of flat networks increases linearly by limiting the scalability of the protocol due to the amount of information that is needed to be shared. When the routing protocol uses hierarchy techniques, a group of nodes are treated as a unique addressable entity from the top of the hierarchy (e.g., OSPF areas). An example of hierarchic routing protocol is HIPR [23] which is based on the Loop-free Path-finding Algorithm (LPA). The three most important advantages derived from this mechanism are the reduction in the number of routing messages needed to converge, the reduction of the size on the route table (reducing the memory needed by the node) and also the reduction of the convergence time.

When a hierarchic protocol is used, it is necessary to define the hierarchy scheme used because it settles for the direction of the information flow. For example, OSPF forces all areas to be in contact with the backbone area by addressing all the traffic through it. On the other hand, it is the ALVA algorithm [24], which implements a more flexible scheme.

C. Algorithms based on sequence numbers

The sequence numbers are introduced in the routing messages in order to allow the routing protocol to identify the updates. For example, DIV numbers the increment updates for identifying to which diffusing computation belongs to. This number also let identify out of order messages or detect duplicated messages.

An important issue that must be taken into account when sequence numbers are used is that they are part of a finite group of numbers. For this reason, it would be necessary to define the maximum possible value and the mechanism to synchronize them when that maximum is reached. Another aspect that must be defined is the scope of a sequence number, it can identify a local sequence number between two neighbors (this is the case of DIV) or it can be global to the entire network (this is the case of AODV [25]). The AODV's option is the most difficult to synchronize because the sequence number must be synchronized across the entire network [26]. The last topic that must be defined is the behavior of a new node in the network because it would not be synchronized with the network. It is related with the neighbor detection mechanism that must also synchronize new nodes reached in the network.

D. QoS metrics

Several routing protocols are oriented to provide QoS. All of them provide the QoS based on the metric used by the routing protocol, which give much more information than the number of hops. This is the reason why the metric chosen is an important parameter when a routing protocol is designed. The metric is the value used to select which path is the best one. Moreover, it is necessary to define the optimization functions which define the objective of the routing policy.

The metrics could be subdivided into two flavors: static and dynamic. Static metrics represent a stable vision of the network. The variation of these metrics is usually caused by disconnected links. On the other side, by using dynamic metrics, the stability of the network could be compromised depending on how frequently the metric is updated. Its main problem is that they can change frequently, making necessary to apply hysteresis cycles or the average of the metric value. In order to promote network stability, such metrics have been dismissed for SEN networks in this paper. A metric classification is depicted below.

1) *Single metrics*

These are the simplest ones and they only represent a single characteristic of a path. Examples of routing protocols that use these metrics are RIP or IS-IS, using the number of hops as a path metric. Another useful metric example for SENs is the Bandwidth-inversion Shortest Path (BSP) [27] or the Enhanced Bandwidth-inversion Shortest Path (EBSP) [28] with better performance in heterogeneous network (see Table II and Table III).

2) *Combined metrics*

These metrics also represent a characteristic of a link using one value. But this value is obtained from a combination of metrics by avoiding NP-complete problem [13]. The metric proposed in [29] is an example of a metric with a combination of bandwidth, delay and reliability of the link (Table IV).

3) *Multiple metrics*

This metric scheme represents a link with more than one cost value. The entire network exchanges this information and then the node could combine this information in order to

decide the best path to the destination. The most known examples of this type of metrics are de Widest Shortest Path (WSP) [30] and the Shortest Widest Path (SWP) [29]. Other examples are [31] and [32]. The former divides bandwidth between numbers of hops and the latter divides delay between bandwidth. The main difference between them is how they prioritize the importance among all metrics to obtain the preferred path. Another example is the metric proposed on [33], whose metric scheme uses the number of hops, link bandwidth and total bandwidth of the path. It establishes a hierarchy to evaluate all the metrics, first it is evaluated the number of hops; if both paths have the same metric, the minimum bandwidth is evaluated and so on. It is known as lexicographic order [14] (Table V).

TABLE II. BSP METRIC

Σ	$\sigma \in R^+$
$\oplus$	$\lambda \oplus \sigma = \lambda + \sigma$
L	$\lambda \in R^+, \lambda = \frac{1}{BW}$
$\preccurlyeq$	$\leq$

TABLE III. EBSP METRIC

Σ	$\sigma \in R^+$
$\oplus$	$\lambda \oplus \sigma = \lambda + 2 \cdot \sigma$
L	$\lambda \in R^+, \lambda = \frac{1}{BW}$
$\preccurlyeq$	$\leq$

TABLE IV. EXAMPLE OF COMBINED METRIC

Σ	$\sigma \in R^+$
$\oplus$	$\lambda \oplus \sigma = \lambda + \sigma$
L	$\lambda \in R^+, \lambda = {bandwith}/{delay \cdot reliability}$
$\preccurlyeq$	$\leq$

TABLE V. EXAMPLE OF MULTIPLE METRIC (LEXICOGRAPHIC ORDER)

Σ	$\Sigma_{\text{hop}} \times \Sigma_{\text{BW}_{\text{link}}} \times \Sigma_{\text{BW}_{\text{total}}} : < \sigma_h, \sigma_{bl}, \sigma_{bt} >$
$\oplus$	$(\lambda_{bl}, \lambda_{bt}) \oplus (\sigma_h, \sigma_{bl}, \sigma_{bt}) = < \sigma_h + 1, \lambda_{bl} + \sigma_{bl}, \lambda_{bt} + \sigma_{bt} >$
L	$\lambda_{bl} \in R^+, \quad \lambda_{bl} = link\ bandwidth$ $\lambda_{bt} \in R^+, \quad \lambda_{bl} = total\ bandwidth$
$\preccurlyeq$	$(\sigma_h, \sigma_{bl}, \sigma_{bt}) \preccurlyeq (\sigma'_h, \sigma'_{bl}, \sigma'_{bt})\ \ iif$ $(\sigma_h < \sigma'_h) \vee (\sigma_h = \sigma'_h \wedge \sigma_{bl} < \sigma'_{bl}) \vee$ $(\sigma_h = \sigma'_h \wedge \sigma_{bl} = \sigma'_{bl} \wedge \sigma_{bt} \geq \sigma'_{bt})$

4) *Metrics based on constraints*

This metric strategy also represents the cost value of a link with many different values [34]. The difference resides in the way to select the best path. This selection is based on a subset of constraints that defines a range of values. If the metric of a path is between these values, it is considered as a feasible path. Therefore, this strategy does not have the objective to minimize or maximize a path metric to a destination. An issue to be solved is the method used to manage *n* constraints. There are situations in which this method cannot find a correct path [35] even though it exists.

E. Efficiency and routing overhead

A routing protocol has to miss the minimum bandwidth by reducing the communication overhead. If this requisite is not accomplished, the traffic of the SEN could be affected by the routing updates causing the failure of the QoS agreements. Communication overhead could be caused by too large periodic updates or due to too frequent updates (for example when dynamic metrics are used). One possible solution is to apply a hierarchical routing protocol to reduce the information advertised inside an area. The best solution could be the use of triggered updates instead of periodic updates, if the network topology does not change frequently and it is almost stable. In this case, periodic updates are a waste of bandwidth because there is no change to announce. On the other hand, in case of a frequently changing network, the use of triggered updates can result in a misleading operation. If a node generates a lot of triggered updates it can saturate the network because of the domino effect. This can be avoided applying a timer when the triggered updates are received; the expected behavior is the reduction of the overhead by waiting enough time to receive and retransmit subsequent updates.

Another solution is used, for example by EIGRP, by limiting the amount of bandwidth that can be used by the routing protocol. This approach is efficient but it can affect the performance of the routing protocol when more bandwidth than the assigned is needed and so leading some nodes to lose the connectivity with some destinations.

F. Load balancing schemes

A load balancing scheme is not a requisite of a routing protocol, but it is necessary to specify it when multipath routing protocol is used in order to take advantage of the multiple paths to a single destination. The efficiency of the load balancing scheme would rely partially on the metric used by the routing protocol because this information is used by some load balancing algorithms. Otherwise, if more than one path is used simultaneously, there is the possibility of introducing variable delays. This delay variation affects dramatically to transmissions based on TCP protocol, by activating the Fast-Retransmit method and wasting more bandwidth. This behavior could increase the number of lost packets thus reducing the actual throughput.

Load balancing could be carried out in a flow-based or packet-based strategy. The flow-based mechanism is based on information such as IP address or a hash of the information of the flow. The advantage of this approach is that the packet reordering is not needed because a single flow uses a unique path. Obviously, its main disadvantage is inefficiency using the bandwidth of the network. For

example, an intensive flow can congest a single path whereas another data flow could be using a better uncongested path as the load balancing scheme cannot cope with this situation. When packet-based load balancing is used, the bandwidth of the network is efficiently allocated but reordering is needed in the destination.

VI. MULTIPATH ROUTING PROTOCOL ANALYSIS BEHAVIOR

A. Study groups for simulations

In this section, DIVs behavior is defined and also a brief description of the random scenario used to run the simulations is given. Even though Distributed Bellman Ford (DBF) has been studied, the behavior of this protocol is not described in this section because it is a well-known protocol and is the base of a standard protocol (RIP).

The modeled routing protocol is based on DIV [9] which is the most evolved distance vector routing protocol studied in [1]. DIV has two types of synchronization mechanism, one called local and another called alternate. Both mechanisms have been modeled in this paper. The local method only synchronizes the routing information with one hop when an increase in the metrics is detected. On the other hand, there is the alternate method: this type of synchronization is a common synchronization method that notifies all nodes affected by the metric increment (L). The local method is the fastest one but the stability of the network could be compromised because it could happen that a node cannot reach another node in the network when the routing is converging. The alternate method is more robust because the old path affected by the metric increment is maintained until all the nodes have been informed of the metric increment and have made the correct changes.

In our approach, we have changed the addressing of the routing protocol to maintain a hierarchy of 16 SEN's areas with a maximum of 256 nodes per area. This hierarchy was applied to limit the routing information exchanged inside an area, improving the efficiency of the protocol in terms of bandwidth. The other decision that has to be taken is the metric used in order to implement a QoS-aware protocol: this metric has been EBSP. EBSP metric has a better response than BSP [27] and fits correctly in heterogeneous networks which are typical in SENs. Another advantage of this metric is its representation with a single value, reducing the amount of bandwidth needed to distribute the routing information.

To understand the model implemented in the simulator, we have represented the implementation with an activity diagram in UML. Fig. 2 represents the general behavior of the protocol.

This protocol has been implemented in the OPNET simulator and it has been generated a subset of scenarios to study the protocol. These scenarios were randomly generated according to two criteria:

- The average number of links per node.
- The number of nodes in the network.

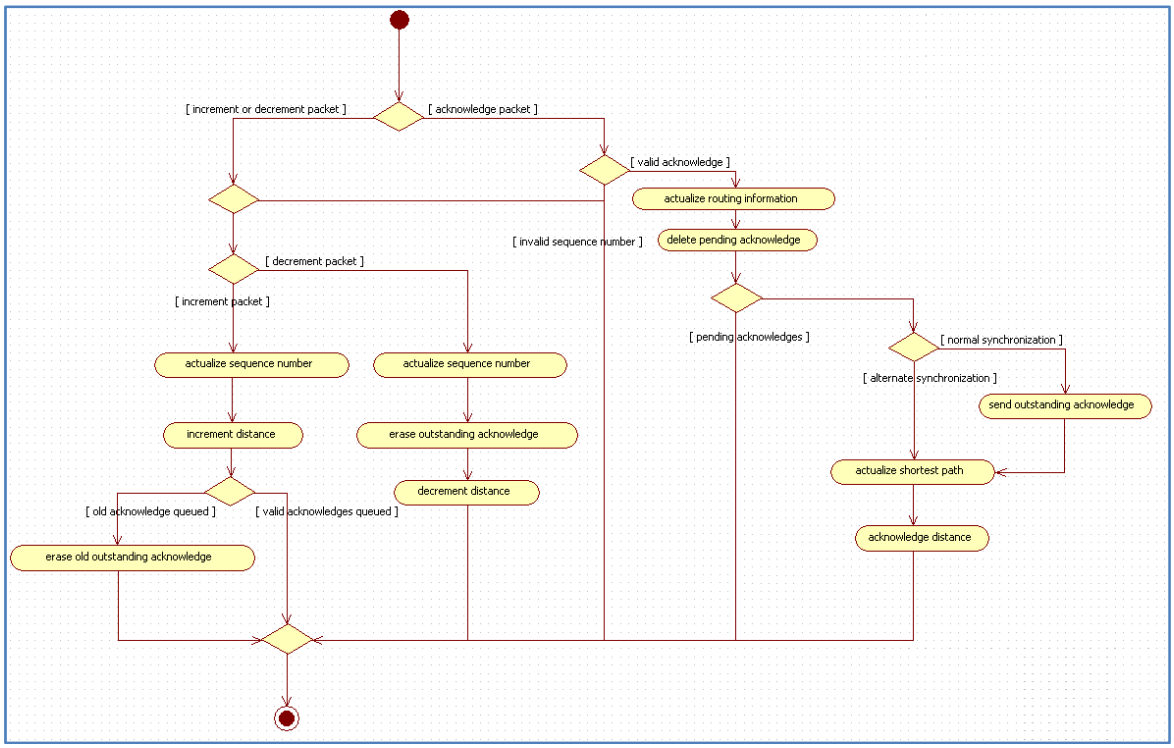


Figure 2. General behavior of the routing protocol [9]

For this study, 4 study groups of 5 scenarios have been generated. The average number of links per node for each study group is: 2 links for the study group number 1, 3 links for the study group number 2, 4 links for the study group number 3 and 5 links for the study group number 4. The number of nodes is 256 and this number is the same for each scenario.

B. OPNET simulator

The OPNET simulator [10] is an event oriented simulator. The language used to program a new node is a proto-C language specifically designed for this simulator. The behavior of the node is modeled through different phases. The first phase is the node model, which consists in designing the flows of information into the node. These flows are modeled among modules and each module contains processes. These processes are modeled in the second phase, which consists in specifying the Finite State Machine (FSM) that rules the behavior of the module. Finally, the third phase consists in programming the different states modeled into the FSM.

This section describes the two parts of the model done over OPNET modeler. The first part is the design of the node to implement the two variants studied in this paper: DBF and DIV. The second part introduces the automatic scenario generation tool created specifically for this study.

The node model designed is very simple. It contains sixteen transmitters and sixteen receivers, all connected to a simple queue module. The node has sixteen point-to-point receivers and transmitters because it is the maximum number of neighbors allowed in the design of the protocol. The module selected to interconnect all the transmitters and receivers was a queue because it can manage different queues and is easier to store, receive and transmit the messages. This queue contains the process model shown in Fig. 3.

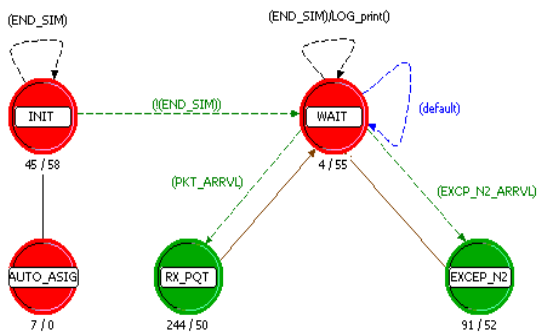


Figure 3. FSM of the models to study the protocols DBF and DIV in the OPNET simulator

The behavior of the machine state can be extracted from the UML diagram shown in Fig. 2. Nevertheless, the tasks done in each state of the finite state machine are briefly described.

- The AUTO_ASSIGN state assigns an address to the node and also initializes some variables used in the INIT state.
- The INIT state focuses on the initialization of all the variables used by the node including all the variables used by the routing protocol.
- The WAIT state waits for the arrival of a packet or the disconnection of the lower level which means that a failure of the link or the node has taken place.
- The RX_PQT is the main state of the process model. It processes all types of packets that can be received: metric increment, metric decrement or acknowledge. In this state is where all the actions derived from a metric increment or decrement are programmed (Fig. 4-6).
- The EXCEP_N2 state is a simple state to detect if the failure is from a link or from a node and initialize the variables according to the type of failure and send the corresponding messages of metric decrement or increment to the rest of the neighbors.

The advantage of this design is the generic states, which allow to model different behaviors with the same FSM and reduce the time wasted to implement the DBF and DIV algorithms.

C. Automatic scenario generation

One of the objectives of the study done in this paper is to analyze a range of similar scenarios to extract the correct results which allow to support the conclusions over them. The scenario is characterized with two parameters: the number of nodes and the average number of interconnections per node. Nevertheless, the automatic scenario generation tool within OPNET does not allow creating a scenario only based on these two parameters. OPNET has a powerful automatic scenario generator which allows creating full-mesh, partial-mesh, tree or bus topologies, but none of these topologies fit in our goal.

For this reason, an application that generates an *xml* file which can be imported into OPNET with the designed scenarios has been created. The main found problem when programming the application was to find out how to write an *xml* file to import it into the OPNET simulator. The second problem was the type of topology, which must connect physically all the nodes in the same network, without any isolated node. The other point of the generated network is the average of interconnections. The number of interconnections cannot exceed the maximum specified to the application.

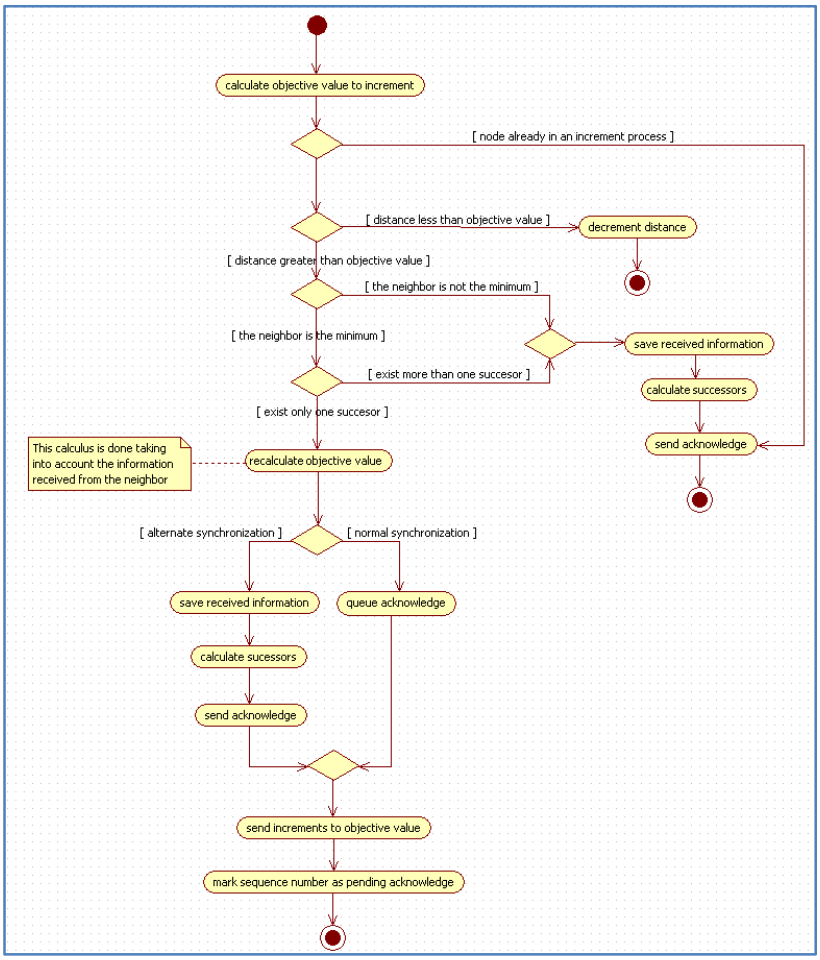


Figure 4. Metric incrementation algorithm

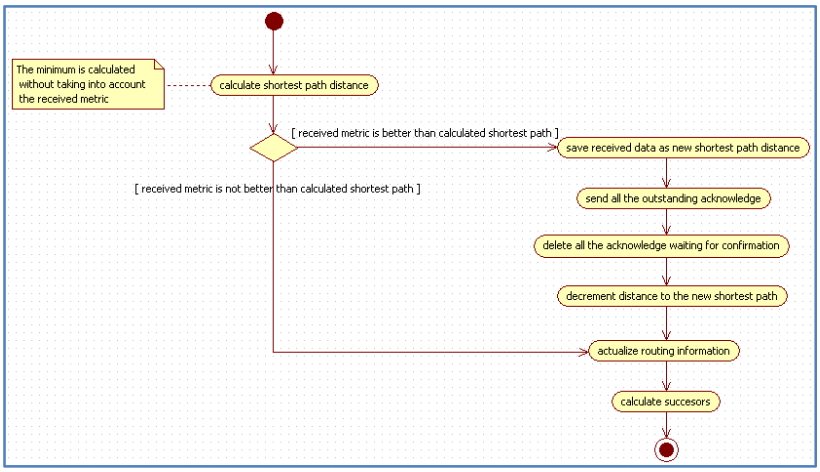


Figure 5. Metric decrementation algorithm

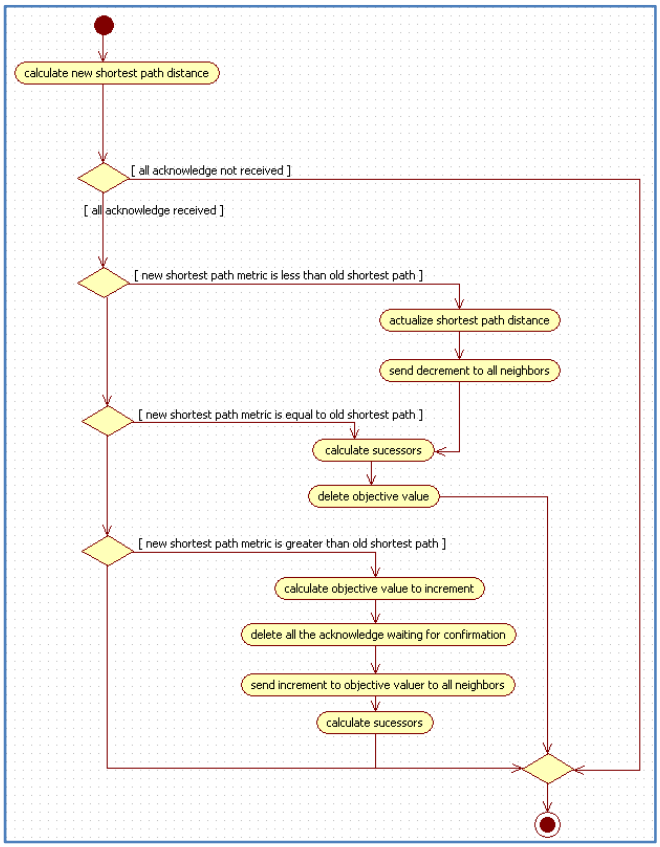


Figure 6. Acknowledgment process

Finally, the most relevant point of the automatic generated network by the simulator tool is the random interconnection of the nodes, which allows to generate five scenarios with the same specifications but with completely different physical interconnections among the nodes. This is the main advantage of this application, the possibility of generating a variety of scenarios with the same definition (number of nodes and average number of interconnections). This variety of scenarios gives the possibility of studying different physical topologies similar to those provided by SENs with the two implementations of the analyzed routing protocols.

D. Simulations and analysis

First of all, it is described the study on the number of found paths in the four groups of simulated scenarios (study groups), which depends on the average number of links in each scenario. Fig. 7 shows the number of paths found by the routing protocol where the x-axis represents the number of found paths and the y-axis represents the percentage of nodes with this number of paths to the destination. In Fig. 7, it can be seen that all nodes in the network find a minimum of one path to its destination, moreover, the scenario with an average of two links per node have a 10% of nodes with 2 paths which is the maximum in the network. When the number of links per node is increased to an average of five links, the 50% of the nodes in the network has more than two paths. This behavior gives a lot of stability to the protocol because a node has an alternative path to the destination if the primary fails.

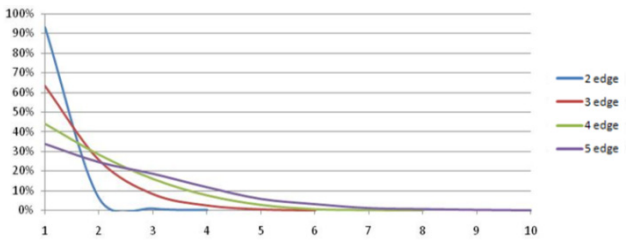


Figure 7. Number of paths found

The second topic that must be analyzed is the delay of an IP packet when there is traffic in the network. To simulate this, the same traffic pattern in all the nodes in the network has been configured. The configured pattern transmits 510 packets per node, which is enough to compare the delay in the network with 2, 3, 4 and 5 links per node. Each packet is transmitted to a different destination in the network and each node has 255 destinations configured, this makes two packets per destination.

This study presents the performance of DBF, which is a basic implementation of a Bellman Ford algorithm based on

a hop count metric. In addition to this, in order to evaluate the performance of the multipath routing, DIV has been modeled with a limitation in the number of feasible shortest paths used to route the traffic. This limitation allows us to study the improvement introduced when multiple found paths for the protocol are used.

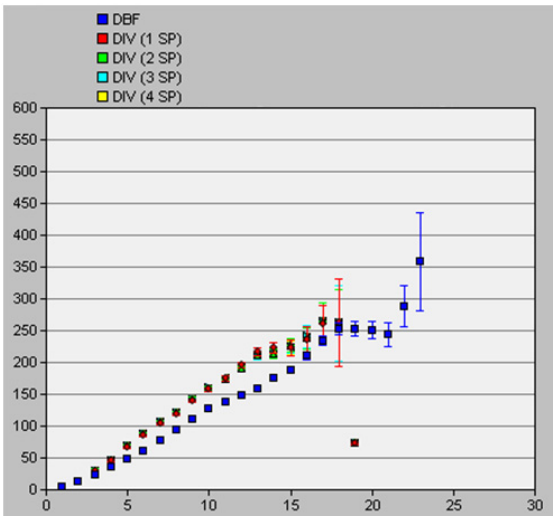


Figure 8. Delay vs. Hop count in the scenario with an average of 2 links per node

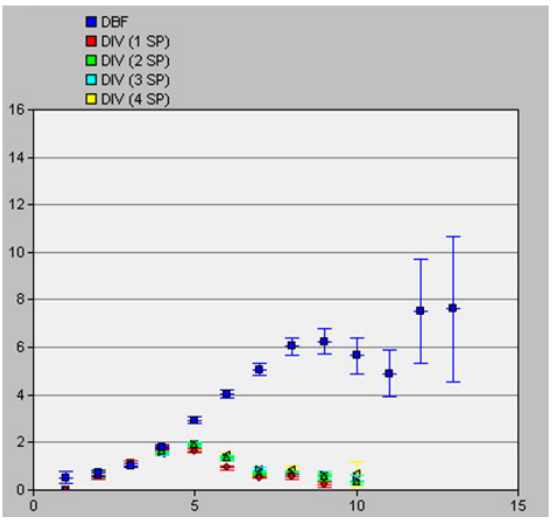


Figure 9. Delay vs. Hop count in the scenario with an average of 5 links per node

Fig. 8 and Fig. 9 show the results of the delay with a confidence interval of 98%. The implementations shown are DBF and DIV with limitations of 1, 2, 3 or 4 feasible shortest paths. Fig. 8 shows the delay (y-axis) according to the hop count (x-axis). The graph presents the results in the 2-link per node scenario where the x-axis represents the number of hops done by the analyzed IP packets and the y-axis represents the delay of IP packets. Fig. 9 presents the results in the 5-link per node scenario where the x-axis represents the number of hops done by the analyzed IP packets and the y-axis represents the delay of IP packets.

The results exposed in Fig. 8 and Fig. 9 show an interesting conclusion: when the scenario has 2 links per node, the best result is obtained by DBF with a metric of hop count; whereas the DIV implementation in the scenario with 5 links per node, which is supposed to be more prone to multipath routing, can take advantage of the network topology by reducing the overall delay of the packets.

When the difference between using 1, 2, 3 or 4 feasible shortest paths on DIV is evaluated, there is no much difference, the only conclusion is that the more paths are used, the more delay is introduced; but this delay is very low.

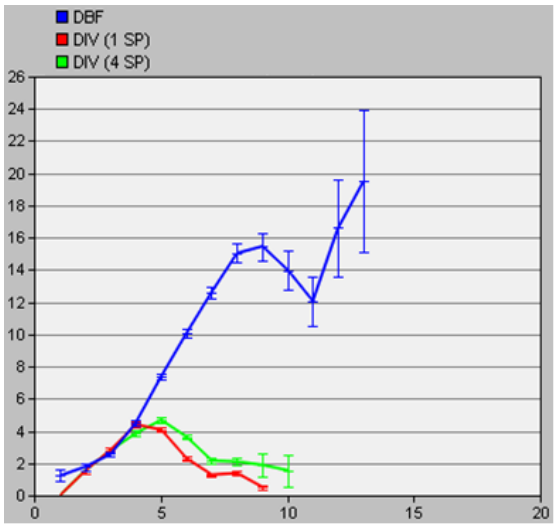


Figure 10. Packet delay vs. Hop count in the scenario with 5 data packets generated per node

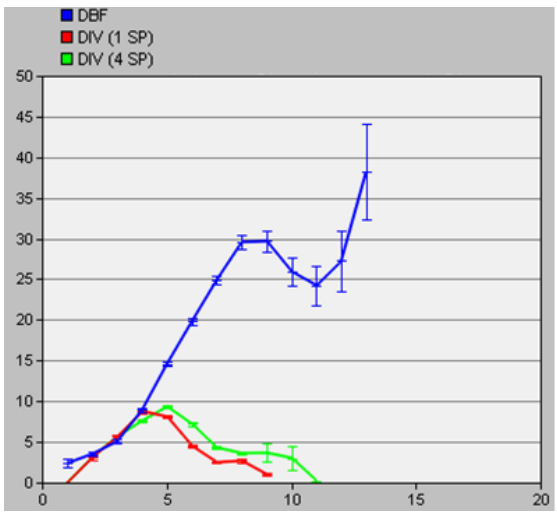


Figure 11. Packet delay vs. Hop count in the scenario with 10 data packets generated per node

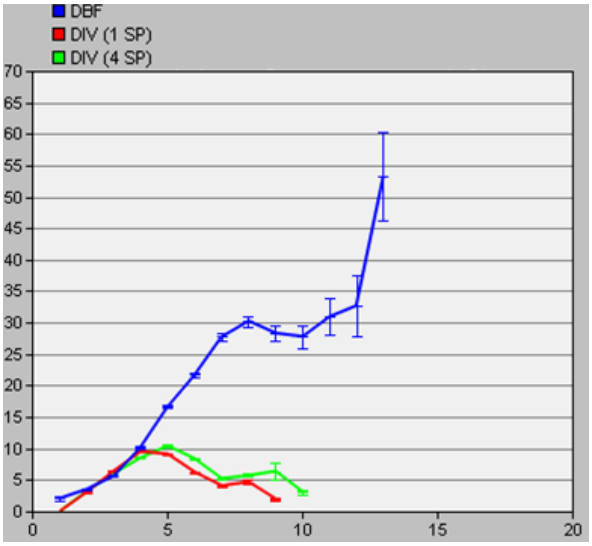


Figure 12. Packet delay vs. Hop count in the scenario with 20 data packets generated per node

The last study done is about the behavior of the routing protocol when the network is congested. The traffic pattern used to congest the network was slightly different. This new pattern configures the transmission of 2, 3, 5, 10 and 20 data packets per node and per destination. Again, the study is focused on the implementations of DBF and DIV, but in the case of DIV the study focuses only on the limitation of 1 and 4 feasible shortest paths. The results obtained with the pattern of 5, 10 and 20 packets are shown in Fig. 10-12 with 98% confidence interval. For example, Fig. 10 shows the average packet delay (y-axis) in function of the hop count metric of the path (x-axis). This graph presents the pattern with 5 packets per node.

The results obtained show that DBF has a very bad performance, whereas DIV can take advantage of the multipath routing. It is, in this situation of congestion, when a protocol oriented to provide QoS must react better and this is the case of DIV.

VII. CONCLUSION

Multipath routing protocols based on distance-vector have emerged as the evolution of its predecessor (shortest-path routing protocols). The study has concluded that these enhanced protocols improve a lot of aspects such as the increment of the network capacity and they improve the redundancy. The main advantage is the increment of the network efficiency, increasing the practicable bandwidth with the same resources and minimizing the delay of the packets.

The design of a multipath protocol is more demanding than a shortest-path routing protocol. The reason is the increase in the number of usable paths. That increase is proportional to the potential loop problems. On the other hand, the amount of routing information to transmit is greater and the mechanism to synchronize all this information is harder to design. All the mechanisms applied on shortest-path such as poison reverse, hold down timers and triggered updates do not fit enough multipath routing. The mechanism with better performance is the use of LFI in order to avoid loops and to synchronize the routing information. From our point of view, DIV-based routing algorithm is the best protocol to use as a baseline in the design of a multipath routing protocol for SENs.

A correct metric must be selected if it is needed to focus the design on QoS providing. Avoiding combined metrics and metrics based on constraints seems to be a good practice, although the last one is a widespread practice to provide QoS-aware routing. Another aspect that the metric must accomplish is to have the enough granularity to find multiple paths, this is the case of the metric used in EBSP which effortlessly finds multiple paths easier than other metrics strategy.

The OPNET simulator is a powerful tool to model a protocol from scratch, giving the possibility of customizing the node behavior. From the point of view of an implementation, it brings the possibility to test the feasibility of a design and study its performance. The practical study has shown that the multipath protocol can find more paths even when there are only two links per node. The increase in paths allows to reduce the overall delay in a transmission and allows to orient the protocol to provide QoS as well. To sum up, the protocol studied in generated scenarios (e.g., Fig. 13) together with the EBSP metric are a good option to implement a QoS-aware routing protocol for SENs.

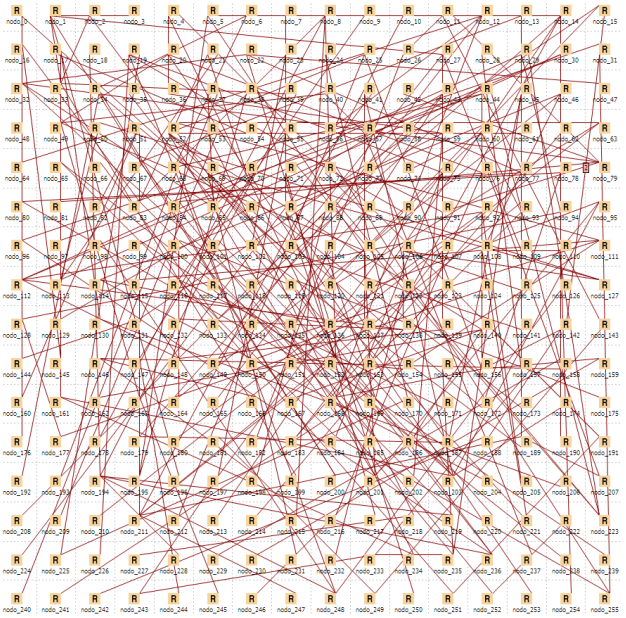


Figure 13. Example of an automatic generated scenario

Finally, an important conclusion is the relevance of the load balancing scheme. This scheme is not usually a part of the routing protocol but there exist different possible implementations that can take more advantage of the multiple paths found by the routing protocol.

ACKNOWLEDGMENT

This work was supported in part by EU's seventh framework funding program FP7 (INTEGRIS project ICT-Energy-2009 under grant 247938). Authors would like to thank "Enginyeria i Arquitectura La Salle" (University Ramon Llull) for their encouragement and assistance.

REFERENCES

[1] Zaballos, A., Vallejo, A., and Ravera, G. "Issues of QoS multipath routing protocol for SEN's data networks". Proceedings of the Sixth Advanced International Conference on Telecommunications, pp. 364 - 369 (AICT 2010).

[2] Corinex Communications Corp. "Broadband over Powerline for Smart Grids Technology Brief". [Available online] [Consulted: Dec 2010] http://www.corinex.com/.

[3] EPRI's IntelliGrid initiative, Electric Power Research Institute. [Available online] [Consulted: Dec 2010] http://intelligrid.epri.com.

[4] European SmartGrids Technology Platform. "Vision and Strategy for Europe's Electricity Networks of the Future". [Available online] [Consulted: Dec 2010] http://www.smartgrids.eu/.

[5] Zaumen, W.T. and Garcia-Luna-Aceves, J.J. "Loop-free Multipath routing Using Generalized Diffusing Computations". Proceedings of the IEEE INFOCOM 2008, pp. 1-10.

[6] Vutukury, S. and Garcia-Luna-Aceves, J.J. "MDVA: A distance-vector multipath routing protocol". Proceedings of the IEEE INFOCOM 2001, vol.1, pp: 557-564.

[7] Vutukury, S. and Garcia-Luna-Aceves, J.J. "A simple approximation to minimum-delay routing". Proceedings of the ACM SIGCOMM 1999, pp.227-238.

[8] Vutukury, S. and Garcia-Luna-Aceves, J.J. "MPATH: A Loop-free Multipath Routing Algorithm". Elsevier Journal of Microprocessors and Microsystems (1995).

[9] Ray, S., Guerin, R.A., and Sofia, R. "Distributed Path Computation without Transient Loops: An Intermediate Variables Approach". *Proceedings of International Teletraffic Congress* (2007).

[10] OPNET Technologies, Inc. OPNET University Program. [Online] http://www.opnet.com/services/university/.

[11] Pritchard, J. K. "Energy Security: Reducing Vulnerabilities to Global Energy Networks," 2009. [Online]. Available: http://www.dtic.mil.

[12] Zaballos, A., Vallejo, A., Jimenez, J., and Selga, JM. "QoS Broker based management for heterogeneous Smart Electricity Networks". Proceedings of the IEEE symposium on Computers and Communications (ISCC 2010), pp. 295-297.

[13] Vallejo, A., Zaballos, A., Vernet, D., Orriols, A., and Dalmau, J. "A Traffic Engineering proposal for ITU-T NGNs using Hybrid Genetic Algorithms". The International Journal On Advances in Internet Technology Edited by IARIA (2009) vol. 2, pp: 162-172.

[14] Sobrinho, J. L. "Algebra and Algorithms for QoS Path Computation and Hop-by-Hop Routing in the Internet". IEEE/ACM Transactions on Networking (2002), vol. 10, num. 4, pp. 541-550.

[15] Sobrinho, J. L. "An Algebraic Theory of Dynamic Network Routing". IEEE/ACM Transactions on Networking (2005), vol. 13, num. 5, pp. 1160-1173.

[16] Orda, A. and Sprintson, A. "Efficient algorithms for computing disjoint QoS paths". Proceedings of the IEEE INFOCOM 2004, vol. 1, pp.730 – 738.

[17] Garcia-Luna-Aceves, J.J. "Loop-free routing using diffusing computations". IEEE/ACM Transactions on Networking (1993), vol.1, num. 1, pp.130-141.

[18] Yuan, X. "On the extended Belman-Ford algorithm to solve two-constrained quality of service routing problems". Proceedings. Eight International Conference (1999), pp. 304-310.

[19] Dijkstra, E.W. and Scholten, C.S. "Termination detection for diffusing computations". Inform process lett, vol. 11, num. 1, pp. 1-4. (1980).

[20] Li, Z. and Garcia-Luna-Aceves, J.J. "A distributed approach for multi-constrained path selection and routing optimization". Proceedings of the 3rd international conference on Quality of service in heterogeneous wired/wireless networks, ACM International Conference Proceeding (2006), vol.191.

[21] Turgay, K., Marwan, K., and Spyros, T. "An Efficient Algorithm for Finding a Path Subject to Two Additive Constraints". Proceedings of ACM SIGMETRICS 2000, vol.28, num.1, pp.318-327.

[22] Katz, D. and Ward, D. "Bidirectional Forwarding Detection". Request for Comments: 5880 of Internet Engineering Task Force (IETF - 2010).

[23] Murthy, S. and Garcia-Luna-Aceves, J.J. "Loop-free Internet routing using hierarchical routing trees". Proceedings of the IEEE INFOCOM 1997, vol.1, num.7, pp.101-108.

[24] Behrens, J. and Garcia-Luna-Aceves, J.J. "Hierarchical routing using link vectors". Proceedings of the IEEE INFOCOM 1998, vol.2, pp.702-710.

[25] Perkions, C., Belding-Royer, E. and Das, S. "Ad hoc On-Demand Distance Vector (AODV) Routing". Request for Comments: 3561 of Internet Engineering Task Force (IETF-2003).

[26] Rangarajan, H. and Garcia-Luna-Aceves, J.J. "Making on-demand routing protocols based on destination sequence numbers robust". Proceedings of the IEEE International Conference on Communications (ICC 2005), vol.5, pp.3068 – 3072.

[27] Gokhale, S.S and Tripathi, S.K. "Routing metrics for best-effort traffic". Eleventh international conference on computer communications and networks (2002), pp: 595-598.

[28] Wang, J. and Nahrstedt, K. "Hop-by-hop routing algorithms for premium-class traffic in DiffServ networks". Proceedings of the IEEE INFOCOM 2002, vol.2, pp.705-714.

[29] Wang, Z. and Crowcroft, J. "Quality-of-Service Routing for Supporting Multimedia Applications". IEEE Journal on selected areas in communications (1996), vol.14, num.7, pp.1228-1234.

[30] Apostolopoulos, G., Kama, S., Williams, D., Guerin, R., Orda, A., and Przygienda, T. "QoS Routing Mechanisms and OSPF Extensions". Request for Comments: 2676 of Internet Engineering Task Force (IETF-1999).

[31] Ming-Hong, S., Si-Bing, W., and Ying-Cai, B. "A bandwidth constrained QoS routing optimization algorithm". Proceedings of the International Conference on Communication Technology (ICCT 2003), vol.1, pp.491-494.

[32] Yang, Y., Zhang, L., Muppala, J., and Chanson, S.T. "Bandwidth-delay constrained routing algorithms". Computer Networks: The International Journal of Computer and Telecommunications Networking (2003), vol.42, num.4, pp.503-520.

[33] Yuen, M.; Cheung, C. "Efficient path selection for QoS routing in load balancing". Proceedings of the 9th Asia-Pacific Conference on Communications (APCC 2003), vol.3, pp.988-992.

[34] Shigang, C. and Nahrstedt, K. "On finding multi-constrained path". Proceedings of the IEEE International Conference on Communications (ICC 1998), vol.2, pp.874-879.

[35] Korlmaz, T. and Krunz, M. "Multi-Constrained Optimal Path Selection". Proceedings of the IEEE INFOCOM 2001, vol.2, pp.834-843.

Efficiency Benefits Through Load-Balancing with Link Reliability Based Routing in WSNs

Chérif Diallo, Michel Marot, Monique Becker

SAMOVAR CNRS Research Lab – UMR 5157
Dept Réseaux et Services de Télécommunications (RST)
Institut TELECOM – TELECOM SudParis
9, Rue Charles Fourier – 91011 Evry CEDEX, France
Email:{cherif.diallo, michel.marot, monique.becker}@telecom-sudparis.eu

***Abstract*—In wireless sensor networks (WSN) energy efficiency of routing protocols is of primary importance. Embedded with local load balancing mechanisms, the proposed L2RP protocol is a link reliability based routing protocol which aims to help source nodes to exploit the potential capabilities of their respective neighbors. As it is a reliability-oriented protocol, L2RP discards unreliable links to avoid the substantial energy cost of packet losses. Simulation results show major efficiency benefits that stem from load balancing which helps in lengthening the network lifetime while minimizing packet losses.**

In WSN, the choice of a routing protocol and its key parameters depends on the nature of the application and on its primary mission. Lot of works addressed routing issues with more or less effectiveness, some of which pointed out the use of the link quality indicator (LQI) as a route selection criterion (metric). In a previous work, following an experimental study, we have shown, under some conditions, the inefficiency of the LQI based routing. In this paper, we propose through L2RP a simple way to improve reliability and efficiency of the LQI based routing in WSN. We also give a comparative study of several metrics including new definitions of LQI based metrics. Simulation results show that our adaptation of the LQI metric is among the best route selection criteria regardless of the performance criterion under consideration.

***Index Terms*—Wireless Sensors Networks (WSN); Load-Balancing Routing; LQI; L2RP; Energy Efficiency.**

I. Introduction

Designing a cold chain monitoring application requires special focus on at least two main phases. In [2], we presented an example of sensor network for cold chain monitoring where sensors are inside pallets. We proposed energy efficient protocols for the transport phase in which the WSN is deployed in trucks with no Base Station (BS) because it would be very expensive to install and maintain Base Stations within each truck. There are a few sensors in the truck.

The second phase concerns the product storage in a warehouse where each pallet is handling temperature sensor. This application specifically collects rare events (alarms) to ensure the proper monitoring of the system. If the temperature is over a threshold, an alarm will be generated; this "interesting event" is then sent towards the BS. Due to the size of a warehouse which hosts large number of pallets, one upon the other, the WSN can reach several hundreds of sensors which collaborate for sending data towards the BS. So, in this environment, the link quality is a key parameter which has many effects on the network performance.

In [3], we used up to 50 Moteiv Tmote Sky [4] sensors, in a small experimental platform, including a 2.4GHz ZigBee [5][6] wireless transceiver (chipcon′s CC2420) [7]. On each packet reception, the CC2420 calculates the error rate, and produces a LQI value. To conduct experiments, we used the multiHopLQI[1] routing algorithm along with the Sensornet Protocol (SP) implementation [8]. In this algorithm, nodes sense and send "interesting events" to the BS. Based on the acknowledgement, a sensor decides to retransmit the data or not. If the acknowledgement fails, the sensor selects another node and routes data towards the BS. Under these conditions, the experimental results pointed out that the LQI based routing could have negative effects on the network performance [3].

After all, we think that the link quality might be a key parameter which some routing protocols could rely on in order to increase the network performance. The link quality indicator (LQI) is defined in the IEEE 802.15.4 standard [5][6] as a measurement of the quality of packet reception between two nodes. The IEEE 802.15.4 standard does not specify the implementation of LQI, which is up to the radio manufacturer. Several works address WSN routing, but only few papers are related to LQI based routing protocols. Sensors are characterized by their low energy level. Thereby load balancing traffic between different nodes, is also an essential idea to increase the lifetime of nodes and thus the network. This work addresses two challenges: improving LQI based routing protocol by load balancing traffic over multiple paths.

When a sensor has to send data towards the Base Station, the load balancing routing consists to elect several nodes as next hop routers depending on the order of packet transmissions and the nodes previously used as the next hop routers. The idea is to involve several sensors in the routing effort to minimize the overall energy consumption and then extend the network lifetime.

[1] http://www.tinyos.net/tinyos-1.x/tos/lib/MultiHopLQI

The metric is a property of a route in computer networking consisting of any value used by routing algorithms to determine whether one route should perform better than another. Commonly, the route with the lowest metric is the preferred route. However, in this paper, a metric means the local value associated with a node: for a source node, the highest value, in its neighbourhood, may lead to the selection of such a node as the next hop router. For instance, The remaining energy level can be used as a metric to promote the selection of the highest powered nodes as next hop routers.

In this paper, we propose WSN local load balancing routing mechanisms using the Wait and See (WaS) protocol [2] by comparing the following metrics: the remaining energy level, the degree of connectivity (number of neighbors), the sensor proximity with respect to the Base Station, the link quality indicator (LQI), and a hybrid metric composed of any pairs of these metrics.

The sensor networks are characterized by low energy constituting their batteries. Then energy consumption and some other performance criteria such as the load imbalance factor (LIF), the average path lengths, the network lifetime and the packet loss percentage are taken into consideration to evaluate the effectiveness of routing mechanisms.

"Achtophorous Node" definition: we focus on homogeneous WSN where all sensors are participating together in the routing effort. Since all nodes are routers, we prefer using the term "achtophorous node" derived from Greek term $\alpha\chi\theta o\varphi o\rho\varepsilon\omega$ which denotes "node handling heavy load". For each node sending data, its achtophorous nodes are its next hop sensors which handle the load due to the routing of its packets towards the BS. Each sensor selects among its neighbors one or more achtophorous nodes. We also examine the influence of increasing the number of the achtophorous nodes on the routing efficiency. The WSN deployed in a warehouse is proned to some unreliabilities of wireless links. Then, we present results pertaining to unreliable links impacts on the network performance.

The rest of this paper is organized as follows. After presentation of a short background in the next part, the next one gives some topics on studied metrics (Section III). Then, we describe load balancing mechanisms (Section IV) and the proposed routing protocol (Section V). Finally, the last two sections present the simulation model and the results.

II. Related Works

Commonly used by the TinyOS community, MultihopLQI is a routing protocol which employs the cost-based paradigm defined in [9]. Link estimation is viewed as an essential tool for the computation of reliability-oriented route selection metrics. In MultiHopLQI, the link metric is the Link Quality Indicator (LQI) which is used additively to obtain the cost of a given route. MultihopLQI avoids routing tables by only keeping state for the best parent at a given time, drastically reducing memory usage and control overhead. A new parent is adopted if it advertises a lower cost than the current parent.

Many experimental studies related to WSN, some of which are based on MultiHopLQI, [3][10][11][12][13][14][15][16] have shown that high unreliability of wireless links must be explicitly taken into account when designing routing protocols. [11][12] address load balancing embedded in reliability-oriented routing protocols and are also using MultiHopLQI.

In [17][18] authors address the problem of minimizing the total consumed energy to reach the destination. The performance objective of maximizing the network lifetime was considered in [19][20].

Several works are related to WSN and ad hoc networks load balancing routing schemes [21][22][23][24][25][26]. In [21], authors show that distributing the traffic generated by each sensor node through multiple paths instead of using a single path allows energy savings. Paper [22] defines a network optimization problem used for performing the load balancing in wireless networks with a single type of traffic. In [23], authors study wireless network routing algorithms that use only short paths, for minimizing the latency, and achieve the load balance. In [24], authors introduce a collision awareness in multipath routing; while [25] propose a multipath routing protocol to address the congestion control issue in WSN. In [26], the challenge of maximizing the network lifetime by load balancing the traffic is covered. In order to balance the energy consumption among sensor nodes, they deploy multiple sinks simultaneously, which are connected through wired or wireless infrastructure. [27][28] and [29] are also related to load balancing routing protocols.

The paper [30] presents a resource-aware and link quality based (RLQ) routing metric. Based on both energy efficiency and link quality statistics, the RLQ metric in [30] is intended to adapt to varying wireless channel conditions, while exploiting the heterogeneous capabilities. This protocol does not include load balancing features.

Some works are taken into account the round-robin cluster based routing [31][32] and [33], where clusterheads are selected on a round-robin fashion. In [34] authors propose a source count (packets) based weighted round-robin forwarding algorithm.

Although all these studies provide a valuable and strong contribution in WSN routing, the problems of load balancing routing mechanisms based on local metrics, with special interest on the LQI based metrics, are yet to be addressed. This is the goal of this paper. To save energy, we exploit the broadcast nature of wireless links, and the fact that the weights, in our proposed L2RP protocol, are built upon the achtophorous nodes capabilities instead of the ones of the source node. This

allows L2RP to avoid doing a per packet load-balancing by the source, as done in [34], where the source node sends its data without being sure that the achtophorous node is able or not to sustain the load assigned. Thus, L2RP hleps in reducing packet losses. Moreover, in most of papers addressing the load balancing routing, both experimental studies and simulation models are validated for only small sized networks (few tens of nodes), whereas our work addresses large sized networks (several hundreds of sensors). The comparative study of different metrics in L2RP is also a contribution of this paper.

III. Routes Selection Criteria

In this paper, "metric" is used to refer to local route selection criterion. As defined in introduction, each time we use "Achtophorous Node" it means next hop router with respect to a specific node having data to transmit towards the BS.

A. Remaining Energy Level

The remaining energy of sensors could be a metric for selecting routes since a node with better battery life seems to be a better candidate for the packet routing from its neighbors. Conversely, if a sensor with low power is selected as an achtophorous node, this can lead to packet losses because it might not have enough batteries to forward packets. In this paper, we consider that each node knows its energy level.

B. Sensor Proximity with respect to the Base Station (Proximity-BS)

Fig. 1. Pallets arrangement inside a warehouse

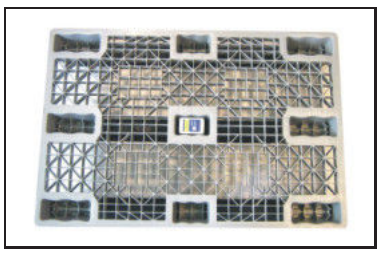


Fig. 2. Sensor plugged inside a Pallet

Fig. 3. Location of a pallet: lane, location and level

In a warehouse (see Figure 1), depending on the nature of their respective contents (frozen foods, fresh produce, etc.), the pallets provided each with a sensor (see Figure 2) are arranged in fixed locations (see Figure 3) designated by the Warehouse Management Software (WMS). Thus, during the warehouse WSN initialization, sensors could be initialized with their respective positions without using the GPS technology.

So, we consider a WSN deployed with a Base Station where each node knows its exact position and that of the BS. As the main goal of the application is to send data towards the BS, it seems natural to look at the metric defined as follows:

$$ProximityBS(S_i, BS) = 1/d(S_i, BS) \quad (1)$$

where $d(S_i, BS)$ is the distance separating the sensor S_i from the BS. We choose inverse of the distance to promote the election of the closest sensor to the BS.

C. Degree of Connectivity

The degree of connectivity of a node, i.e., the number of its neighbors, is also a metric that seems interesting to study because, intuitively, the more neighbors a sensor has, the more it seems to be an appropriate candidate as an achtophorous node since a sensor with a low degree of connectivity might have little information, from its neighbourhood, to forward to the BS. In the initial phase, each sensor is involved in the neighbourhood information exchanges (hello protocol), which allows it to determine its degree of connectivity and the BS position.

D. LQI: Link Quality Indicator

In Zigbee standard [5][6], the LQI measurement is defined as a characterization of the strength and/or quality reception of a packet. The use of the LQI result by the network or the application layers is not specified in [5][6]. The LQI measurement is performed for each received packet, and the result is reported to the MAC sublayer as an integer ranging from 0 to 255. The minimum and maximum LQI values (0 and 255) are associated with the lowest and the highest quality IEEE 802.15.4 reception detectable by the receiver, and the LQI values in between are distributed between these two limits [5][6].

For moteiv′s Tmote Sky [4] sensors equipped with chipcon′s CC2420 [7], the LQI values range from 50 to 110. Even so, we stick with the ZigBee standard [5][6] because some manufacturers, such as SUN-SPOT [35] and WiEye [36], are still using the standard LQI values. Then, we use

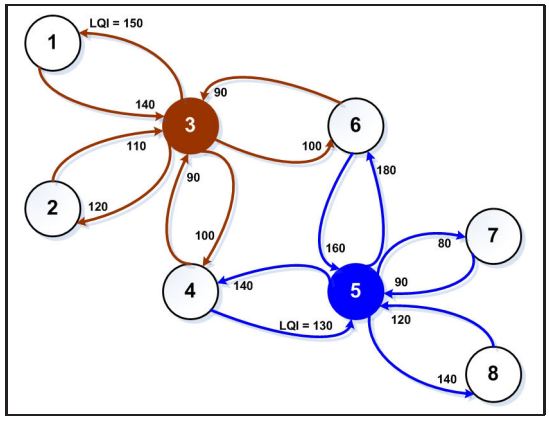


Fig. 4. Example of a WSN with Asymetrical Links

TABLE I
LQI METRIC VALUES RELATED TO THE WSN IN FIGURE 4

Sensor ID	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
AvgLQI	150	120	107.5	120	125	140	80	140
MaxLQI	150	120	140	140	160	180	80	140
MinLQI	150	120	110	100	120	100	80	140

the standard values (i.e., $[0, 255]$), instead of those of CC2420.

In this paper, we define three LQI based metrics: AvgLQI, MaxLQI and MinLQI. The AvgLQI metric is the average calculated from the LQI values of all the links between the node and its neighbors. AvgLQI values give a characterization of sensors throughout their respective coverage quality. This metric might be useful in the context of the WSN deployed in a warehouse which hosts a large number of pallets, one upon the other. Such an environment is proned to high unreliability of wireless links. the MaxLQI metric is the maximum LQI value which matches to the standard definition of the LQI used in the MultiHopLQI routing algorithm [3][8]. As for the MinLQI, it is the minimum value beyond the given LQI threshold. For example (see Figure 4), assuming that the LQI threshold for an acceptable link quality is 100, the MinLQI for node 5 is 120 (LQI of link 5-8) instead of 90 (LQI of link 5-7). Thus, Table I gives LQI metrics values for the WSN in Figure 4.

E. Composite or Hybrid Metric

In this paper, we define the composite metric (hybrid) as follows:

$$Hybrid(LQI, M_i) = \rho * LQI + (1 - \rho) * Sc(M_i) \quad (2)$$

$$Hybrid(M_i, M_j) = \rho * Sc(M_i) + (1 - \rho) * Sc(M_j) \quad (3)$$

where $Sc(M_i)$ is a scale function, which returns remaining energy values comparable to LQI values. This help avoiding the composite metric to be strongly influenced by the M_i component in (2):

$$Sc(M_i) = \alpha + \frac{\beta * \log(1 + (M_i - M_{i,min}))}{\log(1 + M_{i,max})} \quad (4)$$

Where M_i is a metric, $M_{i,min}$ (resp. $M_{i,max}$) is the minimum (resp. maximum) value of M_i. If M_i is the remaining energy of the node, $M_{i,min}$ represents the value under which, the sensor is considered dead (battery depletion); while $M_{i,max}$ is the initial energy value of a new battery. $\alpha = 50$, $\beta = 255$.

Like the LQI metrics definition, we can also define AvgHybrid, MaxHybrid and MinHybrid metrics depending on whether, we are respectively considering AvgLQI, MaxLQI and MinLQI as defined in Table I.

IV. ROUTING MECHANISMS

A. Simple Routing

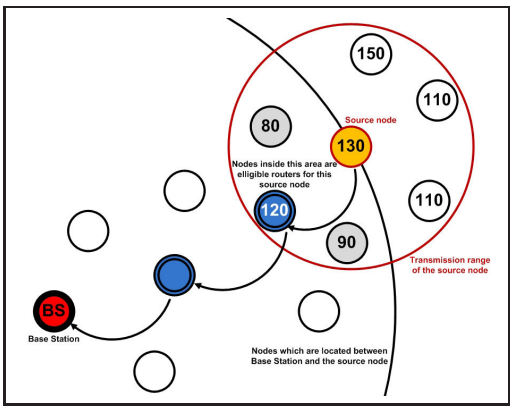


Fig. 5. Simple Routing: Nodes with their Metric Values

In the simple routing mechanism, each sensor S_i selects an achtophorous node which matches the highest metric in its vicinity and located between the sensor S_i and the BS. For each given sensor, a unique achtophorous node plays the next hop role for all its packets until the next election (see Figure 5).

B. Round-Robin Routing

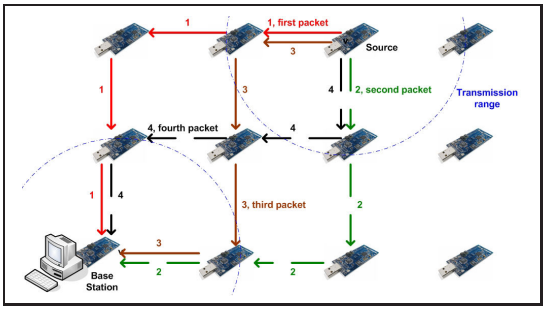


Fig. 6. Round-Robin Routing: Multiple routes from each source

In the round-robin routing, each source node has to elect two or more achtophorous nodes. The source node sends data in round-robin fashion, simply taking turns which achtophorous node it routes each packet out (see Figure 6). This routing mechanism is a per-packet load balancing routing which gives most even distribution across next achtophorous nodes.

This per-packet load balancing method means that packets in a particular connection or flow arrive at their destination out of sequence. This does not cause a problem for most applications, but it can cause problems for the increasingly popular streaming media, both video and audio. In this paper, only data packets are concerned within cold chain monitoring application for which the packet sequence order is not an issue.

C. Weighted Round-Robin Routing (W2R routing)

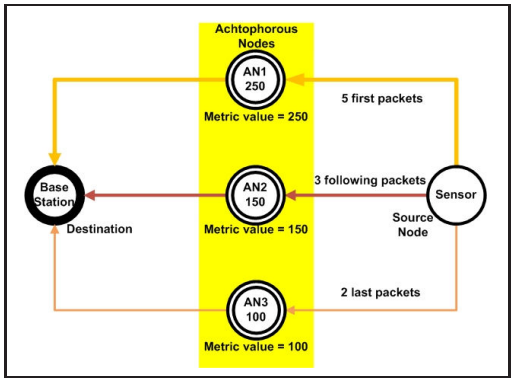


Fig. 7. Weighted round-robin routing (W2R routing)

TABLE II
WEIGHT OF ACHTOPHOROUS NODES IN FIGURE 7

Achtophorous Node	*Metric*	*Weight*	*Load handled*
$AN1$	250	0.5	50%
$AN2$	150	0.3	30%
$AN3$	100	0.2	20%

The weighted round-robin routing (W2R routing) is a load balancing mechanism that involves assigning a weight to each achtophorous node. Weights are proportional to metric values. In the W2R routing, each achtophorous node is assigned a value that signifies, relative to the other achtophorous nodes in the routing table, how the source node performs. The weight determines how many more (or less) packets are sent to that achtophorous node, compared to the other achtophorous nodes (see Figure 7). The W2R routing is one way addressing some shortcomings. In particular, it provides a clean and effective way by focusing on fairly distributing the load amongst available achtophorous nodes, versus attempting to equally distribute data packets.

For example, in Figure 7, the source node routes 50% of its packets through $AN1$, 30% through $AN2$ and 20% through $AN3$. If the BS is not located within the transmission range of an achtophorous node, this one should apply the same mechanism to retransmit the packet towards the BS.

The weighted round-robin routing mechanism is computed as described in the simple Algorithm 1 which is computed each time a source node has to send a packet. The achtophorous nodes, each with its respective $weight$, are listed in the routing table of each source node in an ordered manner such that the first achtophorous node matches the highest $weight$ as shown in Figure 7. For each source node, the $window$ interval is the constant length of each stream of consecutive packets to transmit. The $weight$ of each achtophorous node is converted as an integer value based on the $window$ interval parameter. For example, in Figure 7, $window = 10$ consecutive packets, and $weight(AN1) = 5$. The $use(AN)$ function returns the number of times the current achtophorous node AN is used during the $window$ interval whereas $packet\_idx$ is the index of the current packet during the $window$ interval.

Algorithm 1 : Weighted Round Robin (W2R) Routing

Require: $packet\_idx$, $window$, AN, $weight$, use
1: **if** $packet\_idx < window$ **then**
2: **if** $use(AN) < weight(AN)$ **then**
3: $Send\_packet\_to(AN)$
4: $use(AN) \leftarrow use(AN) + 1$
5: $packet\_idx \leftarrow packet\_idx + 1$
6: **else**
7: $use(AN) \leftarrow 0$
8: $AN \leftarrow Next().achtophorous\_node$
9: $\#\ The\ next()\ of\ the\ last\ AN\ is\ the\ first\ AN$
10: $Send\_packet\_to(AN)$
11: $use(AN) \leftarrow 1$
12: $packet\_idx \leftarrow packet\_idx + 1$
13: **end if**
14: **else**
15: **for** $each\ achtophorous\_node\ AN$ **do**
16: $use(AN) \leftarrow 0$
17: **end for**
18: $AN \leftarrow First().achtophorous\_node$
19: $Send\_packet\_to(AN)$
20: $use(AN) \leftarrow 1$
21: $packet\_idx \leftarrow 1$
22: **end if**
23: **return** $packet\_idx$, AN, use

V. L2RP: THE LINK RELIABILITY BASED ROUTING PROTOCOL

The proposed (L2RP) routing protocol (see Figure 8) consists for a sensor having an empty routing table to elect one next hop router (case of simple routing) or more achtophorous nodes (load balancing routings) amongst its neighbors according to the following:

- **Initial step** : all sensors empty their routing tables.
- The sensors located in the vicinity (transmission range) of the BS send their data directly to it.
- A sensor, located outside of the vicinity of the BS, inspects its routing table:
  - If its routing table is not empty, it checks if the link with the next hop is reliable or not. If the link is unreliable, based on the LQI value, then :

  - ∗ Case of simple routing mechanism: it sends a "ROUTE REQUEST" to its neighbors.
  - ∗ Case of load-balancing routing: it chooses an alternate route and then checks again if the link with this next hop is reliable or not. If no link with achtophorous nodes listed in its routing table is reliable, then it erases the routing table and it sends a new "ROUTE REQUEST" to its neighbors.
- If its routing table is empty, it also sends a "ROUTE REQUEST" to its neighbors.
- Each neighbor, located between the BS and the sensor having sent the "ROUTE REQUEST", computes its own waiting time which is inversely proportional to its metric value. We use the Wait and See protocol (WaS), as in [2], where the only sensor having the highest metric sends a "ROUTE REPLY" to the requester node. The other neighbors simply ignore the "ROUTE REQUEST" avoiding useless "ROUTE REPLY" packets. In the case of a load balancing routing, the number (ANs) of achtophorous nodes is a known parameter in the initialization phase of the network. This parameter is used by the WaS protocol that allows ANs sensors having highest metrics in succession to answer to the requester node, and then be elected, for this node, as achtophorous nodes.
- Upon reception of the "ROUTE REPLY" packet, the requester node updates its routing table, which remains valid until the next election. In the case of weighted round-robin routing, each "ROUTE REPLY" packet contains the metric value of the answering node, which allows the requester node to calculate weights associated with each achtophorous nodes.
- At the end of the current cycle, sensors reset their routing tables and go back to the initial step of the next cycle.

Upon receipt of a "ROUTE REQUEST" packet, a sensor S_i computes its own waiting time according to the following formula:

$$Timer(S_i) = \tau + \frac{\zeta}{1 + \log(1 + M_i + \frac{id(S_i)}{\Gamma} * M_i)} \quad (5)$$

where M_i is the metric value of the sensor S_i. τ and ζ are nonzero positive constants. Γ is a constant which is more large than the network size ($\Gamma = 10^6$, for example). This timer function avoids collisions between nodes having the same metric value. Since $M_i \geq 0$, if $M_i = 0$ then the sensor S_i can not be an achatophorous node.

As we can see, in this protocol the source node uses the link quality indicator (LQI) to check if the link it forms with the nominated achtophorous node is reliable or not. This helps avoiding to send the packet to an achtophorous with which it forms a link of poor quality which could lead to packet loss.

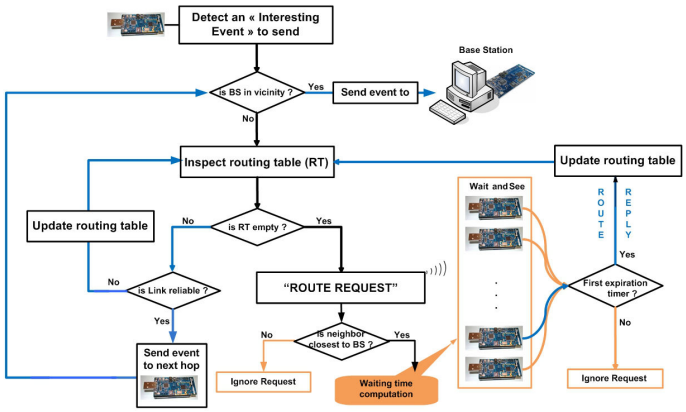


Fig. 8. The Link Reliability based Routing Protocol (L2RP) flowchart

VI. Performance Criteria

A. Average Ratio of the Remaining Energy

The average ratio of the remaining energy is the ratio of the average remaining energy on the average of initial energy. Multiplied by hundred, this value represents the average battery life of sensors, in terms of percentage. The higher this value is, the more energy-efficient the routing protocol is.

B. Average Path Lengths

The average path lengths are calculated in terms of the number of hops traversed by packets before reaching the BS. A large value reflects participation of many sensors in the effort due to the routing, which may increase the overall energy consumption. A good routing protocol is recognized in this performance criterion by a relatively low value. Conversely, too small path length may lead to bad quality link.

C. LIF: Load Imbalance Factor

The load imbalance factor (LIF) is defined as the root of the squared coefficient of variation of the relative remaining energy. This shows the energy spent by communications:

$$LIF = \sqrt{\frac{Var(E_R^i)}{\overline{E_R}^2}} \quad (6)$$

where E_R^i is the ratio of the remaining energy of sensor S_i; and $\overline{E_R}$ is the average ratio of the remaining energy.

D. Network Lifetime

In this paper, we define the network lifetime as the average number of packets routed until the first time a sensor run out of battery. This could also result in network capacity. We focus on the first battery depletion, which means the instant the network stops fulfilling totally its role, because it leads to packet losses. An ideal network is a network where all packets sent by source nodes are actually transmitted to the recipient (BS). The earlier the first packet loss happened, the more ineffective the routing protocol is.

E. Average Percentage of Lost Packets

Beyond the first time a battery depletion is experienced by the network, a high percentage of packet losses might reflect an unreliable network whose routing protocol is less effective.

VII. Simulation Model

A. Energy Consumption Model

Let $E_{Tx}(k,d)$ the energy [37][38] consumed to transmit k bits message over a distance d :

$$E_{Tx}(k,d) = E_{elec} * k + \varepsilon_{amp} * k * d^2 \quad (7)$$

Let E_{Rx} the energy consumed to receive a k bits message:

$$E_{Rx}(k,d) = E_{Rx-elec}(k) = E_{elec} * k \quad (8)$$

$$E_{elec} = 50nJ/bit \text{ and } \varepsilon = 100pJ/bit/m^2$$

B. Network Deployment and simulation parameters

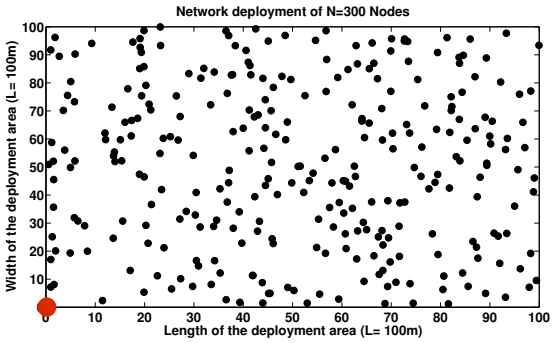


Fig. 9. Netwrok Deployment of N=300 sensors (100m x 100m)

TABLE III
L2RP Simulation parameters

Parameter	Value
Deployment	
Area Length	$L = 100m$
Area Width	$l = 100m$
Base Station Location	$position(SB) = (0,0)$
Radio range	$R = 20m$
Network Size	$N = \{100, 200, ..., 500\}$
Poisson Parameter	
Packet sent by each sensor	$\lambda = 10$
Packet Sizes (bits)	
Alarms	$k_{data} = 128$
L2RP "ROUTE REQUEST"	$k_{rr} = 24$
L2RP "ROUTE REPLY"	$k_{rr} = 24$
L2RP Achtophorous Nodes	
Number of Achtophorous Nodes	$AN = 3$
Window interval for W2R	$window = 10$
LQI parameters	
Treshold for MinLQI	$LQI \geq 100$
Link Reliability (L2RP)	$LQI \geq 70$
Energy	
Initiale Energy Level	$E_0 = (1.404 * 10^5 - \varepsilon)\mu J$ $\varepsilon = random(0,1) * 10^2 \mu J$
Minimum Energy Level	$E_{min} = E_0 * 0.05$

In the simulation model N nodes are randomly (according to a uniform distribution) deployed over an area of length L=100m, and width l=100m (see Figure 9). The BS is located at the (0,0) position. Each node generates a sequence of "interesting events", which are sensed data over the temperature threshold T_{min}, following the Poisson process of parameter $\lambda = 10$. For simulation scenarios, the size of each data packet is set to $k_{data} = 128bits$, and the "ROUTE REQUEST" and "ROUTE REPLY" packets of the L2RP protocol have a size of $k_{rr} = 24bits$. Each node knows its position and its energy level. The initial energy amount of each node is set to $E_0 = (1.5 * 10^5 - \varepsilon)\mu J$, $\varepsilon = rand(0,1) * 10^2$. A node battery exhaustion is experienced when the remaining energy level of the node is under the given treshold $E_{min} = E_0 * 0.05$. All nodes, including the BS, have same transmission range (R = 20m). The main simulation parameters are listed in Table III.

C. LQI Model for Simulation Purposes

The WSN can be modelled as a graph $G = (V, E)$, where two nodes are connected by an edge if they can communicate with each other. Let $x \in V$ be a node in the WSN. $\mathcal{N}_1(x)$ is the neighbourhood of the node x. At each given time t, a node x forms with each $y \in \mathcal{N}_1(x)$ a link of which the link quality indicator (LQI) value is denoted by $\ell(x,y,t) > 0$. For all other nodes $z \in V \setminus \mathcal{N}_1(x)$, $\ell(x,z,t) = 0$. Let ν be a bijective function defined in V which is a totally ordered set. The ν function is defined as follows:

$$\forall\, x \in V,\ \ \nu(x) = (f(x), id(x)) \quad (9)$$

where $f(x)$ is the function which returns the metric value of x, and $id(x)$ returns the address of the node x. The total ordering in V is defined as follows:

$$\begin{array}{l} \forall x \in V,\ \ \nu(x) > \nu(y) \iff (f(x) > f(y)) \\ or \quad (f(x) = f(y) \quad and \quad id(x) > id(y)) \end{array} \quad (10)$$

After the WSN deployment in the warehouse, the BS initially broadcasts a message containing its position. This information is then retransmitted to all sensors in the network. In this phase, each node knows its degree of connectivity. At each given time t, the LQI value of the link formed by any pair (x,y) of nodes is calculated by using the $\ell(x,y,t)$ function defined below:

$$\ell(x,y,t) = f(x,y,t) * g(x,y) \quad (11)$$

$$f(x,y,t) = 1 - Pr\left[\ell ink(x,y,t) = Unreliable\right] \quad (12)$$

$$g(x,y) = \alpha + \frac{\beta * \log(1 + (\gamma(x,y) - \gamma_{min}(x)))}{\log(1 + \gamma_{max}(x))} \quad (13)$$

$$\gamma_(x,y) = \frac{1}{d(x,y)} \quad (14)$$

$$\gamma_{min}(x) = \min_{y \in \mathcal{N}_1(x)} \gamma(x,y) \quad (15)$$

$$\gamma_{max}(x) = \max_{y \in \mathcal{N}_1(x)} \gamma(x,y) \quad (16)$$

where $\alpha = 50$, $\beta = 255$ and $d(x,y)$ is the distance separating y from x.

In the context of a cold chain monitoring application, the warehouse hosts hundreds of pallets, one upon the other. Each pallets is provided with a temperature sensor. This environment is subjected to some unreliabilities of the wireless links. So, in the formula (12), $Pr\left[\ell ink(x,y,t) = Unreliable\right]$ denotes the probability that the link $\ell ink(x,y,t)$ becomes unreliable at time t. This probability is used in some simulation scenarios, in order to evaluate the behaviour of our L2RP protocol with respect to the unrelibility aspect of the wireless links.

The choice of this model, formula (13) similarly to the scale function Sc defined in the composite metric, is guided by experimental results shown in [39] and [10] which stated that the LQI decreases when the distance between nodes increases in Zigbee-based WSN.

As we can see, $\ell(x,y,t) \neq \ell(y,x,t)$, because of the formulas (15) and (16). Hence, the model allows to take into account asymetrical aspects of the wireless links.

For moteiv′s Tmote Sky [4] sensors equipped with chipcon′s CC2420 [7], the LQI values range from 50 to 110. Even so, we stick with the ZigBee standard [5][6] because some manufacturers, such as Sun-SPOT [35] and WiEye [36], are still using the standard LQI values. Then, we use the standard values (i.e. $[0, 255]$) increased by $\alpha = 50$, instead of those of CC2420. The use of $\alpha = 50$ allows to keep the null value, $\ell(x,y,t) = 0$, only for the two cases where the node y is not in the transmission range of the node x, or when the $\ell ink(x,y,t)$ becomes unreliable i.e. $Pr\left[\ell ink(x,y,t) = Unreliable\right] = 1$.

This LQI model is only used for simulation purposes, so sensor nodes do not compute these above formulas.

VIII. Simulation Results

Simulations, using Matlab, are run for a network size ranging from 100 to 500 nodes. The performance results presented here are obtained by averaging the results for 50 different simulations for each scenario comparing the route selection criteria. In each scenario where the three routing mechanisms are compared, 25 different simulations were run. For each simulation, a new random node layout is used.

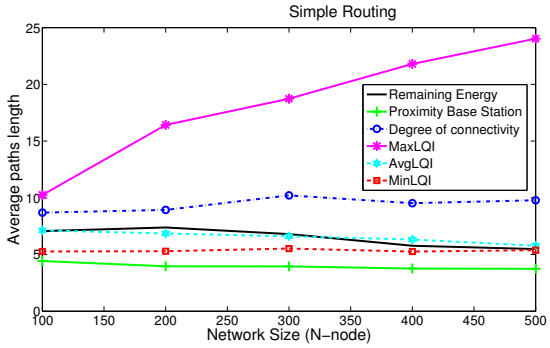
In all simulation results presented below, $\rho = 0.5$ for the composite metric as defined in formulas (2) and (3). If it's not specified, the number ANs of Achtophorous Nodes is set to $ANs = 3$ for each load balancing mechanism.

In all simulation scenarios, except those in Section VIII-H, links are considered reliable, i.e.:

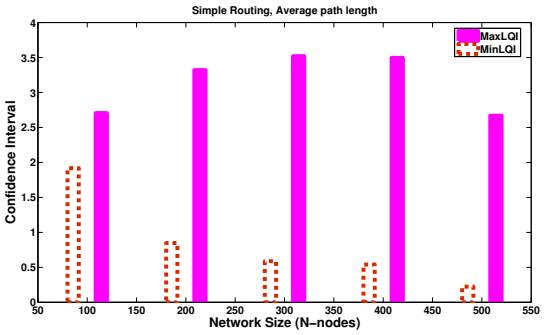
$$\forall t,\ \forall x \in V,\ Pr\left[\ell ink(x,y,t) = Unreliable\right] = 0\ , \forall y \in \mathcal{N}_1(x).$$

For some results, the related confidence intervals for a confidence coefficient of 95% are computed as detailed in the section 3.3 of [40].

A. Average Path Length



(a) The average path length (Simple routing)



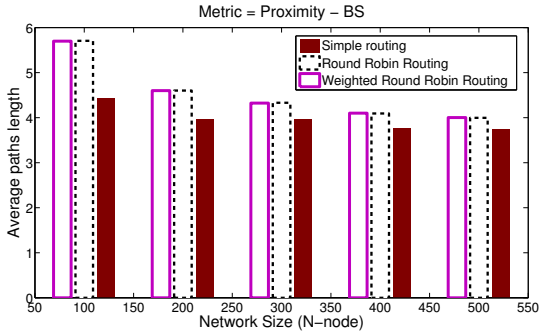
(b) Confidence Interval for MaxLQI and MinLQI metrics

Fig. 10. Average path length: Comparison of metrics in simple routing mechanism (a); and the related confidence interval for a confidence coefficient of 95% (b)

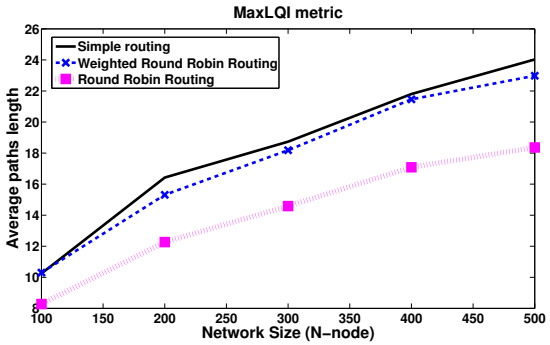
The Figure 10(a) shows the average path length for the simple routing; while the Figure 11(a) compares the average path length related to the "Proximity with respect to the BS" metric when it is used in the simple and load balancing mechanisms.

This result shows that routes are longer for MaxLQI and degree of connectivity metrics. The remaining energy, AvgLQI, MinLQI and "Proximity with respect to the BS" metrics have better average path lengths.

The Figure 10(a) shows, in the case of simple routing mechanism, the average path lengths in terms of the average number of hops obtained with the different studied metrics when the node density is increasing in the deployment area. This result shows that routes are longer for the MaxLQI and degree of connectivity metrics. The remaining energy, AvgLQI, MinLQI and "Proximity with respect to the BS" metrics have better average path lengths. The gap is more important for the MaxLQI metric with respect to the other metrics. Moreover, for MaxLQI, the average number of hops is a monotonically increasing function of the network density. This reflects the fact that the routing according to the metric



(a) The average path length (Proximity-BS)



(b) The average path length (MaxLQI)

Fig. 11. Average path length: Comparison of the three routing mechanisms with the Proximity-BS (a) and MaxLQI (b) metrics

MaxLQI consists of choosing as an achtophorous node the node having the best link quality with the source node. In the absence of obstacles and other phenomena like interferences, the best link quality is determined by the shortest distance separating a node from the source node. So, routing according to the MaxLQI metric is equivalent to a multihop "step by step" routing which is characterized by a great number of hops due to small distances separating each source node and its achtophorous node.

When the network density is increasing, the distances separating sensors decrease. Thus, the distances separating each source node and its achtophorous node also decrease, as well. So, from any source node towards the base station, the number of hops of each sent packet become increasingly high when the MaxLQI is used. By multiplying the number of hops, in this manner, the sensor network could not claim to have a good performance. This result explains the low performance of the MultiHopLQI routing algorithm which is used today in many TinyOS based empirical WSN analysis. Indeed, MultiHopLQI uses the LQI metric as defined in the ZigBee standard [5][6], that is to say the MaxLQI metric.

Conversely, the Proximity-BS and MinLQI metric have the lowest average path lengths (see Figure. 10(a)). For any given source node, the selected Proximity-BS based achtophorous node matches the farthest neighboring node towards the Base Station. Therefore, the routing according to the Proximity-BS metric is equivalent to the shortest geographical path routing. Accordingly, packets are transmitted from the source node to the base station requiring the minimum number of hops. This result (see Figure. 10(a)) is also interesting for the MinLQI metric. Indeed, this metric promotes the use of the links of intermediate quality. Links of good quality are synonymous with the nearest nodes multiplying the number of hops, whereas the links of poor quality stand for lot of packet losses. This explain why MinLQI is a good metric.

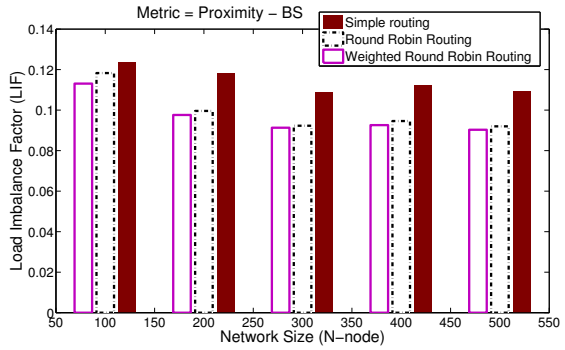
For the Proximity-BS metric, the load balancing mechanisms have the effect of increasing the average path lengths which is almost the same average for the weighted round robin routing and the round robin one (see Figure 11(a)). In the case of load balancing mechanisms, each sensor has in its routing table several achtophorous nodes of which only one exactly corresponds to the achtophorous node used by the simple routing. The other achtophorous nodes are necessarily more distant from the base station. So, the average path lengths slightly increase for load balancing mechanisms with respect to the simple routing using the Proximity-BS metric (see Figure 11(a)). For any given source node, the selected achtophorous nodes are identical for both load balancing mechanisms, their use only differs by the weight introduced in the weighted round robin routing. This leads to an average number of hops almost identical (see Figure 11(a)).

Unlike the Proximity-BS and MinLQI metrics , the MaxLQI one has an average path lengths which is reduced by the load balancing mechanisms (see Figure 11(b)). In this case, the weighted round robin routing mechanism has an average number of hops closer to the one of the simple routing mechanism than the round-robin one. Indeed in the case of W2R, the achtophorous node which forms the better link quality (MaxLQI) is also the one which has the highest weight. Thus, depending on the weight value, the sensors choose to send their packets more frequently to that achtophorous node. Therefore, W2R leads to an average number of hops closer to the one of the simple routing mechanism (see Figure 11(b)).

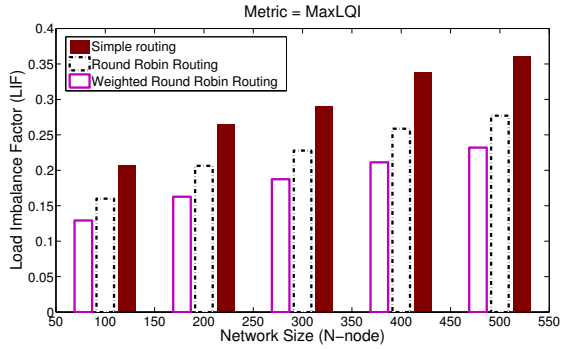
B. LIF: Load Imbalance Factor

The Figure 12(a) shows the LIF when the "Proximity with respect to the BS" is used as metric. It displays results for the simple routing and load balancing mechanisms. The Figure 12(b) for MaxLQI and the Figure 12(c) for MinLQI also display the LIF for the three routing mechanisms.

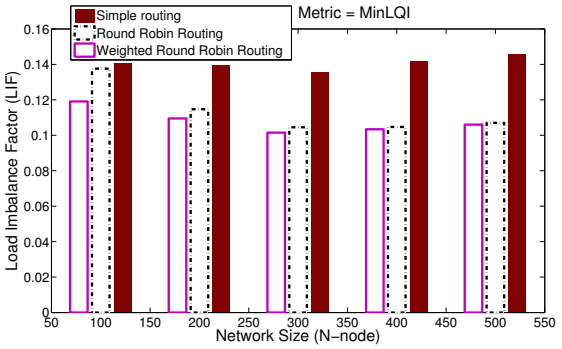
The lowest LIF value indicates the best evenly distribution of the energy consumption between nodes. It would be redundant to say that the load balancing mechanisms (round robin and W2R) help evenly balancing the load. That is to say that the average LIF values are lower for load balancing mechanisms compared to the simple routing, whatever the chosen metric (see Figure 12(a), 12(b) and 12(c)). But the gap is more important for MaxLQI than other metrics.



(a) Proximity-BS



(b) MaxLQI



(c) MinLQI

Fig. 12. Load Imbalance Factor: Proximity-BS (a), MaxLQI (b) and MinLQI (c)

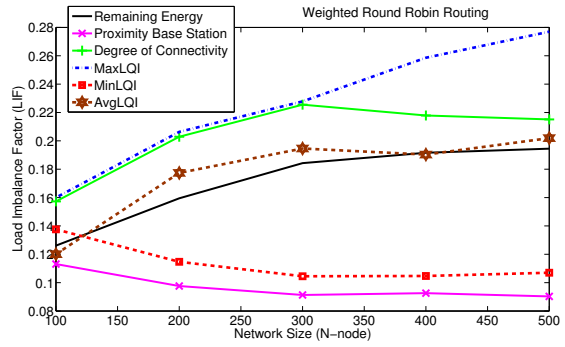


Fig. 13. Load Imbalance Factor: Comparison of different metrics in the Weighted Round Robin Routing, i.e. W2R Routing, mechanism

Moreover, when the network density is increasing, the difference between round robin and W2R tend to vanish for the Proximity-BS and for the MinLQI (see Figure 12(a) and 12(c)). For these two metrics, it would be more suitable in dense wireless sensor networks to use the round robin mechanism than the W2R one. Therefore, in doing so, one saves the power required, mainly by the processor, for the achtophorous weight computations (see Figure 7 and Table II).

These results confirm that load balancing mechanisms help in the distribution of the load across the nodes, because whatever the metric used: the W2R routing produces lower LIF than the round-robin routing which is followed by the simple routing (see Figure 12(a),12(b),12(c)).

As for the Figure 13, it compares the average LIF of different metrics in the W2R routing. The "Proximity with respect to the BS" and MinLQI metrics produce lower LIF values (see Figure 13). The remaining energy metric has an intermediate LIF, while the degree of connectivity and MaxLQI metrics tend to imbalance the energy consumption on the network: some sensors exhaust their batteries while others have a little participation in packet routings towards the BS. This negative phenomenon is much more important for the MaxLQI metric when the network size is increasing (see Figure 12(b) and Figure 13).

This reflects the fact that the degree of connectivity and MaxLQI metrics are the ones for which packets arrive at the Base Station by routes using the largest number of hops as shown in Figure 10(a) and explained in the last section. Thus, along each route, the WSN experiences more retransmissions and then more energy wastage due to the effects of overhead, latency and overhearing phenomena which are more important when the average number of hops is increasing.

C. Average Percentage of Packet Losses

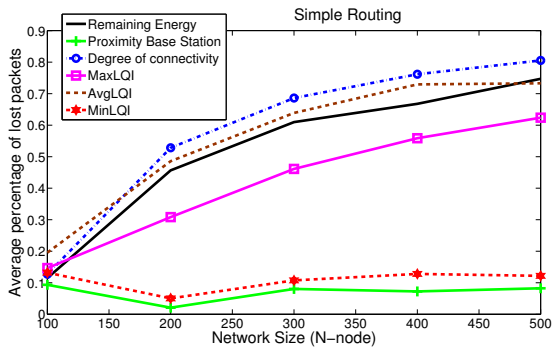


Fig. 14. Average percentage of lost packets: Comparison of the different metrics in the Simple Routing mechanism

The Figure 14 displays, for each metric, the average percentage of packet losses experienced by the network when the simple routing is run. The three routing mechanisms are

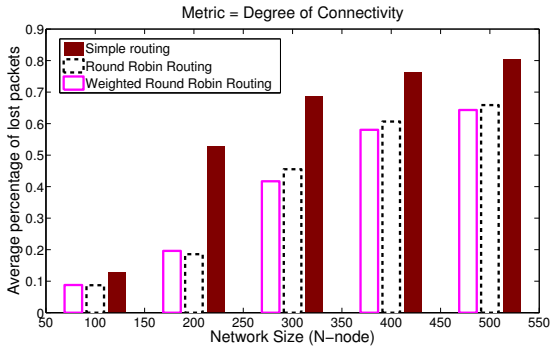


Fig. 15. Average percentage of lost packets: Comparison of the three routing mechanisms with the Degree of connectivity metric

compared (see Figure 15) using the degree of connectivity metric.

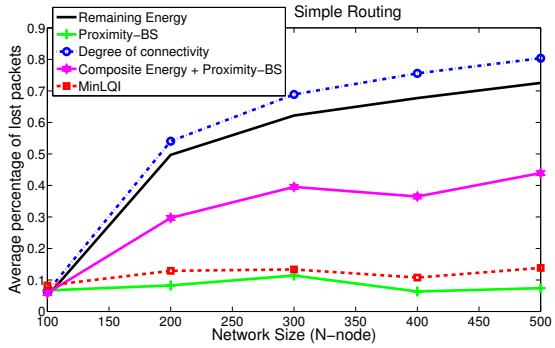
Generally, the loss percentage is quite low. This reflects the fact that, in L2RP, losses are mainly due to the node battery exhaustion. The first result (see Figure 14) compares the different criteria in the mechanism of simple routing. Here again, best results are produced by MinLQI and "Proximity with respect to the BS" metrics. MaxLQI has an intermediate average percentage of packet losses, while the remaining energy and degree of connectivity metrics have higher percentages. For the Proximity-BS metric this result is easy to understand, because according to the previous results (Figure. 10(a)), Proximity-BS is the metric which produces the shortest path lengths. Accordingly, as the overhearing phenomenon and the overhead induced by routing are reduced when the number of hops is minimal, then the node battery exhaustion occurs later (in time) leading to a low loss percentage for the metric Proximity-BS.

Conversely, the degree of connectivity metric has the highest percentage of packet losses (see Figure 14). By choosing to route packets according to this metric, any given sensor which has data to transmit chooses its achtophorous node, in simple routing, as its neighbor which has the highest number of neighbors. Therefore whenever an achtophorous node is requested to route a packet, the overhearing phenomenon causes more energy consumption which leads to a greater packet loss percentage.

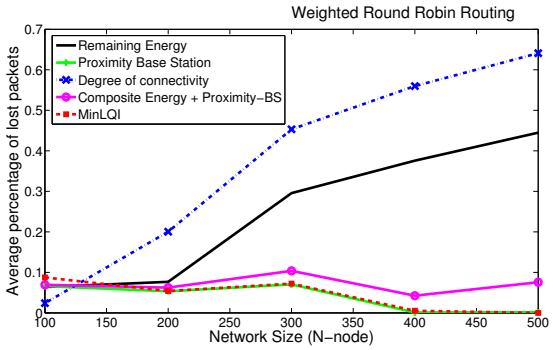
For all metrics, load balancing significantly reduces the average percentage of packet losses (see Figure 15). Load balancing mechanisms produce lower packet losses than the simple routing; differences are more important when load balancing is run with the degree of connectivity metric, the remaining energy metric or the MaxLQI metric.

Indeed, for the degree of connectivity metric of which the overhearing phenomenon is the most important, Load balancing requires the selection of different achtophorous nodes for each source node. So, in the routing table there is exactly one node which has the highest number of neighbors: this the one used by the simple routing mechanism. Then, in load balancing when the other achtophorous nodes with less neighbors are used, this helps reducing the overhearing phenomenon. This justifies why load balancing reduces the percentage of packet losses compared to simple routing which always requires the highest degree of connectivity as achtophorous node (see Figure 15).

D. Composite or Hybrid Metric



(a) Simple routing



(b) W2R Routing

Fig. 16. Average percentage of lost packets: Comparison of different metrics including the hybrid metric (remaining energy level + Proximity-BS) in Simple routing (a) and W2R Routing (b)

The Figure 16(a) (simple routing) and the Figure 16(b) (W2R routing) display the average percentage of packet losses including the hybrid metric which is a combination of the remaining energy metric and the "Proximity with respect to the BS" metric.

These results show that the hybrid metric composed of 50% of the remaining energy and 50% of "Proximity with respect to the BS" (i.e. $\rho = 0.5$ in Formula (3)) is a very good metric. It has a percentage of packet losses which is relatively low, especially when it is used with load balancing mechanisms. As we can see, there are fewer lost packets when the simple routing is run with the MinLQI metric than the W2R routing run with the remaining energy metric, MaxLQI or the degree of connectivity metric (see Figure 16(a) and Figure 16(b)).

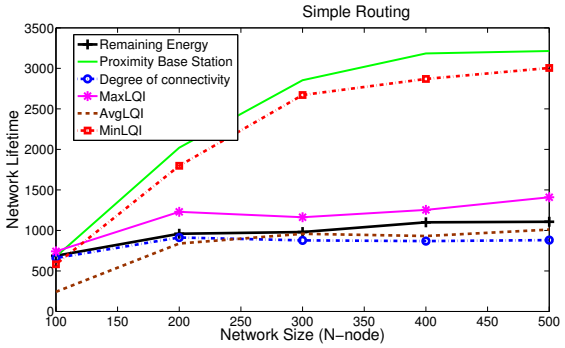
These results show that compared to the "remaining energy level" metric, the hybrid metric helps mitigating losses

particularly in load balancing (W2R Routing) where the average packet loss percentage is less than 0.1% for this metric.

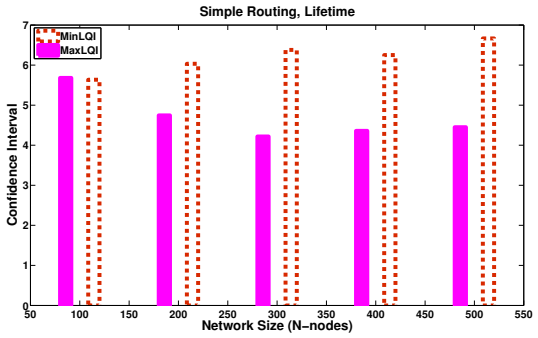
This kind of metric is very interesting to consider because depending on the specific WSN application purposes, it may be useful to consider several criteria for selecting routes by computing a single hybrid metric. In this result, it is more beneficial to route jointly depending on the distance and the remaining energy than to route only along with the remaining energy criterion. This reflects the fact that the "remaining energy level" criterion is not a good metric for route selection. Because in our simulation scenario (see Table III) each node is deployed with an initial energy level E_0 which is randomly and slightly lower than a reference value $E_0 = (1.404 * 10^5 - \varepsilon)\mu J$ with $\varepsilon = random(0,1) * 10^2 \mu J$. This scenario is very realistic, because even if the AA batteries powering the sensors are new, they also have slightly different energy levels in real world scenario.

Although the average percentage of packet losses is generally too low, the load balancing helps reducing the packet loss percentage for the hybrid metric similarly to all other studied metrics.

E. Average Network Lifetime

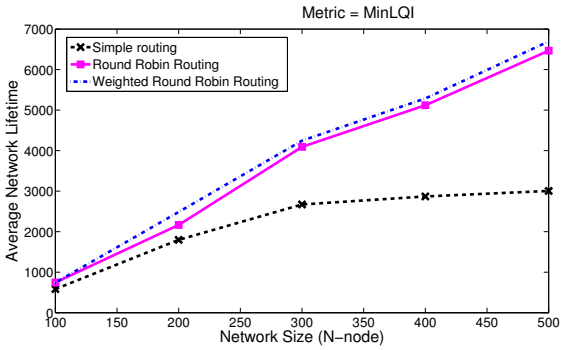


(a) Average network lifetime (Simple Routing)

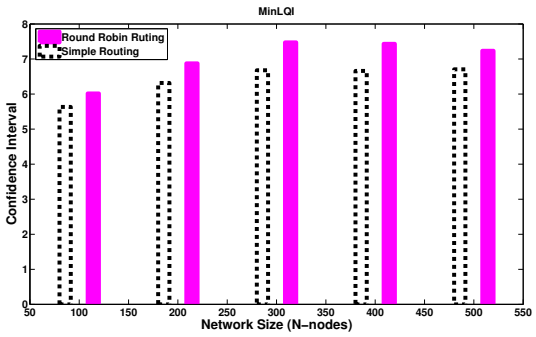


(b) Confidence Interval for MaxLQI and MinLQI metrics

Fig. 17. Average network lifetime: Comparison of different metrics in the simple routing mechanism (a); and the related confidence interval for a confidence coefficient of 95% (b)



(a) Average network lifetime (MinLQI)



(b) Confidence Interval for simple and round robin routings

Fig. 18. Average network lifetime: Comparison of the three routing mechanisms using the MinLQI metric (a); and the related confidence interval for a confidence coefficient of 95% (b)

The Figure 17(a) displays the average network lifetime for the simple routing. The Figure 18(a) shows the average network lifetime when MinLQI is used in each routing mechanism.

Firstly, these results show that more dense networks have better lifetime. The MinLQI and "Proximity with respect to the BS" metrics produce better network lifetime. MaxLQI is better than the remaining energy metric which is followed by the degree of connectivity metric (see Figure 17(a)). Load balancing mechanisms significantly increase the average network lifetime which is more large than the one of the simple routing with more differences for MinLQI (see Figure 18(a)) and "Proximity with respect to the BS" metrics.

The time of first packet loss occurs earlier for the degree of connectivity metric. As we explained in the previous sections, this result is also caused by the overhearing phenomenon of which effects are more important for the degree of connectivity metric with respect to other metrics. The Proximity-BS metric improves the network lifetime by minimizing the number of hops (see Figure 17(a)).

Compared to the simple routing, the load balancing mechanisms (see Figure 18(a)) significantly increase the WSN lifetime. However, even if the weighted round robin routing leads to a better WSN lifetime than the round robin routing,

the gap between the two load balancing mechanisms is not significant for the MinLQI metric (see Figure 18(a)).

By rotating the achtophorous node this helps splitting the load among different sensors. So, load balancing helps delaying the moment of the first node battery depletion of and therefore extending the lifetime of the network: Load balancing adds lifetime benefits to the WSN.

F. Average Ratio of the Remaining Energy

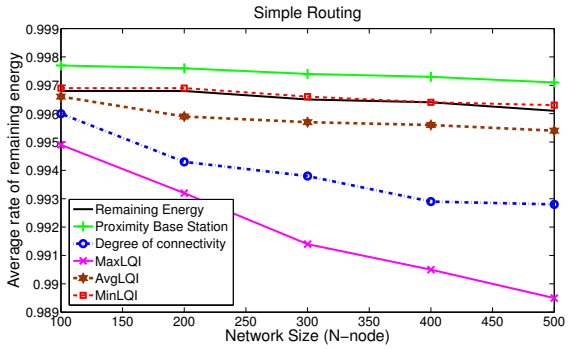


Fig. 19. The average ratio of the remaining energy: Comparison of the different metrics in the Simple Routing mechanism, after one cycle of which all sensors had sent their alarms towards the Base Station.

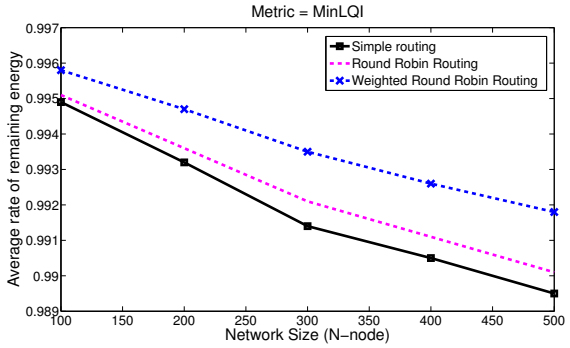


Fig. 20. The average ratio of the remaining energy: Comparison of the three routing mechanism with the MaxLQI metric, after one cycle of which all sensors had sent their alarms towards the Base Station.

The figures (see Figure 19 and Figure 20) show, depending on network density, the evolution of the average remaining energy after a complete cycle. The cycle is constituted by: the network deployment, the detection of alarms and the data routing towards the base station where each source node uses L2RP to build its routing table. The cycle ends when all nodes have sent their alarms.

The degree of connectivity and MaxLQI metrics are the least energy efficient metrics (see Figure 19). In contrast, Proximity-BS and MinLQI are the metrics that ensure better energy efficiency.

The weighted round-robin routing (W2R) leads to less energy consumption than the round-robin routing which is better than the simple routing whatever the metrics used. The Figure 20 shows the result for the MaxLQI metric.

In summary, these results are natural consequences of the previous ones. Indeed, for the MaxLQI metric of which the average number of hops (average path length) is high, the energy consumption is also large because of the increasingly overhearing, latency, and overhead phenomena.

G. Impacts of Increasing the Number of Achtophorous Nodes

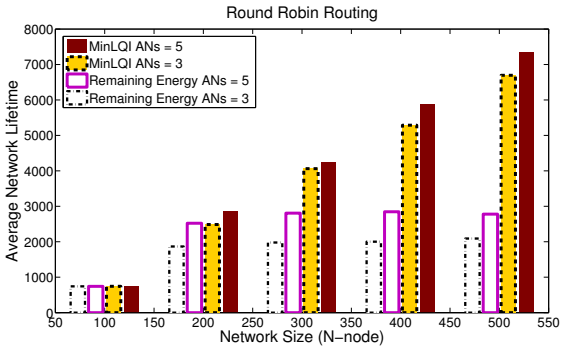


Fig. 21. Impacts of increasing the number of achtophorous nodes on the average network lifetime for the round-robin routing mechanism.

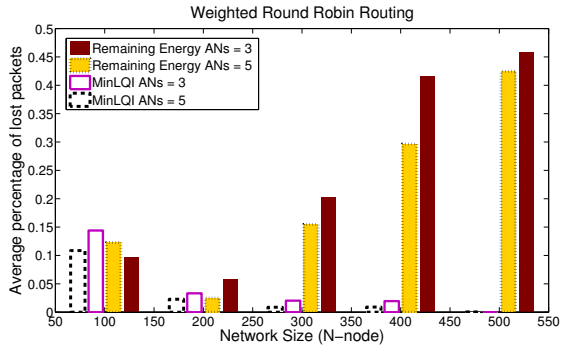


Fig. 22. Impacts of increasing the number of achtophorous nodes on the average percentage of packet losses for the W2R Routing mechanism

The Figure 21 shows the influence of the number (ANs) of the achtophorous nodes on the network lifetime performance criterion by comparing the results for ANs = 3 and ANs = 5, when the remaining energy and MinLQI metrics are combined with the round-robin routing.

The Figure 22 shows the influence of increasing the number (ANs) of the achtophorous nodes on the average percentage of lost packets by comparing results for ANs = 3 and ANs = 5, when the W2R routing is run with the remaining energy and MinLQI metrics.

These two results (see Figure 21 and Figure 22) show that the average percentage of lost packets decreases for the MinLQI metric. The network lifetime increases for both metrics when the number of achtophorous nodes varies from

3 to 5. This is not obvious to predict, because increasing the number of achtophorous nodes might increase the risk of using low-energy sensors in routing process, which could cause more packet losses.

From a given number of achtophorous nodes, the result should be reversed. Nevertheless, until the value $AN = 5$, it remains within reasonable limits for a cold chain monitoring application.

H. Impacts of the Unreliability of Wireless Links

In the context of our application, the warehouse hosts hundreds of pallets, one upon the other. Each pallets is provided with a temperature sensor. This environment is subjected to some unreliabilities of the wireless links. In this section we take into account such a phenomenon. At any given time t, for a sensor S_i, its unreliable links ($Pr\left[\ell(i,j,t) = Unreliable\right] = 1$ in Formula (12)) with some neighbors are modeled by the Poisson process of parameter $\gamma(S_i,t)$ calculated as follows:

$$\gamma(S_i,t) = \frac{\mu}{\delta(S_i)} \tag{17}$$

where $\delta(S_i)$ is the number of nodes located between the node S_i and the BS. If $\delta(S_i) = 0$, then the node S_i has no elligible achtophorous node.

At any given time t, for each sensor S_i, $\gamma(S_i,t)$ is too small, then the Poisson process returns a series $\mathcal{T}_i$ of integers $\mathcal{T}_i$, in which nonzero values $\mathcal{T}_i[j]$ denote the unreliable links formed by S_i with some of its neighbours S_j, i.e. $Pr\left[\ell(i,j,t) = Unreliable\right] = 1$ in Formula (12).

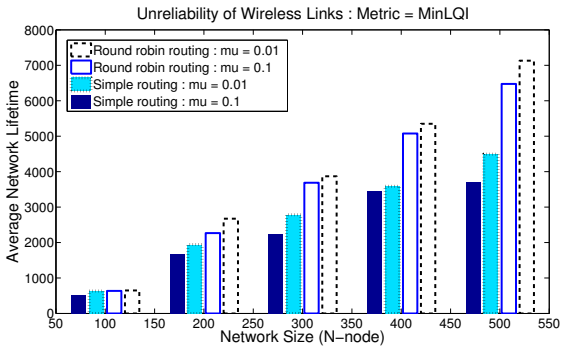


Fig. 23. Impacts of the unreliability of the wireless links on the average network lifetime (MinLQI, $\mu = 0.01$ and $\mu = 0.1$) for both simple and round robin routing mechanisms

The Figure 23 shows the effect of the unreliabilities of the wireless links on the WSN lifetime by comparing results for $\mu = 0.01$ (low unreliability) and $\mu = 0.1$ (high unreliability), when the MinLQI metric is used in the simple routing and in the round-robin routing. The Figure 24 (resp. the Figure 25) shows impacts on the average path length (resp. on the LIF) by comparing results for $\mu = 0.1$ (high unreliability), when MinLQI metric is used in the three routing mechanisms.

The first result in Figure 23, shows that the network lifetime is smaller in high unreliable WSN ($\mu = 0.1$). In this

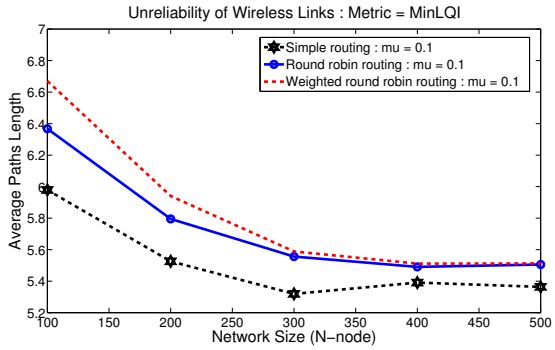


Fig. 24. Impacts of the unreliability of the wireless links on the average path length (MinLQI, $\mu = 0.1$) for the three routing mechanisms

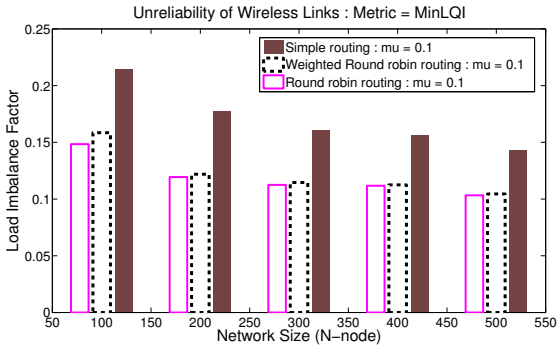


Fig. 25. Impacts of the unreliability of the wireless links on the Load Imbalance Factor (MinLQI, $\mu = 0.1$) for both simple and load balancing routings.

case, the load balancing also increases the network lifetime. Indeed, the round-robin routing in high unreliable WSN ($\mu = 0.1$) is much better than the simple routing in low unreliable links environment ($\mu = 0.01$), even if the simple routing produces lower average path length (see Figure 24) than load balancing mechanisms. Even in the context of high unreliable links, the load balancing routing produces better LIF than the simple routing (see Figure 25), which means that the load is more evenly shared between nodes.

This result pertained to the MinLQI metric, clearly shows that the unreliabilities of the wireless links phenomena reduce the WSN lifetime because of more retransmissions needed in such an environment. But the key point of this result relies on the fact that load balancing mechanisms also add lifetime benefits in high unreliable environment. Indeed, in the case of simple routing, a weak link between a sensor and its achtophorous node involves the sending of a new "ROUTE REQUEST" message. In contrast, for load balancing mechanisms, each sensor has several achtophorous nodes in its routing table. If a link between a sensor and its achtophorous node were to be unreliable, the source node first examines the quality of the link it forms with the next achtophorous node listed in its routing table. So, if this link is reliable, it simply sends the packet without having to request a new route.

Therefore, in load balancing mechanisms, a source node has to send a new "ROUTE REQUEST" message, if and only if all the links it forms with all the achtophorous nodes listed in its routing table were to become unreliable at the same time.

In the scenarios which do not take into account the unreliabilities of the wireless links, the L2RP protocol leads to identical routes for the two load balancing mechanisms (see Figure 11(a)). The Figure 24 shows the impacts of the unreliabilities of the wireless links on the average path lengths (number of hops) in an environment subjected to high unreliable links ($\mu = 0.1$). In this result, we observe that the routes obtained with the round robin mechanism are now different from those obtained with the weighted round robin routing (W2R) (see Figure 24). This is due by the fact that link quality parameters are very fickle and time variant and are greatly dependent on the poisson parameter $\gamma(S_i, t)$ (see Formula 17).

The Figure 25 plots, for $\mu = 0.1$, the impacts of high unreliabilities of the wireless links on the LIF criterion performance. This still confirms the effectiveness of the load balancing mechanisms in unreliable environments. Indeed, the load imbalance factor (LIF) is lower for the round robin and W2R routing compared to the simple routing. Contrary to the previous result (see Figure 12(c)), this one (see Figure 25) shows that the gap between the average LIF of the simple routing mechanism and those obtained via the load balancing routings decreases as the network density is increasing. Indeed, the unreliabilities of the wireless links become more and more important when the density of the WSN is increasing. Consequently, load balancing mechanisms gradually begin to loose some of their interest.

IX. Conclusion

In this paper, we have proposed the L2RP routing protocol (Link Reliability based Routing Protocol) which takes into account the quality of the links formed by any source node with the achtophorous nodes listed in its routing table. This avoids sending data over a link disrupted, unreliable or unstable.

The L2RP protocol also includes load balancing mechanisms where the source node, based on "ROUTE REPLY" packets, is able to estimate the load sustainable by each of its achtophorous node. This property allows L2RP to avoid doing a per packet load-balancing by the source, as done in [34], where the source node sends its data without being sure of the capacity of the achtophorous node to sustain the load assigned. Thus, by doing so, L2RP helps to reduce packet losses.

Applications often have their specific objectives and constraints, so it is essential to have the choice between several possible settings when deploying a wireless sensor networks. Thus, in its design, the L2RP protocol can use any chosen metric. This allows L2RP to be able to support different applications by offering the choice of the metric which ensures the best performance in the specific context of each application.

We therefore evaluated the L2RP performance based on routing mechanisms (simple or load balancing) and then presented a comparative study of the different metrics in each routing mechanisms. This work has shown that:

- The degree of connectivity metric is the metric that leads to the highest percentage of packet losses. This metric also has the lowest network lifetime. Indeed, it is the metric which is the most sensitive to the overhearing phenomenon.

- The Proximity-BS metric provides better energy efficiency. With this metric, the alarms sent by any sensor reach the Base Station in less hops. By minimizing the number of hops, it helps in reducing energy wastefulness due to overhearing, overhead and latency.

- The LQI used as a metric by considering the best link quality (the MaxLQI metric) leads to an inefficient routing regardless of the performance criterion considered. This confirms our previous experimental results obtained in [3]. The MaxLQI metric matches the standard definition of the LQI used in the MultiHopLQI routing algorithm [8]. Indeed, this metric is characterized by a relatively high average number of hops. In the absence of obstacles and any interferences, the best link quality is often observed for the nodes which are located relatively close to each other. By multiplying the number of hops, the MaxLQI metric has the effect to increase energy wastefulness due to overhearing, overhead and latency.

- Accordingly, despite its popularity in WSN empirical analysis based on TinyOS platforms, the MultiHopLQI routing algorithm is not suitable for WSN applications, because it uses the MaxLQI metric for route selection.

- By setting a given LQI threshold, i.e. a value of acceptable LQI, and considering the lowest LQI value beyond this threshold (the MinLQI metric), we obtain an optimal LQI based metric which highly enhances the energy efficiency. As the LQI decreases when the distance between the nodes increases, the average path length is larger for MaxLQI than for MinLQI: this explains why MinLQI is more energy-efficient than MaxLQI. Then, the average percentage of packet losses is larger for MaxLQI. There is a trade-off between routes consisting of good links quality and small average path length (i.e without too many retransmissions).

- This interesting result shows that it is better for LQI based routing algorithm to promote links of intermediate quality

(such as MinLQI metric) to avoid:

  - better links which are synonymous of nodes located relatively close to each other and also synonymous of higher number of hops which are responsible for excessive energy consumption;

  - bad links (low quality) which are synonymous of higher percentage of packet losses.

- The load balancing mechanisms significantly improve the routing efficiency by extending the network lifetime, while minimizing the average percentage of packet losses. The load balancing also helps evenly splitting the load on all nodes in the WSN.

- Increasing the number of achtophorous nodes improves the network performance: a low average of packet losses and a longer network lifetime.

- The composite metric, resulting of the remaining energy metric combined with the Proximity-BS metric, offers good routing performance. This metric is interesting, as each node ignores the settings of its neighbors (such as the remaining energy, the position) when selecting its achtophorous nodes.

- Since it is LQI based routing algorithm, the question that naturally arises is how L2RP behave in an environment subjected to high unrelibilities of the wireless links. Simulation results has shown that, such an environment slightly impacts the L2RP efficiency. Generally, packet loss percentage is relatively low because in L2RP a source node avoids sending data to an achtophorous node with which it forms an unreliable link at the moment it has data to transmit.

Embedded with load balancing mechanisms, L2RP adds lifetime benefits to the wireless sensor network. Nevertheless, it would be more profitable to combine L2RP with aggregation techniques like cluster formation and data aggregation in order to gain more scalability and lifetime. So, in [41] we used L2RP in a cold chain monitoring application where regular sensors send alarms to their respective clusterheads which aggregate received alarms and then forward the aggregated data towards the BS using the L2RP routing protocol. In this application L2RP is run with the weighted round robin load balancing mechanism using the "MinLQI" metric.

REFERENCES

[1] C. Diallo, M. Marot, and M. Becker. Link quality and local load balancing routing mechanisms in wireless sensor networks. In *Proc. of the 6th Advanced International Conference on Telecommunications, AICT 2010*, Barcelona, Spain, May 2010.

[2] C. Diallo, A. Gupta, M. Becker, and M. Marot. Energy aware database updating protocols for autoconfigurable sensor networks. In *GlobeNet 2009, the 8th international conference on Networks, ICN'09*, Cancun, Mexico, Mar. 2009.

[3] A. Gupta, C. Diallo, M. Marot, and M. Becker. Understanding topology challenges in the implementation of wireless sensor network for cold chain. In *Proc. IEEE Radio and Wireless Symposium, RWS'10*, New Orleans, LA, USA, 2010.

[4] Tmote Sky datasheet. http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf.

[5] IEEE Std 802.15.4-2006. Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (wpans). In *IEEE Computer Society*, 2006.

[6] Zigbee specification. Zigbee specification v1. June 2005.

[7] CC2420 Radio. http://www.chipcon.com. Last access, Mar. 2010.

[8] J. Polastre, J. Hui, J.Z.P. Levis, D. Culler, S. Shenker, and I. Stoica. A unifying link abstraction for wireless sensor networks. In *SenSys*, 2005.

[9] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. *1st International Conference on Embedded Networked Sensor Systems, SenSys'03, Los Angeles, CA, USA*, 2003.

[10] M. Becker, A.-L. Beylot, R. Dhaou, A. Gupta, R. Kacimi, and M. Marot. Experimental study: Link quality and deployment issues in wireless sensor networks. In *Proc. NETWORKING'09, LNCS 5550*, pages 14–25, NETWORKING , Aachen, Germany, 2009.

[11] D. Puccinelli and M. Haenggi. Lifetime benefits through load balancing in homogeneous sensor networks. *IEEE Wireless Communications and Networking Conference, WCNC'09, Budapest, Hungary*, April 2009.

[12] D. Puccinelli and M. Haenggi. Arbutus: Network-layer load balancing for wireless sensor networks. *IEEE Wireless Communications and Networking Conference, WCNC'08, Las Vegas, NV, USA*, March 2008.

[13] D. Lal, A. Manjeshwar, F. Herrmann, E. Uysal-Biyikoglu, and A. Keshavarzian. Measurement and characterization of link quality metrics in energy constrained wireless sensor networks. In *Proc. IEEE Globecom 03*, San Francisco, USA, 2003.

[14] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Proc. ACM Sensys'03*, CA, USA, 2003.

[15] D. Son, B. Krishnamachari, and J. Heidemann. Experimental analysis of concurrent packet transmissions in low-power wireless networks. In *Proc. ACM Sensys'06*, Colorado, USA, 2006.

[16] G. Zhou, T. He, J. Stankovic, and T. Abdelzaher. Rid: Radio interference detection in wireless sensor networks. In *Proc. IEEE Infocom 05*, Miami, USA, 2005.

[17] S. Singh, M. Woo, and C. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proc. ACM Mobicom'98*, Dallas, Texas, USA, 1998.

[18] K. Scott and N. Bamboos. Routing and channel assignment for low power transmission in pcs. In *Proc. of ICUPC'96*, Cambridge, USA, 1996.

[19] R. Shah and J. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'02)*, Orlando, Florida, USA, March 2002.

[20] J. Chang and L. Tassiulas. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12:609619.

[21] F. Othman, N. Bouabdallah, and R. Boutaba. Load-balanced routing scheme for energy-efficient wireless sensor networks. In *IEEE*

Globecom 08, New Orleans, LA USA, 2008.

[22] S. Toumpis and S. Gitzenis. Load balancing in wireless sensor networks using kirchhoff's voltage law. In *IEEE infocom 09*, Rio de Janeiro, Brazil, 2009.

[23] J. Gao and L. Zhang. Load balanced short path routing in wireless networks. In *IEEE Infocom 04*, Hong Kong, China, 2004.

[24] Z. Wang, E. Bulut, and B.K. Szymanski. Energy efficient collision aware multipath routing for wireless sensor networks. In *IEEE ICC'09*, Dresden, Germany, 2009.

[25] L. Popa, C. Raiciu, I. Stoica, and D.S. Rosenblum. Reducing congestion effects by multipath routing in wireless networks. In *Proc. of the 14th IEEE International Conference on Network Protocols, ICNP'06*, pages 96–105, Santa Barbara, USA, 2006.

[26] C. Wu, R. Yuan, and H. Zhou. A novel load balanced and lifetime maximization routing protocol in wireless sensor networks. In *Proc. IEEE Vehicular Technology Conference (VTC) Spring*, pages pp. 113–117, Singapore, 2008.

[27] K. Sha, J. Du, and W. Shi. Wear: A balanced, fault-tolerant, energy-aware routing protocol for wireless sensor networks. *International Journal of Sensor Networks*, 1((3/4)):156–168, 2006.

[28] I. Raicu, L. Schwiebert, S. Fowler, and S.K.S. Gupta. Local load balancing for globally efficient routing in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 1:163185, 2005.

[29] R. Vidhyapriya and P.T. Vanathi. Energy efficient adaptive multipath routing for wireless sensor networks. *IAENG International Journal of Computer Science*, 34:1(IJCS-34-1-8), 2006.

[30] V.C. Gungor, C. Sastry, Z. Song, and R. Integlia. Ressource-aware and link quality based routing metric for wireless sensor and actor networks. In *Proc. IEEE International Conference on Communications, ICC'07*, Glasgow, Scotland, 2007.

[31] S. Hussain and A. W. Matin. Hierarchical cluster-based routing in wireless sensor networks. In *Proc. IEEE/ACM International Conference on Information Processing in Sensor Networks (IPSN)*, Nashville, TN, USA, 2006.

[32] D. Nam and H. Min. An energy-efficient clustering using a round-robin method in a wireless sensor network. In *Proc. of the 5th ACIS International Conference on Software Engineering Research, Management & Applications, SERA'07*, pages 54–60, 2007.

[33] D. Choi, J. Shen, S. Moh, and I. Chung. Virtual cluster routing protocol for wireless sensor networks. In *Proc. (641) Parallel and Distributed Computing and Networks, PDCN2009*, Innsbruck, Austria, 2009.

[34] M.O. Rashid, M.M. Alam, A. Razzaque, and C.S. Hong. Reliable event detection and congestion avoidance in wireless sensor networks. In *Proc. of High Performance Computing Conference, HPCC'07, LNCS 4782*, page 521 532, Houston, Texas, USA, 2007.

[35] Sun SPOT World. http://www.sunspotworld.com. Last access, Mar. 2010.

[36] EasySen WiEye Sensor Board. http://www.easysen.com/wieye.htm. Last access, Mar. 2010.

[37] W.B. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4):660–670, October 2002.

[38] C. Diallo, A. Gupta, M. Marot, and M. Becker. Virtual base station election for wireless sensor networks. In *ACM Notere 2008, the 8th international conference on New Technologies in Distributed Systems, Vol. 2*, Lyon, France, Jun. 2008.

[39] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann. Weighted centroid localization in zigbee-based sensor networks. In *IEEE International Symposium on Intelligent Signal Processing, WISP'07*, 2007.

[40] M. Becker and A.L. Beylot. Simulation des réseaux. In *Traité IC2, Série Réseaux et Télécoms*, Hermes, 2006.

[41] C. Diallo, M. Marot, and M. Becker. Single-node cluster reduction in wsn and energy-efficiency during cluster formation. In *Proc. of the 9th IFIP Annual Mediterranean Ad Hoc Networking Workshop, Med-Hoc-Net 2010, IEEE Communications Society*, Juan-Les-Pins, France, June. 2010.

Evaluation of Distributed SOAP and RESTful Mobile Web Services

Feda AlShahwan
Centre for Communications Systems Research
University of Surrey
Surrey, UK
F. AlShahwan@surrey.ac.uk

Klaus Moessner
Centre for Communications Systems Research
University of Surrey
Surrey, UK
K. Moessner@surrey.ac.uk

Francois Carrez
Centre for Communications Systems Research
University of Surrey
Surrey, UK
F.Carrez @surrey.ac.uk

***Abstract*— Even mobile Web Services are still provided using servers that usually reside in the core networks. Main reason for not providing large and complex Web Services from resource limited mobile devices is not only the volatility of wireless connections and mobility of mobile hosts, but also, the often limited processing power. Offloading of some of the processing tasks is one step towards achieving optimal mobile Web Service provision. This paper presents two frameworks for providing distributed mobile Web Services: One mobile service provision framework is built on *Simple Object Access Protocol* (SOAP), while the other implements *Representational State Transfer* (REST) architecture. Both frameworks have been extended with offloading functionality and different types of resource intensive operations, i.e., process intensive and bandwidth intensive services, have been tested. The results show that using a REST-based framework leads of a better performing offloading behaviour, compared to SOAP-based mobile services. Distributed mobile services based on REST consume fewer resources and achieve better performance compared to SOAP based mobile services. The paper describes the approach, evaluation method and findings.**

Keywords-Mobile Web Services; REST; SOAP; Service Distribution.

I. INTRODUCTION AND MOTIVATION

Mobile Web services are self-contained modular applications that are defined, published and accessed across the Internet using standard protocols in a mobile communications environment. This technology has evolved from advances in the mobile device technology, rapid growth of Web Services development and progression of wireless communication in parallel with widespread use of Internet applications. However, it is still in its early stages and there are many challenges to overcome. Those challenges result from constraints in mobile resources, mobility issues and intermittent wireless network.

In literature, three different types of Mobile Web-services have been explored; they are characterized by the role acted by the mobile device when providing or consuming Web Services (see Fig. 1). These types include: **(Mobile) consumer, provider, and P2P Web Services**. In the mobile

Figure 1: Classification of Mobile Web Services

Web-service consumer case, mobile devices act as clients and request a service. In the provider case, mobile devices act as servers and provide them to any type of client. In the P2P case, mobile devices are connected in Ad hoc manner and each node may act as client or server, or both.

Most research into mobile Web Services has focused on consuming standard Web Services from mobile devices. However, the ubiquitous availability of mobile devices and their capability to provide information (e.g., Sensing information), or to provide complete/integrated services is a viable proposition. Hence, there is a need of exploring the provisioning of Web Services from mobile hosts. Our previous work [1] has investigated providing Web Services from mobile devices.

Hosting Web Services from mobile devices has an enormous number of useful real life applications. Location-based applications are an example of these useful applications. Location-based Web Services can be provided from mobile devices and have shown performance enhancement to companies who have employees deployed in the field. For example, a *Mobile Host* (MH) with a built-in *Global Positioning System* (GPS) receiver allows tracking of products and goods [2]. Health care applications are further evidence of the kind of applications provided by hosting Web Services from mobile devices. They might be useful for both doctors as well as patients. For example, deploying an appropriate service on a doctor's mobile allows tracking professionals' location and context to handle

emergency cases. Health care services can also be extended and provided from patients' mobile devices. This takes place by exposing a remote tele-monitoring service on the patient's MH [3] that allows monitoring their conditions using log files with the aid of some measurement devices such as a *Body Area Network* (BAN) sensor suite [4]. Not all location-based applications can be provided from the conventional fixed servers. This is because providing any location-based service is highly dependent on the actual current location of the service provider. For instance, providing the latest updated news and scene snapshots for a specific location in a predefined format requires portable devices with built-in GPS and cameras that are capable to move to the actual location of the event. Furthermore, it requires MHs that are aware of their location to publish the event as a live feed and takes latest information gathered at the current location. MHs allow processing of the gathered information and can then make it available, instantly to clients. Consequently, for the server, it may be more efficient in terms of cost and performance since it eliminates the need to upload the gathered location dependent information to static web server. Mobile devices are ubiquitous; they have small form factors, portable and almost anywhere accessible. As such, managing and maintaining handheld mobile hosts is easier, faster and more portable than static terminals. Moreover, mobile Web Services can be useful in polling-based applications that require using and triggering the most recent data, which is changing dynamically. Since checking an updated Really Simple Syndication (RSS) feeds through polling scheme requires exchanging a significant amount of information between each client and the standard fixed server. However, if the web server is a mobile device then the polling scheme is eliminated and substituted by sending a message from mobile host to all mobile clients when an update occurs. Context-based applications constitute another application discipline that benefits from hosting Web Services from mobile devices. Accessing the user profile of the mobile host and sharing the contents with others could be a useful application that allows clients to access the mobile host data contents, pictures and share the profile or modify it. The owner can also use web user interface of his mobile using standard desktop or laptop to get messages, information about incoming phone calls and phone book log when mobile host is currently unavailable or a better interface is required for accessing mobile contents. However, there are some issues related to the internal and external resource limitations of mobile hosts see Table1 that act as a barrier against the easy development of this area.

The motivation that leads towards this research is the large number of useful applications that can be provided from hosting services on resource constrained mobile devices. However, there are clear limitations in terms of complexity and size of the services that may be executed on mobile host.

TABLE 1. Internal and External mobile Constraints

Internal Constraints	External Constraints
• **Memory capacity, processing power and short battery life** • **Some data types that are defined with web services are not supported by the mobile devices** • **Most mobile devices support only short range wireless communication.**	• **Heterogeneity of the wireless environment** • **Limited bandwidth and large communication delay.** • **Frequent context and location change of mobile host** • **Mobile devices continuously need static IP address**

Our goal is to allow providing large and complex mobile Web Services continuously and without interfering with the main functionality of the mobile host that is making phone calls. Thus, lightweight processing and provisioning of mobile Web Services is needed to compensate for the limited resources of mobile hosts. This can be achieved through supporting automatic and autonomous self configuring distributed systems.

The technology used for developing Web Services can be classified into two main categories: *Representational State Transfer* (RESTful) and *Simple Object Access Protocol* (SOAP) Web Services. This classification is based on the architectural style used in the implementation technology. SOAP is an object-oriented technology that defines a standard protocol used for exchanging XML-based messages. It is defined as protocol specification for exchanging structured information in the implementation of Web Services in computer networks [5]. The specification defines an XML-based envelope for exchanging messages and the protocol defines a set of rules for converting platform specific data types into XML representations. REST is a resource oriented technology and it is defined by Fielding in [6] as an architectural style that consists of a set of design criteria that define the proper way for using web standards such as HTTP and URIs. Although REST is originally defined in the context of the Web, it is becoming a common implementation technology for developing Web Services. RESTful Web Services are implemented with Web standards (HTTP, XML and URI) and REST principles. REST principles include addressability, uniformity, connectivity and stateless. RESTful Web Services are based on uniform interface used to define specific operations that are operated on URL resources. Both SOAP and REST are used for implementing Web Services. However, each has its own distinct features and

shortcomings that make it more or less suitable for certain types of applications as shown in Table 2.

This paper is an extended version of [1] . It focuses on investigating mechanisms that facilitate distribution of provisioning and executing mobile Web Services. This can be accomplished through extending our previous SOAP- and REST-based *Mobile Host Web Service Frameworks* (MHWFs) that were implemented to deploy, execute and provide mobile Web Services. Our original implementation is extended in this paper to allow offloading of services and service fragments. In addition, this paper evaluates the performance and offloading overhead for both SOAP- and RESTful-based frameworks. This evaluation assists in selecting the framework that best suits mobile environment capabilities and fulfils our goal to provide mobile Web Services continuously with a light-weight processing requirement.

TABLE 2. Comparison of SOAP/ RESTful-based Web Services

Criteria	SOAP-based WS	RESTful-based WS
Server/ Client	Tightly coupled	Loosely coupled
URI	One URI representing the service endpoint	URI for each resource
Transport Layer Support	All	Only HTTP
Caching	Not Supported	Supported
Interface	Non Uniform Interface (WSDL)	Uniform Interface
Context aware	Client context aware of WS behaviour	Implicit Web Service behaviour
Data Types	Binary requires attachment parsing	Supports all data types directly
Method Information	Body Entity of HTTP	HTTP Method
Data Information	Body Entity of HTTP	HTTP URI
Describing Web Services	WSDL	WADL
Expandability	Not Expandable (No hyperlinks)	Expandable without creating new WS (using xlink)
Standards used	SOAP specific standards (WSDL ,UDDI, WS-Security)	Web standards (URL, HTTP methods, XML, MIME Types)
Security/Confidentiality	WS-security standard specification	HTTP Security

The rest of the paper is organized as follows: Section II presents a short introduction to the current state of art for providing Web Services from mobile devices and highlights the main issues encountered when distributing mobile Web Services. Section III describes the main modules that are used for building standard SOAP and RESTful mobile services. Section IV presents an evaluation between SOAP and RESTful MHWFs in non-offloading environment. Section V explores some distribution mechanisms that allow reliable and light weight provisioning of complex mobile Web Services and outlines different types of offloading mechanism. Section VI describes our architecture and implementation that supports provisioning of distributed mobile services. Section VII introduces a critical analysis between the two extended frameworks (i.e., the SOAP and REST MHWFs) in handling offloading strategies for different types of resource intensive applications. Some of their features and issues are also addressed in this section. Finally, conclusions from this work are presented in the last section along with recommendations for some future work.

II. STATE OF THE ART

There has been extensive research into the development of MHWFs. Most of the implemented frameworks allow deploying and providing SOAP-based mobile Web Services either in a client / server environment [7-9] or in a P2P network [10-11]. Some researchers have focused on applying mechanisms that allow adaptation and compensation for the lack of resources. For example [12] proposed a partitioning technique to the layered MHWF approach [13] that allows the execution of complex large Web Services on mobile hosts. However, in this approach clients send requests first to a stationary intermediate node, which contradicts an essential mobility requirement of mobile Web Service hosts.

Furthermore, this approach relies only on SOAP-based Web Services that require heavy weight parsers and large message payloads. Consequently the overall MH performance is degraded. The Modular Hosting Web Services architecture [14] contains built-in modules to support continuous provisioning of mobile Web Services in P2P network environment. This is accomplished through migrating services to another surrogate mobile node when the mobile host becomes inaccessible due to location changes or drained battery power. However, this framework provides only SOAP-based simple Web Services and does not allow light weight processing of complex services. Recent research studies focus on building resource aware mobile Web Service provisioning architecture that supports RESTful-based mobile Web Services. An evaluation of RESTful Web Services that are consumed from mobile devices is presented in [15], however, this evaluation is constrained to mobile Web Service consumers and does not include mobile Web Service providers. The concept of REST-based *Mobile Web Services* (MobWS) is introduced in [16] and a comparison with SOAP architecture in terms

of HTTP payload is carried out in [17] but the implementation of a mobile host that provides RESTful Web Service is not addressed. Providing adaptive mobile Web Services and testing REST for distributed environment are also not tackled. RESTful-based mobile Web Service framework is proposed for the first time in [1] and a detailed comparison is carried out between SOAP- and RESTful-based MHWFs and analyzed. The evaluation involves performance, resource consumptions and scalability. The analyzed preliminary results showed that RESTful-based MHWF is a promising technology that is more suitable for limited resource mobile network environments. However, the proposed frameworks have not address the provisioning of complex mobile Web Services. Mobile Web Service distribution is acquired for executing complex and large applications to lessen the burden on mobile host and preserve its resources and energy consumption [18].

In contrast to the approaches described above for providing mobile Web Services from mobile hosts, we aim to allow light weight provisioning of mobile Web Services, reduce mobile host energy usage and increase scalability and throughput. This aim can be achieved through distributing the execution of mobile Web Services for both SOAP and RESTful-based MHWFs and comparing them to each other. This comparison is needed to allow us to define the most suitable framework for distributing the execution of complex mobile Web Services. The selection criteria used for comparison are based on minimizing the offloading overheads and increasing overall performance.

III. System Architecture

Web Services are not explicitly defined for the mobile wireless environment. The current standard Web Service frameworks are developed for static servers. In addition, these standard frameworks are too large to be deployed on resource constraint mobile devices and they require a running time environment that is not available on mobile devices. Also providing Web Services from mobile hosts consumes a large amount of resources and drains the batteries within a short period of time. Thus, providing Web Services from mobile devices requires building a dedicated framework for deploying, providing and executing Web Services. In our previous work [1] we developed two different frameworks. One supports RESTful-based mobile Web Services that is built for the first time up to our extent knowledge and the other supports SOAP-based mobile Web Services. In implementing our framework, Java for Mobile Edition JME is used as the best language for launching applications on limited resource mobile devices. JME defines two configurations: the *Connected Device Configuration* (CDC) and the *Connected Limited Device Configuration* (CLDC). In this research CLDC has been selected because it is a low-level specification, suitable for wide range of mobile devices with limited memory capacity. Thus, CLDC achieves scalability and generality. APIs and libraries are added to support more features *through Mobile Information Device Profile* (MIDP). In this research MIDP 2.0 is chosen because it supports devices with limited network communication resources and device internal resources. Also it provides more networking functionality and it supports HTTP protocol. In addition, it supports the Server Socket connection that is required for implementing mobile server. In general the execution model and the architecture of the two frameworks are identical MHWF. The architecture is presented in Fig. 2.

The model consists of five main building blocks:

1. Web ServiceServlet
2. HTTP Listener
3. Request Handler
4. Parser Module
5. Response Composer

Although the overall architecture of SOAP and RESTful-based MHWF is similar, they differ in the details for handling and parsing the request. For example, in SOAP-based MHWF the Request Handler will un-wrap the incoming HTTP POST request to extract the hidden SOAP envelope then it will dispatch the envelope to the message parser. On the other hand the request handler for RESTful-based MHWF will extract the HTTP request directly and send it to the Message Parser Module. The main function for the Parser Module is to get the needed information for invoking a Web Service such as the name of the service, service URL and some parameters. Then the extracted information is sent to the Service Servlet. However, the way this is performed is different between the two frameworks. In SOAP-based MHWF, the SOAP parser de-serializes the SOAP object and maps the data types into Java objects using kSOAP2 and kXML2 that are open source APIs for SOAP parsing. However, in RESTful MHWF we have created our own String Manipulator -based parser. This parser will extract the server name and the parameters that are required for executing this service.

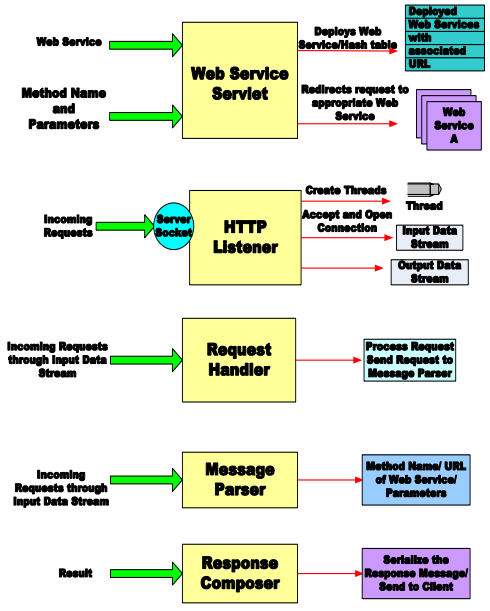


Figure 2: Architecture of Mobile Web Service Framework

The next section introduces an analytical and experimental analysis between the two SOAP and RESTful architectures in non-offloading environment.

IV. NON-OFFLOADING EXPERIMENTS AND RESULTS

On a first claim the difference between the two previously implemented architectures are fairly similar and there is no apparent difference in complexity but the major different comes when we have tested the architectures' performance, scalability and amount of resource consumption.

The evaluation is conducted using a small test-bed that consists of a mobile host developed on N80 Nokia mobile device running Symbian OS, MIDP 2.0 profile. It is connected in a wireless network through built-in IEEE 802.11b interface and it provides services to a client that is simulated using Sun Wireless Toolkits 2.5.2 emulator. The evaluation involves three different scenarios. The first set of experiments is done to test the performance of the mobile host. Performance is analyzed through measuring the effect of varying the request message size on the average processing time. Results in Fig. 3 and Fig. 4 show that the average processing time increases when the request message size increases.

Moreover, the average processing time for SOAP-based MHWF is larger than the average processing time for RESTful-based MHWF for the same message request. This is because processing SOAP requests requires heavy weight parsers to un-wrap the SOAP envelope from the incoming

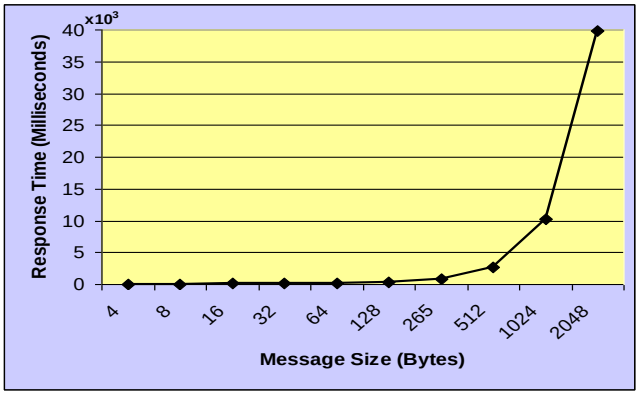


Figure 3: Effect of message size on process time of SOAP-based MHWF

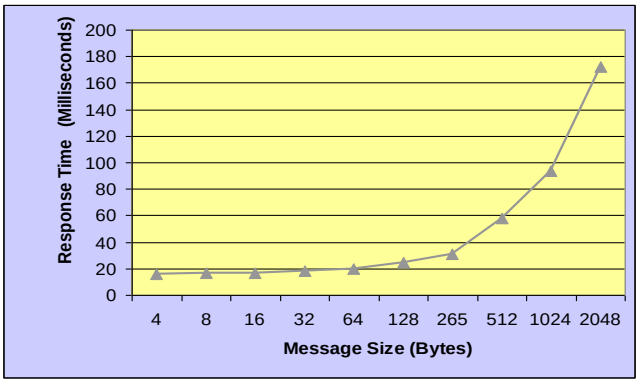


Figure 4: Effect of message size on process time of RESTful-based MHWF

HTTP POST request, then de-serialize the SOAP object and map the data types of the XML-based message into Java objects. This is done to extract the hidden information needed for invoking the required Web Service. However, processing RESTful requests uses light weight parser that is created by us to extract the information required for invoking the designated Web Service. Moreover, the required information resides explicitly on the HTTP request. Thus, RESTful-based MHWF has better performance than SOAP-based framework.

The second scenario evaluates reliability and scalability of the frameworks. This evaluation is carried out by testing concurrency where a number of clients send requests to the same host simultaneously. Concurrency is accomplished through initiating threads and loops on the client emulator. Then the average process time for each concurrent request is calculated. Results Fig. 5 and Fig. 6 show that as the number of concurrent requests increases, the average process time also increases. This increase is more obvious in SOAP-based framework where more time is consumed to parse the SOAP envelope and to manage the threads. However, we observe that the increase in RESTful-based MHWF is almost steady. This is because RESTful Web Services support caching and demand light processes power. Hence, RESTful-based MHWF is more rigid and robust to changes in the number of concurrent requests.

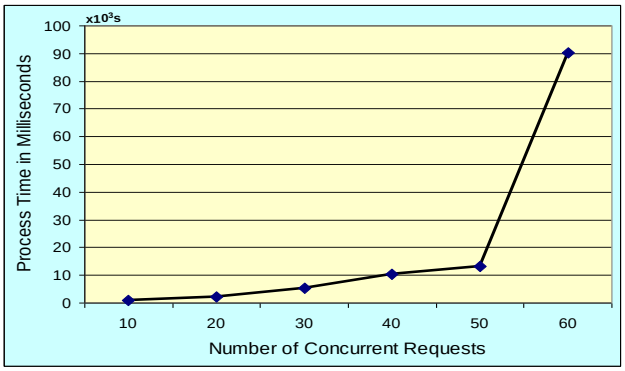


Figure 5: Effect of Concurrent requests on process time for SOAP-based MHWF

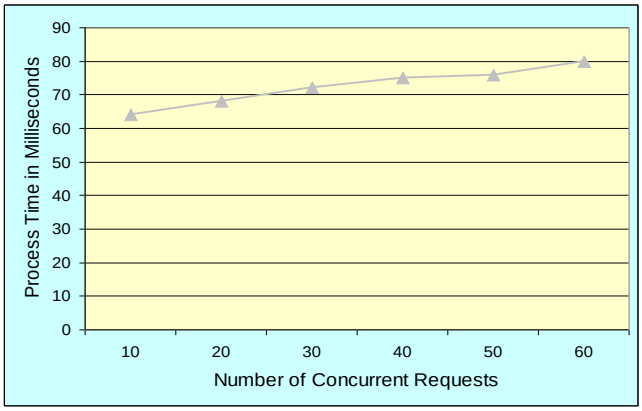


Figure 6: Effect of Concurrent requests on process time for RESTful-based MHWF

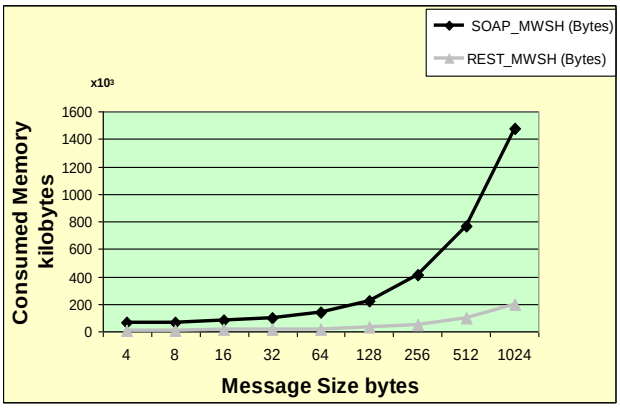


Figure 7: Comparison of consumed amount of memory between SOAP and RESTful-based MHWFs

After that, the two MHs are stressed by adding more concurrent requests to measure the threshold value. The threshold value is defined as the maximum number of concurrent requests that can be handled without failure. It is observed that in Table 3 SOAP-based MHWF starts to reject requests earlier when the threshold is beyond 60 but RESTful-based MHWF starts to reject requests when the threshold is beyond 80. This is expected because processing SOAP-based requests requires more time. Consequently, the consumed response time is larger and the server queue of the SOAP-based framework will be occupied and filled within a short period of time. As a result, there will be no more resources to accept new connections. Thus, RESTful-based MHWF is more scalable and reliable than SOAP-based MHWF.

The last scenario is for testing resource consumption and measuring memory footprints. Results in Fig. 7 illustrate that the amount of consumed memory during processing Web Service requests is increased as the message size increases. As shown in the graph the amount of consumed memory in SOAP-based framework is larger than the amount of consumed memory in RESTful-based framework for the same message size. The reason for this is that SOAP-based framework demands more memory footprint during processing. This consumed memory footprint is used to store general temporary parsed objects and to load the classes, kSOAP and kXML libraries.

TABLE3. Comparison of rejected requests between SOAP-based and RESTful-based MHWFs

No of Requests	Average rejected requests (SOAP)	Average rejected requests (REST)
60	10	0
80	59	4
100	64	9
120	86	14

V. MOBILE WEB SERVICE DISTRIBUTION

The purpose of this research as mentioned before is to investigate, define and provide mechanisms that will facilitate continuous provisioning of complex services in a light-weight processing power with efficient levels of performance. This is achieved through distribution of mobile Web Services. There are some factors that necessitate Web Service distribution in mobile environments. An important issue relates to the enormous spreading of distributed computing systems in a *Peer to Peer* (P2P) network. In P2P networks, nodes are both providers and consumers. P2P networks have some advantages that make it outperform its corresponding typical client/server networks. Avoiding single point of failure and increasing system capacity are some of these advantages. Since P2P is increasingly evolving, therefore, the application of distributed mobile Web Services executed and deployed in a distributed network environment is an important direction for future research.

Moreover, distributing Web Services is done to lighten the processing weight on limited resource mobile web servers. In spite of the fact that these constraints may be eliminated in the future and the resource capabilities might advance, the ideal performance and the minimum latency will always be the dominant requirements. In addition, resource limitations will still exist as user demands increase. For example, the memory capacity of mobile devices will continue to increase but memory limitation occurs when user wants to run multiple services or multiple instances of the same service on the MH. Furthermore, battery life will, for the foreseeable future, remain a bottleneck. Hence, the distribution of mobile Web Services results in preserving energy resources, scalability increase and an overall performance enhancement. It should also be noted that running complex large Web Services on an overloaded MH requires large processing power and might affect its core functionality.The first step for distributing Web Services is to define criteria for triggering distribution, in our case this has been done using Fuzzy Logic, however, this and the

resource monitoring system are beyond the scope of this paper. The next step is to partition the execution tasks of a Web Service and execute partitions on different remote machines. This mechanism is called *offloading*.

We have defined different schemes for applying the offloading mechanism in mobile network environment. The main difference between these schemes is the methodology used by the mobile host for handling requests and responses. The first scheme is called Forward-Offload and shown in Fig. 8. In Forward-Offload a client sends a request to the MH then it forwards the request to an AMH for processing. After that, the AMH sends its response to the MH, which forwards the response to the client. This type of communication relies on the MH to partially process the request, select the AMH and to maintain communication subsystem TCP. However, it supports ubiquitous computing through distributing the execution autonomously without the client being aware.

The second case is called Bounce-Offload. Fig. 9 illustrates Bounce-Offload where the client sends a request to the MH, which then bounces the request back to the client, redirecting the request to another host for processing. This type of communication lessens the load on MH, preserves its resources and reduces the signaling exchanges (compared to Forward-Offloading). Thus, it increases the capability for the mobile host to handle more requests concurrently and increases scalability. However, these benefits are gained at the expense of putting a greater burden on the client to tackle the task of contacting another host. The critical analysis between the two offloading strategies has been carried out by us and will be published in another paper.

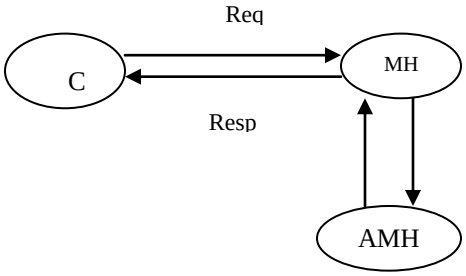


Figure 8: Forward-Offload

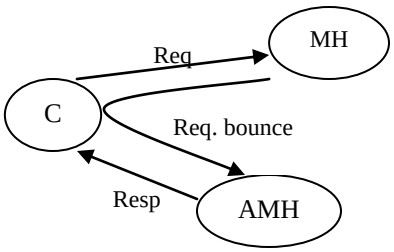


Figure 9: Bounce-Offload

In this publication, Forward-Offload is examined to support ubiquity and autonomy. However, this scheme consumes more resources than Bounce-Offload. Thus, our aim is to minimize resource consumption as much as possible. This goal can be achieved through a coherent study of the signaling and processing overheads for both extended SOAP- and REST- based MHWFs. The next section explains and illustrates the architecture of the aforementioned extended MHWFs.

VI. Mobile Web Service Distribution Architecture

The MHWFs architecture that has been implemented previously [1] for providing, deploying and executing SOAP and RESTful- based mobile Web Services is extended to allow distribution and offloading functionality. This is accomplished by using the previously implemented architecture for developing the AMH. The AMH will take the role of a mobile host temporary and performs its typical tasks such as handling the forwarded requests, invoking the required service, executing it and sending the result back to the MH. However, the architecture of the mobile host is an augmentation of the basic built MHWFs. The augmentation is taken place through adding an Offloading Module as shown in Fig. 10. The main task of the Offloading Module is to transform the role acted by the MH from server to client temporary. This is carried out to allow MH to forward incoming requests to AMH. MH partially processes incoming requests to extract the name of the requested Web Service and its associated parameters. Another important task for MH is to select the appropriate AMH that satisfies some predefined conditions. The following section introduces the prototype that is used for testing and examining the validity of distributing SOAP and RESTful-based Web Services in mobile environments.

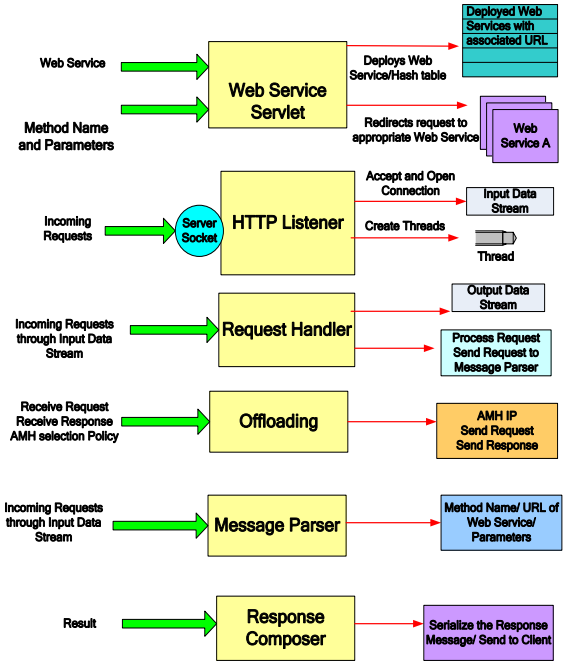


Figure 10: Architecture of MHWF with offloading functionality

VII. EXPERIMENTAL RESULTS AND EVALUATION

As aforementioned, the main objective is to investigate the offloading mechanisms and to examine the feasibility and validity of distributing SOAP and RESTful based-Web Services in mobile environments. Another objective is to test and compare two different architectures to assist in selecting an architecture that is most suitable for distributing mobile Web Services with fewer overheads and less resource consumption.

The experimental approach we followed evaluates functional and non functional properties in two different environments: offloading and non-offloading environments. It also applies two different resource intensive applications for each environment: processing and bandwidth intensive application types. Tests for non-offloading environment have been carried out in the previous section. Following is a description of the test taken for offloading environment.

A. *Offloading Experimental Environment*

A small prototype is proposed to carry out the experiments needed to address the validity of offloading mobile Web Services and distributing the execution tasks of a large complex Web Service between different mobile hosts. We have extended the two architectures for the main MH by adding an Offloading Module and using the same previous MHWF architectures for the AMH. The evaluation was conducted using a prototype compromising three mobile devices as shown in Fig. 11: The MH is executed on a mobile device (Nokia N97m) running MIDP 2.1 over Symbian OS. The other device, implementing the auxiliary AMH that acts as mobile host when the original MH is

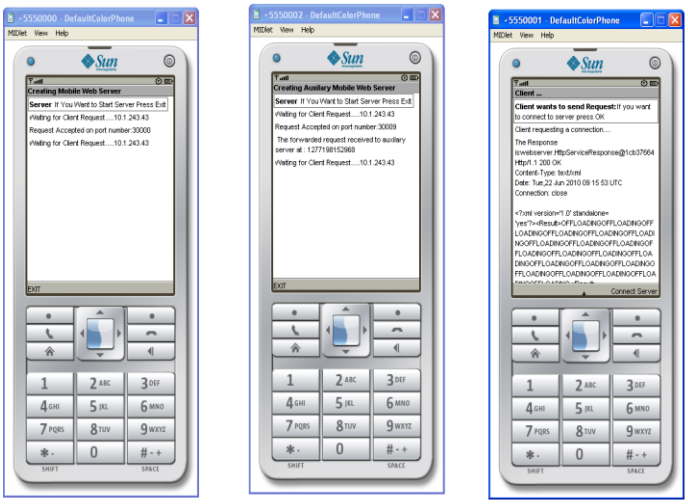


Figure 11: Prototype for offloading mobile Web Services

overloaded, was executed also on an N97m. The client was executed on a Laptop using the Sun Wireless Toolkit and emulator. The devices were connected via a wireless network. In this experiment Forward-Offload strategy has been applied. Since the MH is assumed to be overloaded it processes part of the incoming requests and forwards it to AMH. The MH elects an AMH.

The election is carried out using probe requests sent to all mobile devices that satisfy set of predefined criteria. However, this is beyond the scope of this paper. The evaluation has been accomplished for two different services. The first Web Service represents processing intensive application. The example used for this type of applications was a simple PI calculation service. In this service the accuracy for calculating PI depends on the number of terms that are added together. The number of terms is controlled by a client using an integer parameter. The other type of services represents bandwidth intensive application. The service used for bandwidth intensive applications was a simple String-Concatenation. In this service, the number of times constant is merged and concatenated depends on a parameter (i.e., an integer value) set by the client.

The evaluation for both services is carried out using three different scenarios. In the first set of experiments the level of internal resource consumption is examined including both memory and processor resources. In the second set of experiments the level of external resource consumption is estimated by calculating the total amount of interactions between the three connected mobile devices. In the third set of experiments the overall performance is evaluated by measuring total elapsed response time for execution of each request. After that, the offloading overhead is analyzed. Finally, the performance improvement is evaluated for both (SOAP and REST) architectures in the last set of experiments.

B. Results for Offloading Process Intensive Web Service

The first application scenario demands intensive processing power. The application represents a simple mathematical service called PI Web Service used to calculate the constant π whose value can be approximated using Gregory-Leibniz series [19]:

$$\pi = 4 * \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1}$$

We used different values of *k* in our experiments to vary the computational intensity of the Web Service sample. PI is a suitable service for accomplishing the required tests. This is because it represents intensive power applications where the amount of consumed power can be controlled via *k* parameter, which determines the number of accumulated terms. First, the amount of internal resources consumption is examined for different values of *k* to investigate the effect of varying application process complexity level on the MH resources. These internal resources include both MH memory and MH processing power that are required during executing and offloading incoming Web Service requests. Tests run for both architectures RESTful and SOAP-based MHWF. The memory consumption is averaged for 50 requests for different values of *k*. Memory is estimated by calculating the difference between the total available amount of memory on MH before processing incoming requests and the available memory after processing requests before sending them to clients. However, since the heap memory size of mobile devices is variable, then a technique for controlling the variation of mobile host memory is applied. This is done by releasing the unused objects then freeing the memory heap by running garbage collection before measuring the total available memory amount. Results presented in Fig. 12 show that with offloading, changing the application processing complexity has no effect on the memory consumption amount for the main mobile host. This is because the real processing and memory allocation are delegated to another auxiliary mobile host. Moreover, RESTful-based architecture saves more memory resources than the conventional SOAP-based architecture. The average amount of CPU processing power is also tested for different values of *k*. In general the amount of CPU processing power can be estimated by measuring the processing time required to execute a predefined task by the CPU. In the offloading process the MH processing time includes two parameters they are: the time required to process incoming requests from clients and the time required to process incoming responses from the AMH. Thus, the average processing time is the summation of the average time spent for client requests in MH before it being forwarded to the AMH plus the average time spent for responses that are delivered from AMH to MH before it being forwarded to designated client. This average process

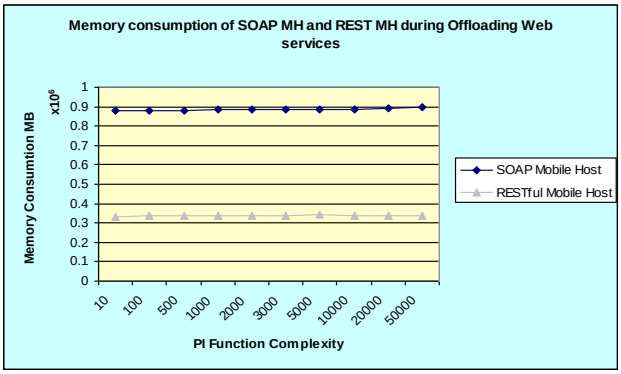


Figure 12: Memory Consumption of SOAP and REST mobile hosts

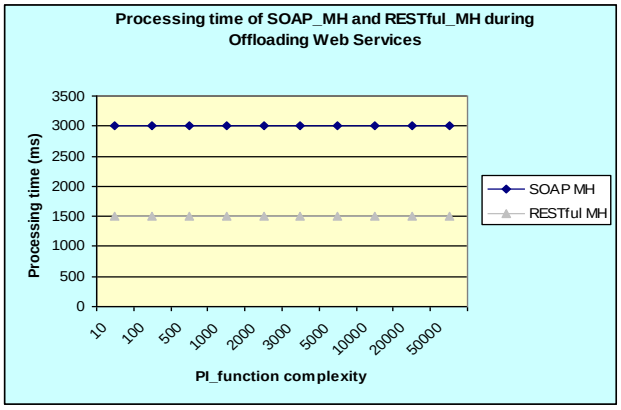


Figure 13: Processing time for SOAP and REST mobile hosts

time is measured for different *k* values. Fig. 13 illustrates that the average processing time required by MH is constant since it compromises the process of parsing requests and responses that have invariable payload length. On the other hand invoking and executing the required service that has variable complexity takes place remotely on AMH. Moreover, SOAP-based MHWF demands larger processing power than its corresponding RESTful-based MHWF. This is because processing SOAP requests requires heavy weight parsers to un-wrap the SOAP envelope from the incoming HTTP POST request. However, processing RESTful requests uses a light-weight String-based parser that is created by us to extract the information, which resides explicitly on the HTTP request. Thus, RESTful-based MHWF consumes fewer amounts of internal resources than SOAP-based framework. This preserves more resources for the MH to allow it to handle more requests and deploy more active Web Services. Consequently RESTful-based MHWF increases scalability and throughput in distributed mobile Web Service environment.

Second, the level of external resource consumption is tested for different values of *k*. Bandwidth consumption is one of the most critical external resources in mobile wireless environment. This resource is predicted through computing the total amount of data transferred in a predetermined amount of time, which mainly depends on the size of both request and its corresponding response.

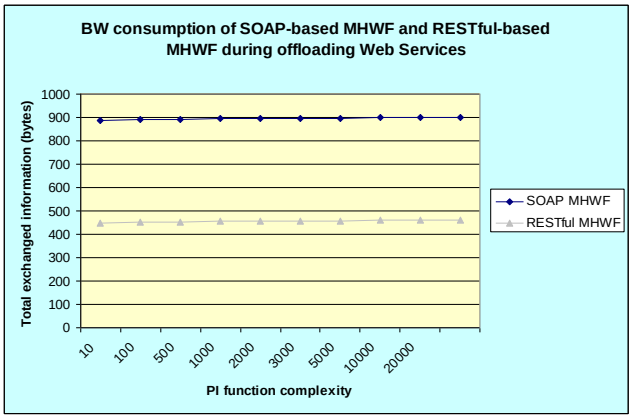


Figure 14: Bandwidth Consumption for SOAP and REST-based MHWF

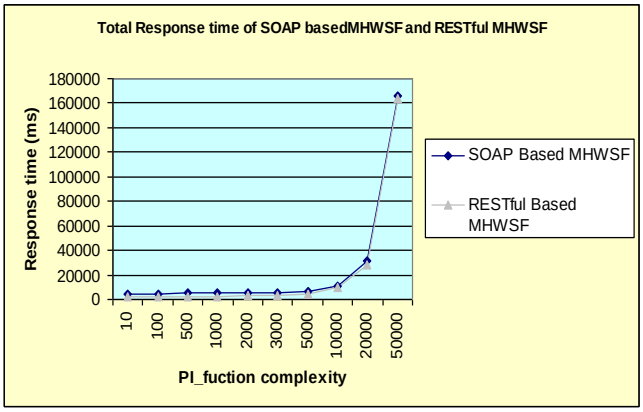


Figure 15: Total response time for SOAP and RESTful-based Web Services

For simplicity we used the average total amount of interactions between the three mobile nodes (client, MH and AMH). With respect to Fig. 14, it is shown that RESTful-based MHWF outperforms the standard SOAP-based MHWF and contains approximately 50% less amount of data exchanged. This result is expected because SOAP messages are verbose XML and they require an envelope to hide the service name and parameters in the body of the HTTP request. However, RESTful-based messages are based on the standard HTTP and the service name with its associated parameters are explicitly reside in the HTTP URL. Hence, RESTful-based MHWF requires less bandwidth than SOAP-based MHWF.

Finally the average response time is measured for different *k* values and for both architectures. Response time is defined as the time that a client spends waiting to receive the result from the MH. This is measured by calculating the difference between the time when a response is received by the client from the MH and the time when a request is sent by the client to the MH. Results presented in Fig. 15 show that the average response time is directly proportional to the complexity degree of the application being processed. The proportional relation refers to the two parameters that dominate the response time value: communication delay and the processing time on both MH and AMH. Although the processing time on MH is constant and does not change with different *k* values, the processing time on AMH as shown in Fig. 16 is variable and it increases for larger values of *k*. Moreover, SOAP-based MHWF requires more response time than RESTful MHWF for the same *k* value. This is because SOAP-based MHWF requires more communication delay and processing time on MH than RESTful-based MHWF.

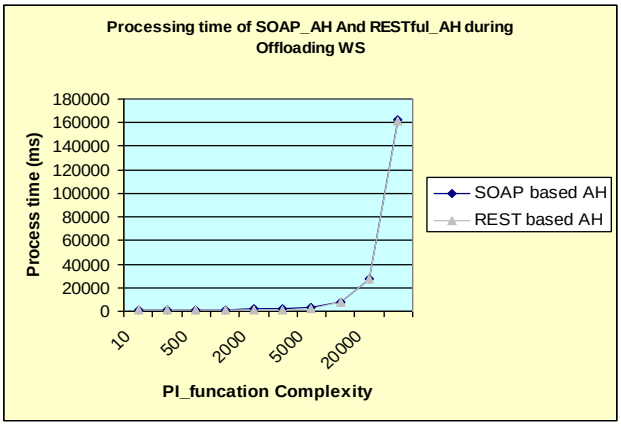


Figure 16: Processing time on Auxiliary Mobile Host for SOAP and RESTful-based MHWF

C. *Results for Offloading Bandwidth Intensive Web Service*

The second application scenario is aimed to carry out tests for applications requiring intensive bandwidth. The String-Concatenation service used to evaluate the architectures consumes network bandwidth and demands CPU processing power depending on the size of the concatenated string. The request contains an integer parameter value *l*. *l* determines the number of iterations for concatenating a specific string. The output of this service (a concatenated string) is then returned to the client. The size of the concatenated string is controlled by varying the value of *l*. Consequently the size of response message payload is increased by increasing the input value *l*.

The first set of experiments is conducted to examine the amount of internal resources consumption for different values of *l*. These resources include both MH memory and MH processing power that are required during executing and offloading incoming Web Service requests. Tests run for both architectures. The memory consumption is averaged over 50 requests. Memory is estimated by calculating the difference between the total available

amount of memory on MH before processing incoming requests and the available memory after processing requests before sending them to clients. However, since the heap memory size of mobile devices is variable, then a technique for controlling the variation of mobile host memory is applied. This is done by releasing the unused objects freeing the memory heap before measuring the total available memory. Results presented in Fig. 17 show that with offloading, the memory consumed on the MH increases as the response message size increases. MH allocates more memory for storing the increased response before it is forwarded to the corresponding client. Another observation is that the REST implementation uses less memory than the SOAP based architecture. This is due to the smaller overhead of REST messages compared to the corresponding SOAP messages. Then, the second examined resource is the CPU load consumed by the MH. This is determined by measuring the average process time on MH (averaged over 50 requests). Fig. 18 presents the effect of varying response message lengths on the average processing time for the SOAP- and REST implementations. The results show that the MH spends more time receiving and reading responses with larger payloads than those with smaller payloads. Moreover, the average processing time needed by the SOAP implementation to run a service is larger than the average processing time needed by the REST implementation. SOAP requests require comparatively heavy weight parsers to un-wrap the SOAP envelope from the incoming HTTP POST request while requests in REST use light weight string-based parsers. Thus, the REST implementation consumes overall fewer resources than the SOAP implementation.

The second set of experiments designed to evaluate the bandwidth required to offload and distribute the execution of mobile Web Services between several mobile nodes. This was accomplished through measuring the total amount of information that is transferred between client, MH and AMH. String-Concatenation service is used again, and as the input value *l* increases, the size of the concatenated string increases as well, which results in an increase of the response message size. This is clearly shown in Fig. 19.
In this case SOAP needs more information than REST by approximately 482 bytes to store the Web Service parameters and method names inside the body of the HTTP request. Therefore, SOAP messages require more wireless bandwidth than REST messages.

The third set of experiments measured the average response time for different input values of *l* for both architectures. Response time includes the processing time spent on both MH and AMH for handling client request, invoking the required Web Service, executing it, composing the result and sending it back to the client. In addition, it involves the transmission delay for messages to transfer between the designated mobile nodes through socket connections.

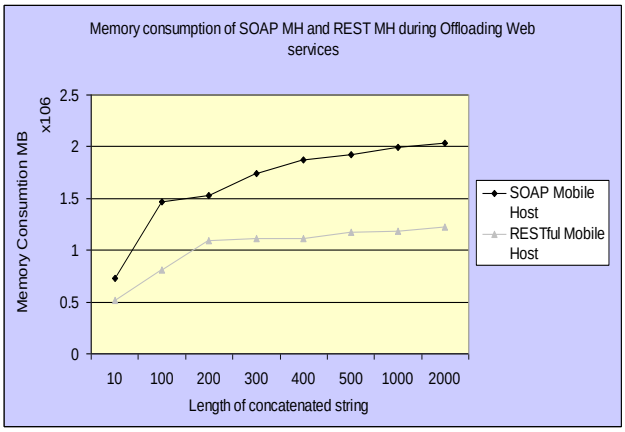


Figure 17: Memory Consumption of SOAP and REST mobile hosts

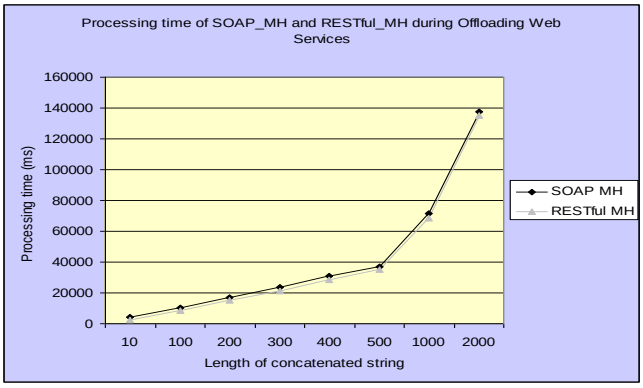


Figure 18: Processing time for SOAP and REST mobile hosts

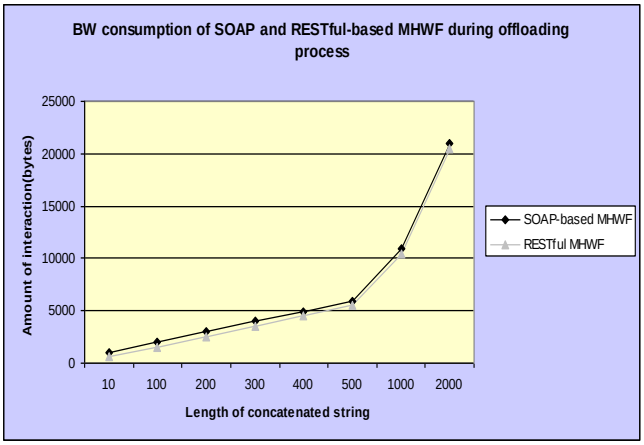


Figure 19: Bandwidth consumption of SOAP- and RESTful-based MHWF during offloading

The results of this experiment are presented in Fig. 20. As the size of the response message increases, the average response increases. This is expected because for this experiment, the response time is composed of the MH processing time, which increases with increasing message size as shown in Fig. 6. AMH processing time is another component for the response time that also increases with increasing message size as shown in Fig. 18.

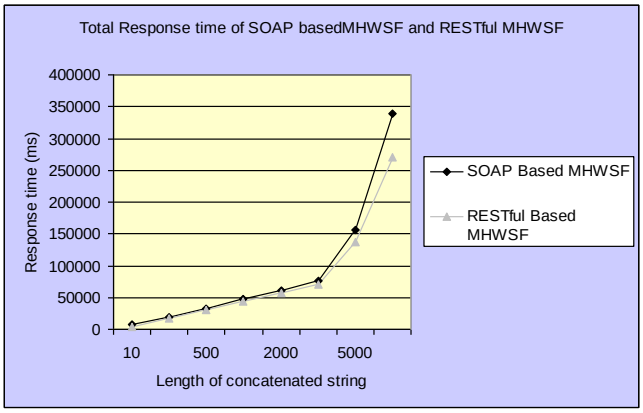


Figure 20: Total response time for SOAP- and RESTful-based Web Services

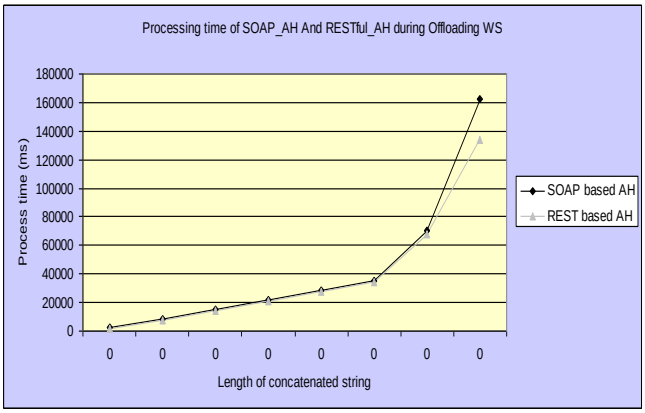


Figure 21: Response time for SOAP/RESTful-based MHWF during offloading

D. *Offloading overhead Experimental Results*

The overhead of distributing the execution of conventional SOAP-based MHWF and the new RESTful-based MHWF is examined in this section. The overhead is caused by the coordination and management of the task partitioning. The overheads include memory, processing, response time and signaling/messaging. Moreover, this is measured in for both implementations (as fore described). In this set of experiments we implemented prototypes for both architectures based on the typical original MHWF. Each of these prototypes consists of a client simulated using Sun Wireless Toolkits 3.0 emulator and n97 Nokia mobile host. The mobile host and client are connected in a wireless network. The test is carried out using the two aforementioned resource intensive applications. (i.e., PI and String-Concatenation)

In all experiments only one parameter is measured at a time. Each client operates cyclically and sends one request waits until it receives the response back then repeats the cycle and sends the same request again. This cycle is repeated 50 times for each experiment and the average of these 50 measurements is calculated. Then the measured parameters are compared with its corresponding parameters that are measured during applying offloading mechanism. As mentioned above these parameters include memory and processing consumption on the MH that indicate the amount of resource consumption overheads. Other parameters are the amount of interaction and response time that indicate the amount of communication/signalling overheads. RESTful-based MHWF framework shows an inferior performance in comparison to SOAP-based MHWF framework regarding distribution of mobile Web Services. Fig. 22 and Fig. 23 emphasize this fact and prove that RESTful MHWF shows smaller resource consumption and signalling overheads than SOAP- based MHWF. RESTful MHWF is also preserve approximately 42% more amount of memory than SOAP MHWF for *k*=10 in PI application. Moreover, the difference in overhead is more obvious for applications with more processing and bandwidth intensity. For example, in String-Concatenation test case REST-based implementation requires approximately 70% less processing cycles, 68% reduced delay and 59% fewer messages to provide the same service in SOAP-based implementation.

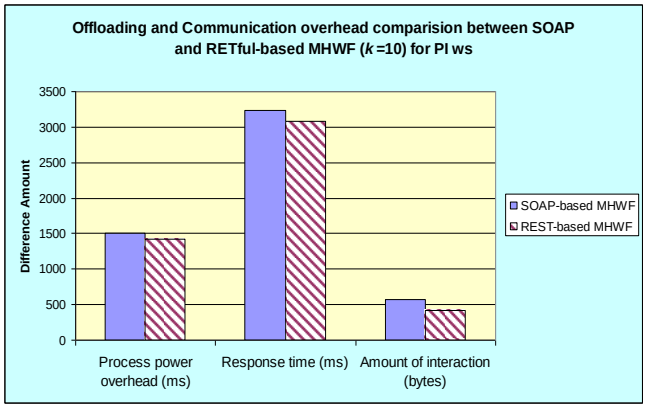


Figure 22: Offloading and Communication overhead for SOAP and RESTful-based MHWF (N=10) for PI Web Service

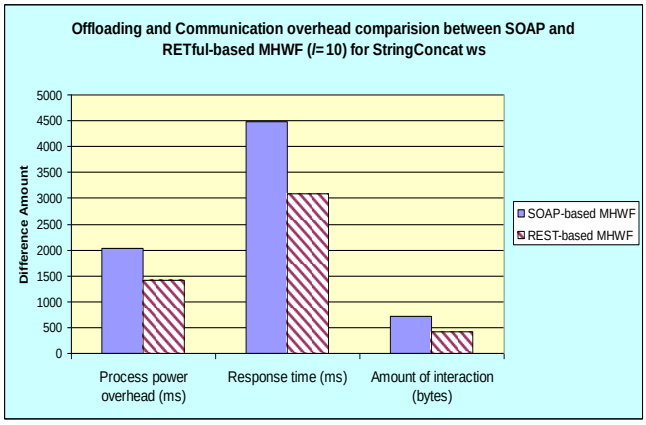


Figure 23: Offloading and Communication overhead for SOAP and RESTful-based MHWF (N=10) for String-Concatenation Web Service

E. Performance Improvement Experimental Results

The performance of distributing the execution of conventional SOAP-based MHWF and the new RESTful-based MHWF has been further analyzed and examined in this section. This analysis is carried out to critically measure the amount of REST over SOAP performance improvement gained from offloading. The parameters that are used for measuring performance improvement include amount of memory, response time and total message length enhancement. These parameters are evaluated for both Web Service samples (PI and String-Concatenation). Results in Fig. 24 and Fig. 25 show that offloading and distributing RESTful Web Services can achieve more performance improvement over its corresponding SOAP Web Services compared to the improvement that can be achieved in non distributed environments. In addition, the amount of processing power enhancement is slightly more for computational intensive. On the other hand, the amount of communication delay enhancement is more for bandwidth intensive applications.

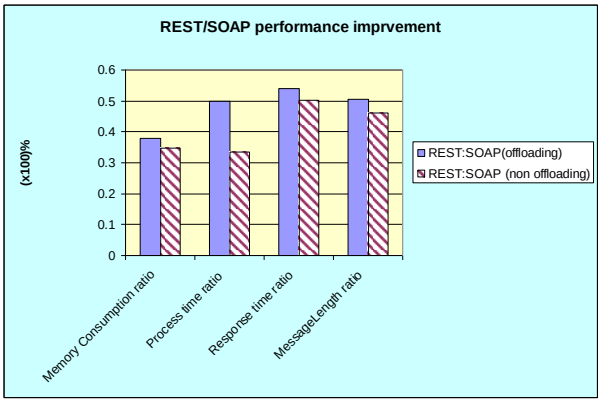


Figure 24: REST/SOAP performance improvement for offloading and non-offloading Web Services (PI Web Service)

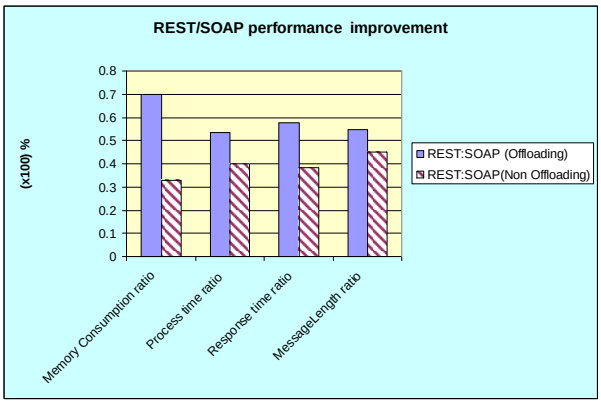


Figure 25: REST/SOAP performance improvement for offloading and non-offloading Web Services (String-Concatenation Web Service)

VIII. DISCUSSION

RESTful- versus SOAP-based mobile Web Service distribution are evaluated based on four main parameters. These parameters constitute an essential infrastructure used for selecting the most appropriate WS provisioning framework for providing distributed mobile Web Services. One of the most vital parameters is performance, which always forms the main goal for building efficient frameworks

Performance is measured by testing the average processing time on the main mobile host in addition to the average response time for Web Service requests. Results meet our expectations and show that RESTful-based MHWF provides improved processing time and response time over its corresponding SOAP-based Web Services. This is because SOAP-based Web Services require heavy weight parsers to un-wrap incoming request and extract the hidden SOAP envelope from the body of HTTP request. But RESTful Web Services require light-weight parsers based on string manipulator. This String-based parser is needed to extract the information required for invoking Web Services. This information resides explicitly on HTTP request.

Moreover, the improvement achieved in the average processing time is more for offloading case than non-offloading case. This is due to the distribution of Web Services and partial execution of Web Services on MH. Results have also shown that the processing time for Web Services with fixed length message payloads is almost steady state in the offloading environment and does not vary with increasing the processing power complexity. This is because processing time on MH consisted of reading the incoming request, identifying the parameters required for invoking a Web Service such as method name, service name and related parameters, forwarding these parameters to AMH, reading incoming responses from AMH, comparing the response and sending it to client. Thus, processing Web Service on MH depends mainly on the size of the incoming and outgoing respective requests and responses. Processing time on MH does not depend on the complexity of the Web Service logic that will be executed remotely on AMH. Similarly, RESTful-based MHWF provides better average response time than SOAP-based MHWF due to support for caching. In addition, response time involves processing time on MH, processing time on AMH and communication delay.

As illustrated earlier processing SOAP requires comparatively heavy-weight parsers and consumes more time. Furthermore, communication delay is directly proportional to the size of transferred message, which is larger for SOAP than REST. The second dominant parameter is scalability and reliability of the developed framework. Since RESTful Web Services are idempotent, therefore, sending repeated request to compensate for reliability is safe and simple. On the other hand, reliability of SOAP is achieved by using a WS- reliability standard that encounters

some implementation complexity and augments the size of the original SOAP message. Results present more scalability with RESTful-based MHWF and more requests can be executed concurrently than the conventional SOAP-based MHWF. This is because REST requests are stateful and reduce the need for the MH to maintain communication state. RESTful-based MHWF is also more scalable, because its corresponding requests are smaller in size and occupy less space waiting in the server queue than SOAP requests. Another parameter that is addressed by that evaluation is the amount of consumed resources: internal and external constrained mobile resources. Results have proved that RESTful-based MHWF preserves more processing power, memory storage space and network bandwidth than SOAP-based MHWF. This is because processing SOAP requests requires more extensive processing power for parsing and serializing SOAP object. More memory is also needed to load parser libraries and to store temporary parsed objects.

Furthermore, in comparing SOAP and REST requests we can easily notice a significant reduction in requests payload. Hence, RESTful-based MHWF consumes less network bandwidth during transmission of smaller REST message payloads. This result is more trivial with bandwidth intensive applications where the amount of interaction reduction increases approximately from 54%-97%.

The last parameter is the overhead caused by adding offloading module to the existing framework. Results have shown that RESTful-based MHWF intercepts less overhead than SOAP-based MHWF. This is due to less total amount of interactions, processing time, response time and memory requirement.

However, there are some limitations with RESTful Web Services. First they are only used for HTTP transport layer. In addition, transaction and federation are not supported by REST. SOAP is more suitable for complex Web Services that require a contract in advance between client and Web Service provider.

IX. Conclusion and Future Work

Mobile Web Services are provided from resource constrained mobile hosts in an intermittent wireless network. Thus, so far there were clear limitations in terms of complexity and size of the services that may be executed on mobile hosts. Providing adaptive mobile Web Services is vital to allow reliable provision of complex Web Services from resource limited mobile devices to overcome resource constraints.

This paper has explored one of the mechanisms used to facilitate the provisioning of adaptive mobile Web Services. The explored mechanism is known as offloading. This is accomplished by extending the two frameworks SOAP-based MHWF and RESTful-based MHWF developed in [1]. The novelty of this work to the best of our knowledge is that it is the first work that investigates provisioning of RESTful-based distributed Web Services from mobile devices.

The two frameworks are extensively tested and analyzed using two types of applications, process intensive application and bandwidth intensive application.This analysis is needed to select the most appropriate implementation technology that suits adaptive and distributive mobile Web Services.

Our preliminary work shows that extended RESTful-based MHWFs outperform SOAP-based MHWFs. Moreover, RESTful-based MHWF has less offloading and interaction overhead. In addition, it has more performance improvement over SOAP-based MHWF and less resource consumption in offloading environment than in non-offloading environment.

The level of resources consumption improvement depends on the type of application. Performance enhancement is obvious for resource intensive applications.

In addition, RESTful-based MHWF supports caching; this saves the limited network bandwidth and increases reliability and scalability. It also reduces consumption of mobile resources. Another feature of RESTful Web Services is the loosely coupled relation between the server and client because of the uniform interface that adds a balance towards using it for distributed mobile Web Services.

Regarding future work, the first area of interest is to investigate other schemes for offloading Web Services such as the Bounce-offload strategy. Another interesting issue is to define a general structure for implementing Web Service logic to facilitate partitioning it and build an interface for orchestrating the services [20]. Moreover, distributing and offloading Web Services in dynamic mobile environment must consider multiple, possibly contradictory, issues. For example, executing a code component on a remote AMH might reduce MH energy usage at the cost of increasing execution time. Moreover, due to the variable nature of the environment, it is not feasible to use static policies to determine when and where to remotely offload services as the current resource situation may make any statically chosen policy obsolete.

References

[1] Moessner, K. and AlShahwan, F., "Providing SOAP Web Services and RESTful Web Services from Mobile Hosts," in ICIW 2010 Fifth International Conference on Internet and Web Applications and Services, 2010, Barcelona, pp. 174-179.

[2] Srirama, N., Vainikko, E., Sor, V., and Jarke, M. "Scalable Mobile Web Services Mediation Framework," in 2010 Fifth International Conference on Internet and Web Applications and Services, Barcelona, Spain 2010.

[3] Ayyagari, D., Yongii, Fu., Jingping, Xu., and Colquit, N., "Smart Personal Health Manager: A Sensor BAN Application: A Demonstration," in Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE, 2009, pp. 1-2.

[4] S.-A. Ong and N. R. Center. 2006, A Mobile Webserver-Based Approach for Tele-Monitoring of Measurement Devices. Available:http://www.sigmobile.org/mobisys/2006/demos/Ong.pdf, Access:03.01.11

[5] SOAP Definition. Available: http://en.wikpedia.org/wiki/SOAP, Access:03.01.11

[6] Fielding, R., "Architectural styles and the design of network-based software architectures," PHD, University of California, Irvine, 2000.

[7] Berger, S., McFaddin, S., Chandra, N., and Mandayam "Web services on mobile devices-implementation and experience," in Mobile Computing Systems and Applications, 2003. Proceedings. Fifth IEEE Workshop on, 2003, pp. 100-109.

[8] Asif, M. and Majumdar, S.,"Performance analysis of Mobile Web Service Partitioning Frameworks," in ADCOM 2008, 16th IEEE International Conference on Advanced Computing and Communications, 2008., 2008, pp. 190-197.

[9] L. Luqun, "An Integrated Web Service Framework for Mobile Device Hosted Web Service and Its Performance Analysis," in HPCC '08, 10th IEEE International Conference on High Performance Computing and Communications, 2008, pp. 659-664.

[10] Srirama, N., Jarke, M., and Prinz, W., "A Mediation Framework for Mobile Web Service Provisioning," in Enterprise Distributed Object Computing Conference Workshops, 2006. EDOCW '06. 10th IEEE International, 2006, pp. 14-19.

[11] Aijaz, F., Adeli, S., and Walke, B., "Middleware for Communication and Deployment of Time Independent Mobile Web Services," in ICWS 2008, IEEE International Conference on Web Services,2008,pp. 797-800.

[12] Majumadar, S., Asif1, M., and Dragnea2, R., "Partitioning the WS Execution Environment for Hosting Mobile Web Services," in SCC 2008, IEEE International Conference on Services Computing, 2008, vol. 2, pp. 315-322.

[13] Asif, M., Majumadar, S., and Dragnea, R., "Hosting Web Services on Resource Constrained Devices," in ICWS 200, IEEE International Conference on Web Services, 2007, pp. 583-590.

[14] Kim, Y.-S. and Lee, K.-H.,"A light weight framework for mobile web services " Computer Science - Research and Development, pp. 199-209, May 2009.

[15] Saad, M., Hamad, H., and Abed, R., Computer Engineering Department, Palestine, "Performance Evaluation of RESTful Web Services for Mobile Devices" International Arab Journal of e-Technology vol. 1, p. 7, January 2010.

[16] Aijaz, F., Ali, S., Chaudhary, M., and Walke, B., "Enabling High Performance Mobile Web Services Provisioning," in Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th, 2009, pp. 1-6.

[17] Aijaz, F.,Ali, S., Chaudhary, M. A., and Walke, B., "Enabling resource-oriented Mobile Web Server for short-lived services," in Communications (MICC), 2009 IEEE 9th Malaysia International Conference on, 2009, pp. 392-396.

[18] Corroy, S., Beiten, J., Ansari, J.,Baldus, H., and Mahonen, P., "Selection of Computing Elements for Energy Efficiency in Wireless Sensor Networks using a Statictical Estimation Method," International Journal on Advances in Networks and Services vol. 2, p. 10, 2009.

[19] Available: http://en.wikipedia.org/wiki/Pi,Access:03.01.11

[20] Pietschmann, S., "A Model-Driven Development Process and Runtime Platform for Adaptive Composite Web Applications," International Journal on Advances in Internet Technology, 2009, vol. 2, pp. 277-290.

Traffic Shaping via Congestion Signals Delegation

Mina Guirguis
Computer Science Department
Texas State University - San Marcos
San Marcos, TX 78666, USA
Email: msg@txstate.edu

Jason Valdez
Computer Science Department
Texas State University - San Marcos
San Marcos, TX 78666, USA
Email: jv1150@txstate.edu

***Abstract*—This paper presents a new architecture that enables a set of clients to enforce traffic shaping policies among them through the delegation of congestion signals. When congestion-aware Internet flows share a bottleneck link, they compete for bandwidth and must respond to congestion signals promptly by decreasing their throughput. For clients running real-time applications (e.g., gaming, streaming), this may impose strict limitation on their achievable throughput over short time-scales. To that end, this paper presents an architecture, whereby a set of TCP connections (we refer to them as the Stunts) sacrifice/trade their performance on behalf of another TCP connection (we refer to it as the Free) by picking up a delegated subset of the congestion signals and reacting to them in lieu of the Free connection. This gives the Free connection just enough freedom to meet specific throughput requirements as requested by the application running on top, without affecting the level of congestion in the network. We present numerical model and analysis, which we validate by extensive simulation as well as through a pluggable module implementation for the Linux kernel.**

***Keywords*-Service-oriented architecture; TCP; Congestion Control; Traffic Shaping; Control Theory;**

I. INTRODUCTION

Motivation: Certain classes of applications (e.g., gaming, audio and video streaming) need to acquire/maintain certain guarantees in order to perform adequately. Due to the "best-effort" nature of the Internet, it is very difficult to ensure that these guarantees are met or even to predict what possible guarantees could be provided. Hence, these applications are often left with unspecified guarantees on their quality of service. Research efforts have addressed this problem and proposed two major architectures, Integrated Services (IntServ) and Differentiated Services (DiffServ). IntServ architectures require *every* router to maintain per-flow state. Applications make reservations based on their needs. The main problem with IntServ is that it does not scale to a size that is as large as the Internet. Thus, it is limited to small-scale deployments. DiffServ, on the other hand, push traffic management towards the edges of the network, while keeping its core simple. Edge routers maintain and classify flows based on classes. Core routers still need to maintain brief information on how to treat each class. In both architectures, some modifications had to be made to some routers.

The "Free and Stunts" Architecture: In this paper, we propose a new end-host service architecture that provides an application with *soft throughput guarantees* over a best-effort network, without *any* modification to routers. Moreover, this is achieved in a completely friendly manner to the network through strictly adhering to the Transmission Control Protocol (TCP) rules. In particular, we envision a set of TCP connections (we refer to them as the Stunts connections) that are willing to sacrifice their own performance on behalf of another TCP connection (we refer to it as the Free connection). This would enable the Free connection to match its throughput to the target throughput from the application. In [1], we have demonstrated the feasibility of this idea through simulations only. In this paper, we extend this work by introducing a dynamic model (along with numerical solutions) as well as real implementation of this architecture in the Linux kernel.

TCP employs congestion control mainly via the Additive Increase Multiplicative Decrease (AIMD) mechanism that seeks to constantly probe for available capacity while remaining fair to other TCP flows [2], [3]. Congestion signals (as in dropped/ marked packets) signal TCP senders to slow down, by halving their congestion windows. The quantity and the timing of these congestion signals may prevent the Free connection from achieving any throughput guarantees. The main idea behind our architecture is to allow the Free connection to delegate a subset of those congestion signals to the Stunt connections. Thus, the Stunt connections would be the ones that decrease their sending rates instead of the Free connection, which would be liberated (to a larger extent) to match the guarantees requested from the application above. It is important to note that this architecture *does not* increase network congestion at the bottleneck link because it ensures that the total decrease in throughput from all the Stunt connections is at least as large as what the Free connection would have decreased, if it were to observe those delegated losses. For example, it is possible for a single packet loss delegated from the Free connection to cause more than one Stunt connection to back-off, in order to have the same equivalent effect on the bottleneck link.

The choice of Stunt connections is important due the

dynamic nature of Internet traffic. It has been evident through many measurement studies that TCP connections can be characterized into long-lived and short-lived flows, also known as elephants and mice, respectively [4], [5], [6], [7]. Elephant flows, while there are few of them, they account for around 80% of the traffic. These are long-lived stable TCP flows. Mice, on the other hand, while there are plenty of them, they account for only around 20% of the traffic. These are short-lived flows (simple HTTP requests and they typically finish before leaving the slow-start phase of TCP [8], [9]). In our proposed architecture, we limit the choice of Stunt connections to elephants as they are more stable and control the majority of Internet traffic. Moreover, delegating losses to mice would hurt their performance significantly, if they were chosen to act as Stunts.

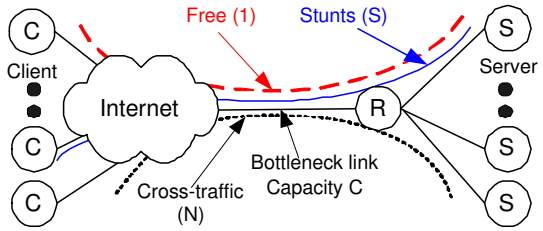


Figure 1. The Free and the Stunts setup. The proposed architecture is implemented at the End-host at the server's side.

Deployment Scenarios: This architecture is proposed to be used by Internet Service Providers (ISPs) to enforce differentiated service among its clients, by having some behave as the Free connections, while others play the Stunt roles. For example, by dropping appropriate packets (or marking them if the use of the ECN bit is enabled), an ISP (e.g., the first-hop router) can delegate losses between the Free and the Stunts. Such delegations could be based on a marketplace in which clients' agents agree upon what is fair and efficient for each one [10].

This architecture is also envisioned to be used by Internet servers that serve different forms of media content to different clients. With this architecture, a server can give a particular flow (say a media stream) the freedom to achieve requested guarantees, while making other flows (say long bulky file transfers, i.e., "elephants") behave as Stunts. In this scenario, the architecture is implemented at the end-host and thus the server can accurately identify the elephant flows based on the requested content (e.g., large files) from the clients. Figure 1 shows an example of such deployment where the server implements the proposed architecture to manage its first-hop to the Internet (which is the one that is typically prone to congestion).

Paper Organization: Section II puts this work in contrast to other related work. Section III describes our proposed architecture in more detail with all its components and logistics. Section IV captures the dynamics involved through numerical results based on a non-linear fluid model. We evaluate the performance of our proposed architecture with extensive simulation experiments in Section V. In Section VI we present results from our Linux implementation. We conclude the paper in Section VII.

II. Related Work

As hinted in the introduction, many Quality of Service (QoS) mechanisms have been proposed that belong to the general IntServ [11] or DiffServ [12] architectures. Due to their reliance on modifying some in-network components (whether core routers or edge routers), their acceptance for deployment is difficult. Moreover, some of them do require the participation of all entities, which imposes a significant scaling and implementation issues.

The work in [13] focused on managing the end-to-end behavior of TCP connections through sharing congestion information among them. A congestion manager module is used to regulate the transmission rates of the TCP connections in order to achieve an overall better performance. This method, however, does not aim to provide specific guarantees to the TCP connections. In [14] a coordination protocol (CP) is proposed which seeks to optimize cluster-to-cluster communication of computing devices across a bottleneck aggregation point. Their proposal entails, among other aspects, giving the flows across the aggregation point the ability to sense the network state and adapt at the end-points. The authors in [15] propose a Rate Management Protocol (RMP) that controls the rates of the flows passing through an aggregation point based on their QoS requirements. Once the fair share of the flows are decided by the RMP, a new TCP sliding window is used to realize that fair share. This method, however, requires the modification of the current architecture (aggregation and end points) to be realized. In [16], the authors divide the problem of managing QoS into two components; the first one utilizes a probing scheme at the IP layer and the other policies the rates at the edge routers. This method also requires the modification of edge routers.

In [17] an elastic tunnel is created using a number of regular TCP connections to provide soft bandwidth guarantees. The number of connections that form the elastic tunnel is adjusted dynamically in tandem with cross-traffic, so their aggregate throughput ensures the QoS guarantees. This work only considered constant QoS guarantees. Moreover, it required modifications to edge routes to manage the elastic tunnels.

The authors in [18] present an adaptive Forward Error Correction (FEC) scheme that aims to ensure a specific end-to-end rate for video streaming applications using TCP connections. The idea is to adjust the degree of redundancy in packets based on the difference between the achievable

throughput and the required rate, without wasting network bandwidth.

In addition to the above related work, we refer the readers to [19], [20] that provide surveys on bandwidth adaptation and control mechanisms for managing Internet traffic.

III. The Architecture

In this section, we describe the main components and the operation of our proposed service architecture.

A. The Components

We envision a setup composed of a single Free TCP connection and s Stunt TCP connections. Those $s+1$ TCP connections traverse a bottleneck link along with n other TCP connections representing normal cross-traffic. Thus a total of $(1+s+n)$ TCP connections traverse that bottleneck link. Figure 1 depicts this setup. The application running on top of the Free connection requests its throughput requirements through a trace file. This trace file is first checked by a preprocessor for feasibility. If any of the checks fail, another feasible trace is created that is closest to the original trace file; otherwise, the trace file is passed directly to the controller. The controller compares the achievable throughput to the requested throughput over every time instant. Based on the difference, the controller adjusts the ratio of congestion signals to be delegated from the Free connection to the Stunts in order to match the achievable throughput to the requested throughput. A monitor measures the throughput achieved by the Free connection and reports this value back to the controller. Figure 2 represents the different components in the architecture.

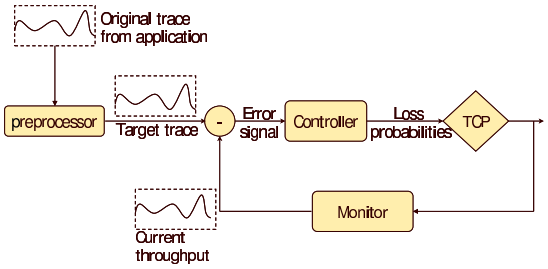


Figure 2. The Components of the proposed architecture.

The trace file: An application specifies its requirements via a trace file. A trace file describes the *shape* of the throughput over time. It is composed of two-tuple entries in the form of *time* and *throughput*. An entry in the form $< i, T_i >$ indicates a request of T_i throughput at time instant i.

The preprocessor: The main goal of the preprocessor is to check the feasibility of the trace file and to create another feasible trace, if any of the checks fail. It performs two checks, a slope check and a region check. The slope check ensures that the requested throughput can be attained from one time instant to the next based on the round-trip time (RTT) of the Free connection. For any two successive time instants, i and j, the requested throughput T_j at time instant j, is bounded by:

$$T_j \leq T_i + \frac{j-i}{RTT^2} \quad (1)$$

Since TCP increases its congestion window by 1 packet every RTT, the congestion window at time j, cannot be more than $\frac{j-i}{RTT}$ of the congestion window at time i. Dividing by RTT to obtain the throughput leads to the above equation. The region check prevents the Stunts from achieving zero throughput. Thus for any time instant i, the requested throughput is bounded by:

$$T_j \leq s \times \bar{x}_s \quad (2)$$

where $\bar{x}_s$ is the average expected throughput per Stunt connection. This is a preliminary check and a more strict check is enforced online.

The controller: The controller decides which losses are picked up by the Free connection versus those delegated to the Stunts. The decision is based on the error signal between the current throughput and the requested throughput. We have experimented with two difference controllers. An On/Off controller and a Proportional Integral (PI) controller. An On/Off controller, decides the delegation percentage g_i, at time i according to the following equation:

$$g_i = \begin{cases} 0 & x_i > 1.3\bar{3}T_i \\ 1 & otherwise \end{cases} \quad (3)$$

where x_i is the instantaneous throughput of the Free connection at time instant i. An On/Off controller will try to delegate all the losses to the Stunts, whenever the current throughput is not matching the requested throughput. Otherwise, the Free connection will pick up its own losses and react to them. Notice that we compare x_i to $1.3\bar{3}T_i$ as opposed to T_i directly. This is due to the AIMD mechanism and the operation of the controller at short time-scales. A packet loss at $1.3\bar{3}T_i$ will cause the throughput to drop to $0.6\bar{6}$, leading to the correct average value of T_i, assuming that the RTT is kept constant.

The above controller may lead to oscillations, thus we experiment with a PI controller that adjusts the delegation percentage based on the following equation:

$$g_i = g_{i-1} + K \times (x_i - 1.3\bar{3}T_i) \quad (4)$$

where K is a constant the decides the aggressiveness of the controller in reaction to the error signal between the

current throughput and the requested throughput. The higher the value of K is, the more aggressive the controller would react.

B. Delegation of Congestion Signals

The Free and the Stunts architecture capitalizes on the fact that a delegation of a congestion signal from the Free connection to the Stunts will both (1) allow the Free connection to achieve a higher target data rate, and (2) it will not violate *any* TCP congestion control rules.

Since the Free connection can delegate congestion signals, it can continue sending as dictated by the Additive Increase component of the AIMD mechanism, by increasing its congestion window by 1 packet every RTT. Since the requested throughput is slope-checked by the preprocessor, then it should be able to achieve the requested target. However, in some conditions (explained below), the Free connection may not be able to delegate a congestion signal and thus it would have to cut its congestion window in half.

It is very important to realize that the impact of a congestion signal is not the same – as far as the network is concerned – whenever it gets delegated, since one connection may have a different congestion window size than the other. Thus, the reduction in the sending rate will not be equivalent.

In our particular case, to make sure that the network sees an equivalent reduction, we first check to see if the total congestion windows (of all the Stunts) is larger than that of the Free connection. If so, then we go through the Stunts, one by one in a round robin fashion, and we halve each one's congestion window, until the total reduction is at least as large as half the Free connection's congestion window size. If not, then we cannot delegate this loss and the Free connection has to react to it in the normal way, since we can only cause each Stunt connection to back-off one time for a given loss. Note that the round robin algorithm may cause the last Stunt connection to decrease its rate by a bit more of what is actually required, since we do not optimize to find the best fit among the Stunt connection's congestion window sizes that would sum to exactly the Free connection's congestion window size. This is wasted bandwidth that is essentially given up by the Stunts to be acquired by all connections. The effect of this is diminished over time as all connections grab more throughput.

The reason we go in round robin fashion is to provide some notion of fairness across the Stunts without hurting any one or group of Stunt connection. Notice that TCP fairness in our case is considered globally across the group of Free and the Stunt connections, as they can be abstractly considered as a single entity. The group of Free and Stunt connections should not together be more aggressive than an equivalent number of ordinary TCP flows when increasing their data flow rate through the normal TCP rules.

Paying back the Stunts: In some situations, the Free connection may not need a higher data rate. Either because the requested throughput at some point in time may go under its fair share or its data rate has increased to a point that is above the requested target rate. In both cases the Free connection can easily give up this bandwidth by having the application send less data. However, this slack of bandwidth will be naturally acquired by all connections (Stunts and cross-traffic). We have modified both controllers described above to allow for the reverse loss delegation from the Stunts to the Free connection. Reverse delegation helps the group of Free and Stunt flows to retain bandwidth as a whole, as apposed to releasing it to the network. Furthermore, it allows the Free connection to closely match its target when the target is low (typically below its fair-share). Also, as discussed above, delegation in this case would ensure that the Free connection would have a larger congestion window than the Stunt that is delegating. Otherwise, the Stunt connection cannot delegate a loss and would have to react to it.

We have chosen to delegate losses during the AIMD behavior, since we focus in this paper on longer data transfers with TCP. It is possible to delegate other behaviors such as timeouts and slow-start, but we do not consider those in this work for reasons having to do mostly with complexity and rareness of those particular events, in comparison to the AIMD behavior, on a well provisioned network.

IV. The Model

We extended a nonlinear fluid model, similar to those proposed in [21], [22], [23], [24], to capture the performance of m TCP flows traversing a bottleneck of capacity C, where m is equal to $(1 + s + n)$ as depicted in Figure 1.

A. Model Derivations

The round trip time $r_i(t)$ at time t for connection i is equal to the round-trip propagation delay D_i between the sender and the receiver for connection i, plus the queuing delay at the bottleneck router. Thus $r_i(t)$ can be expressed by:

$$r_i(t) \quad = \quad D_i + \frac{b(t)}{C} \tag{5}$$

where $b(t)$ is the backlog buffer size at time t at the bottleneck router. We denote the propagation delay from sender i to the bottleneck by D_{s_ib}, which is a fraction α_i of the total propagation delay.

$$D_{s_ib} \quad = \quad \alpha_i D_i \tag{6}$$

The backlog buffer $b(t)$ evolves according to the equation:

$$\dot{b}(t) \quad = \quad \sum_{i=1}^{m} x_i(t - D_{s_ib}) - C \tag{7}$$

which is equal to the input rate $x_i(.)$ from the m connections minus the output link rate. Notice that the input rates are delayed by the propagation delay from the senders to the bottleneck D_{s_ib}.

We assume RED, as proposed in [25], is employed at the bottleneck link as an active queue management scheme. Thus, the congestion loss probability $p_c(t)$ is given by:

$$p_c(t) = \begin{cases} 0 & v(t) \leq B_{min} \\ \sigma(v(t) - \varsigma) & B_{min} < v(t) < B_{max} \\ 1 & v(t) \geq B_{max} \end{cases} \quad (8)$$

where σ and ς are the RED parameters given by $\frac{P_{max}}{B_{max}-B_{min}}$ and B_{min}, respectively, and $v(t)$ is the average queue size, which evolves according to the equation:

$$\dot{v}(t) = -\beta C(v(t) - b(t)), \qquad 0 < \beta < 1 \quad (9)$$

Notice that in the above relationship, we multiply β by C since RED updates the average queue length at every packet arrival, whereas our model is a fluid model as indicated in [21], [23].

The loss delegation between the Free and the Stunts causes them to pick up different congestion signals than those set by RED. In particular, the Free connection, upon delegating $g(t)$ of its congestion signals, would pick up:

$$q(t) = p_c(t) - g(t) \quad (10)$$

Each Stunt connection would pick up:

$$q(t) = p_c(t) + \frac{g(t)}{s} \quad (11)$$

The normal cross-traffic are not affected and will simply pick up:

$$q(t) = p_c(t) \quad (12)$$

The throughput of TCP, $x_i(t)$ is given by

$$x_i(t) = \frac{w_i(t)}{r_i(t)} \quad (13)$$

where $w_i(t)$ is the size of the TCP congestion window for sender i .

According to the TCP Additive-Increase Multiplicative-Decrease (AIMD) rule, the dynamics of TCP throughput for each of the m connections can be described by the following differential equations:

$$\begin{aligned} \dot{x}_i(t) &= \frac{x_i(t - r_i(t))}{r_i^2(t)x_i(t)}(1 - q(t - D_{bs_i}(t))) \\ &- \frac{x_i(t)x_i(t - r_i(t))}{2}(q(t - D_{bs_i}(t))) \\ i &= 1, 2, .., m \end{aligned} \quad (14)$$

where $q(.)$ is the congestion signals observed by each connection based on its type. The first term represents the additive increase rule, whereas the second term represents the multiplicative decrease rule. Both sides are multiplied by the rate of the acknowledgments coming back due to the last window of packets $x_i(t - r_i(t))$. In the above equations, the time delay from the bottleneck to sender i, passing through the receiver i, is given by

$$D_{bs_i}(t) = r_i(t) - D_{s_ib} \quad (15)$$

Mode Assumptions: The model above makes the following assumptions: (1) It ignores the effect of slow-start and timeout mechanisms of TCP, since our main focus is on the AIMD. (2) The delegation of some losses can be distributed in a linear fashion among the Stunts (as indicated in Equation 11). In general, this does not hold except for small value of losses, since the throughput is inversely proportional to the square-root of the loss probability. Despite these assumptions, however, the model above still captures the main dynamics as we illustrate below.

B. Numerical Results

We instantiate the model above with specific parameters and we solve it iteratively. We assume there is 1 Free connection, 4 Stunts and 15 cross-traffic, for a total of 20 connections. The bottleneck has a capacity 2000 packets/sec. The RTT for each connection is chosen at random around 100 msec.

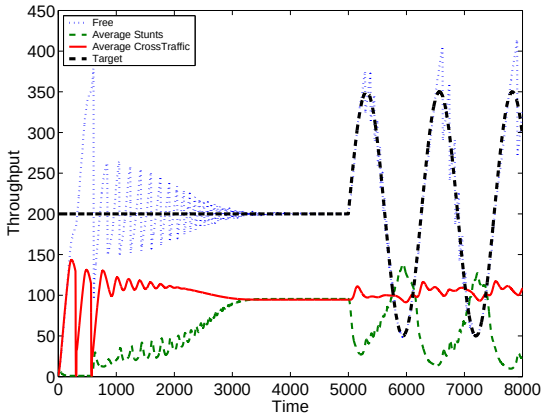


Figure 3. Numerical Results.

Figure 3 illustrates the performance of the Free connection in matching a target trace that starts with constant throughput at 200 packets/sec and then follows a sin wave. The figure also shows the average throughput across the stunts as well as the average throughput across the cross-traffic connections. One can observe how the Stunt connections make room for the Free connection to match the target throughput. Notice also, how little the normal cross-traffic is affected, except for the initial startup time (the first 3 seconds) where the whole system is still in a transient behavior. One can also see the impact of reverse delegation around time 6000.

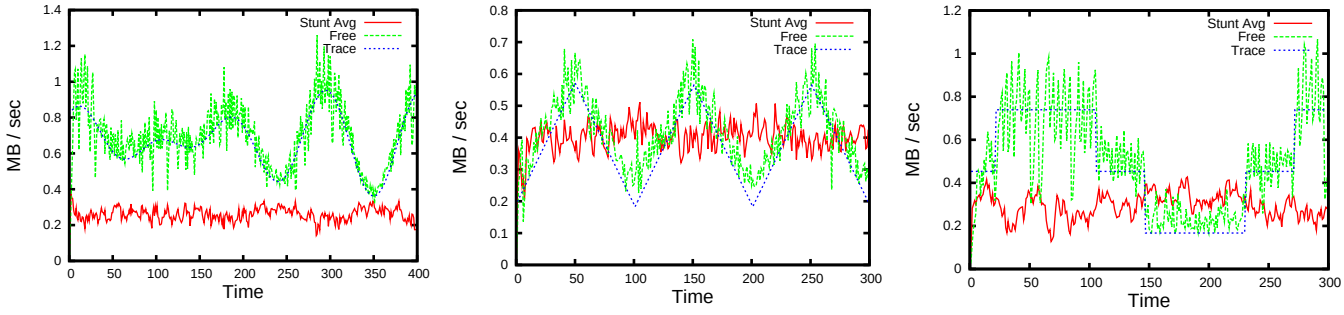


Figure 4. Three simulation traces to assess the Free connection's throughput in matching the target throughput.

Since the target throughput drops below the fair-share (100 packets/sec), the Stunts can delegate congestion signals to the Free connection and thus they are able to increase their throughput a bit above their fair-share.

V. Simulation Experiments

We have implemented our proposed architecture in NS-2 [26]. In this section, we study the performance of our proposed architecture under differing environments (topologies and trace files) and parameters.

The Setup: Figure 1 depicts the general topology of the simulated network. It is composed of a single bottleneck link that is traversed by the Free, Stunts, and cross-traffic connections. We assume all connections have an infinite supply of data to transmit. However, to study the impact of different dynamics that arise in practice, a number of the cross-traffic connections are turned on and off at varying times during a given simulation run. We also vary the number of Stunt connections to demonstrate and examine the behavior of our proposed architecture under different congestion levels.

The bottleneck link is configured with RED [25]. The queue size at the bottleneck link is chosen to be $\frac{RTT \times C}{\sqrt{m}}$ as advocated in [27], where C is the bottleneck link capacity and m is the total number of connections traversing the bottleneck. This is because on fully utilized simulation networks, those composed of only long lived TCP flows, the justification for dividing by the square root of m breaks down due to synchronization of network flows [27] [28]; however, when randomized cross traffic flows are added the premise is regained for the reduction in the buffer size. The RED parameter B_{min} is set to 0.25 the size of the queue resulting in the distance between B_{min} and B_{max} being three times B_{min}. Other parameters were chosen to encourage the stability of the average queue size.

Performance Metrics: To measure the effectiveness of our proposed architecture in matching the achievable throughput to the requested throughput, we propose a weighted variant of the standard "sum-of-squared errors" method. The main problem with the standard "sum-of squared errors" is that it does not differentiate between the case where the achieved throughput is above the target, versus the case where the achieved throughput is below the target (since in both cases we may get the same value). So we define the positive variance V^+ to be:

$$V^+ = \frac{\sum(x_i - T_i)^2}{C^+} \quad \forall \; x_i > T_i \tag{16}$$

where C^+ is the number of times (sample points) the achieved throughput is above target. Similarly, we define the negative variance V^- to be:

$$V^- = \frac{\sum(x_i - T_i)^2}{C^-} \quad \forall \; x_i < T_i \tag{17}$$

where C^- is the number of times (sample points) the achieved throughput is below target. To capture the overall performance we use a weighted variance, defined by:

$$V^* = \delta \times \frac{\sum(x_i - T_i)^2}{C^+ + C^-} \tag{18}$$

where δ is a ratio that is given by:

$$\delta = \frac{max(V^+, V^-)}{min(V^+, V^-)} \tag{19}$$

where δ is always greater than or equal to 1. If the matching is achieved ideally, then δ would be 1. A larger value of δ indicates a bias in the matching, either above or below the target, and this would increase the weighted variance in turn. The above metrics are computed over an entire simulation experiment.

A. Matching the Target Throughput

This set of experiments assess the ability of the Free connection in matching its throughput to different target trace files.

Figure 4 shows representative results using three different trace files with three different parameters. All results were obtained using a PI controller and with 10 Stunt connections. In each one, we plot the target trace, the throughput of the Free connection and the average throughput across all Stunts. Figure 4 (left) was obtained using a topology with 40 Mbps

bottleneck link capacity. In order to provide some variability, 8 of the cross-traffic connections through the bottleneck were randomized at ten second on/off intervals with the exception of 2 cross-traffic flows, which were continuous. This experimentation ensured that the flows through the bottleneck did not experience many timeouts.

Figure 4 (middle and right) were obtained using a topology with 80 Mbps bottleneck link capacity. The number of cross-traffic links was kept constant at 20 connections. Overall, one can see that the Free connection does a fairly good job in matching the target throughput while the throughput achieved by the Stunts changes in tandem. Notice the larger oscillations at higher data rates; these are expected due to the normal behavior of the AIMD mechanism. We see the opposite effect at lower data rates, due to a smaller decrease in bandwidth.

Notice also that the reverse delegation of losses from the Stunts to the Free connection allows the Stunts to achieve higher rates than they would have achieved otherwise. The slack of bandwidth given up by the Free connection goes directly to the Stunts as opposed to going to the Stunts and the cross-traffic. This is evident from Figure 4 (right) from time 150 until around 235 seconds. During this time interval, the Stunts are achieving higher throughput than their fair share, since the Free connection does not need that throughput. This also confirms our numerical results in Figure 3.

This experiment makes it clear that the Free flow can acquire a variety of target waveforms. Experimentation showed that the limiting factors were virtually all related to network latency and congestion. We understand this to be due to the fact that the architecture is designed around TCP congestion signals. Naturally, the ability of the Free flow to achieve a requested target rate is dependent on the ability of the network to support that link utilization. As was mentioned earlier, Figure 4 (left) is an example of the Free flow simulated in a well provisioned network with moderate cross-traffic dynamics and link utilization. That figure also shows very good fitness with regard to the target waveform.

B. Impact of the Number of Stunt Connections

To study the impact of the number of Stunt connections on the performance of the Free connection, we vary the number of Stunts while holding all other parameters constant and we plot the weighted variance (as given in Equation 18) versus the number of Stunts. As mentioned in Section I, these Stunt connections already exist due to the normal operation of the server. We do not advocate creating them to make this architecture work.

Figure 5 shows the results obtained (the non-random cross-traffic plot), where each point represent an independent simulation run. One can observe that there is an optimal number of Stunts (around 5 or 6) that minimizes the

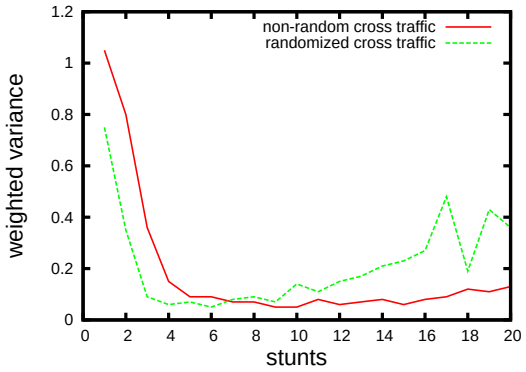


Figure 5. Impact of the number of Stunts on the weighted variance for non-random cross traffic and randomized cross-traffic.

weighted variance. Increasing the number of Stunts further, not only shows diminishing returns but also harms the performance of the Free connection as indicated by a slight increase in the weighted variance towards the higher number of Stunts.

Figure 6 shows the exact performances with 2, 10 and 20 Stunts, respectively. These plots were generated on topology of an 80 Mbps bottleneck link with 20 cross-traffic connections. Clearly, a very small number of Stunts (2) has a noticeable degrading effect on the performance of the Free connection, which improves with an increased number of Stunts up to a point where is starts decreasing again.

The number of Stunts affects the overall efficiency of this method because Stunts act more than just being a reservoir of bandwidth for the Free connection. In particular, they adjust the level of congestion in the network for the Free connection to better match the target throughput. If their number is very low, the Free connection would not be able to delegate losses (since we strictly enforce the same reduction in congestion windows from all Stunts). If their number is very large, the network would be more congested and all flows would go into timeouts/slow-start, which may prevent the Free connection to match the target throughput. One approach to handle this case would be to delegate timeouts, however, our focus in this paper was mainly on the AIMD mechanism as explained earlier.

C. Impact of Cross-traffic Dynamics

To study the impact of dynamics that arise in practice, we allow a number of the cross-traffic connections to be turned on and off at random times during a given simulation run. The cross-traffic flows are turned on and off every 10 seconds randomized with a uniform distribution.

Figure 7 shows the impact of varying the number of the cross-traffic connections, while keeping the number of Stunts steady at 10. We plot the positive variance, the negative variance, and the weighted variance to better explain a unique behavior observed in this experiment.

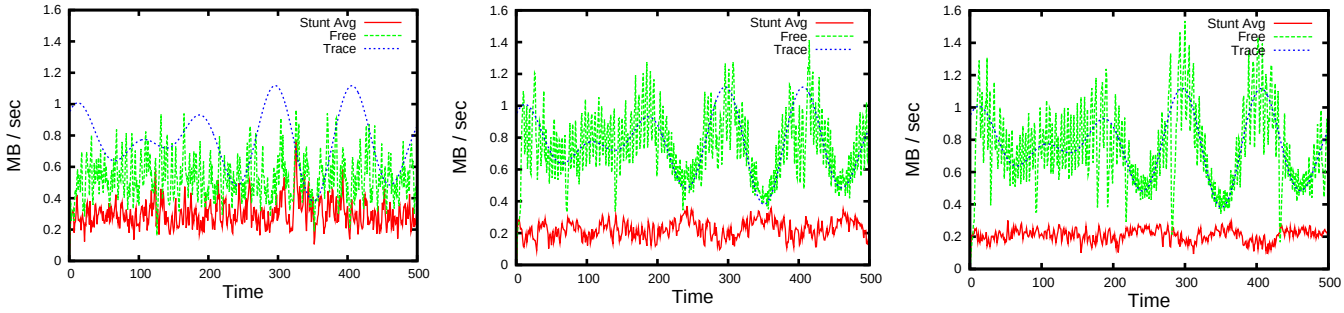


Figure 6. Impact of the number of Stunt connections on the performance. Left plot with 2 Stunts, middle plot with 10 Stunts and right plot with 20 Stunts.

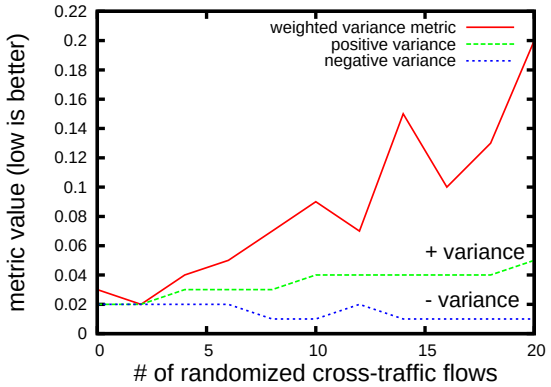


Figure 7. Impact of increasing the level of dynamic cross-traffic connections.

It is clear that the higher the dynamics from the randomized cross-traffic, the harder it is for the Free connection to match the target. However, when we examined the exact performance of the Free connection, we found that its shape was rather intact than deformed, but it was above the requested target. This was confirmed visually in each simulation run and is easy to see by examining the positive and negative variance metrics. Notice how the positive variance grows larger as the negative variance grows smaller. Such divergence increases the weighted variance due to a higher δ. The throughput of the Free connection was above the target because with a larger number of randomized cross-traffic, their combined throughput decreased the utilization at the bottleneck. This caused the Free connection to acquire more than the target because the lower link utilization also resulted in a lower packet loss probability.

Figure 5 (the randomized cross-traffic plot) shows the impact of the number of Stunts when most of the cross-traffic connections (19 out 20) were randomized on a 10-second on/off intervals. The presence of dynamics, coupled by an increase in the Stunts lead to a degraded matching between the the throughput of the Free connection and the target throughput.

D. On the Feasibility of the Free and the Stunts Architecture

As hinted in Section I, Internet measurement studies have indicated that around 80% of the traffic is controlled by a small number of connections (elephants) while the majority of connections (mice) control only around 20% of the traffic [4], [5], [6]. This is a direct effect of the heavy-tailed nature of file sizes on the Internet.

To study the execution of our proposed architecture in an environment with such traffic properties, we used a simulation network that is composed of 1 Free flow and 3 Stunt flows, all are elephant flows. In addition, the cross-traffic connections consisted of 4 elephant flows and 24 mice flows. Although, we initially classified flows as being elephants versus mice, such classification can be achieved dynamically as indicated in [29]. The bottleneck link is 10 Mbit with 10 ms latency. All links into and out of the bottleneck are 100 Mbit with 2 ms latencies. All the flows have an on/off state with a Pareto distribution with shape 1.5. The elephants have a mean burst time of 120 seconds and idle time of 5 seconds. The mice have a mean burst time of 1 second and idle time of 5 seconds. The parameters were chosen to produce a close distribution of traffic between elephants and mice that is close to the 80%-20% rule observed on the Internet. We limit the choice of Stunts to elephant flows.

Figure 8 (left) shows the results of an experiment run using the above parameters. We observe that, even with the presence of dynamics across all flows, utilizing the elephants as Stunts achieves a good matching between the Free connection's throughput and the target trace. That is because they control a large percentage of the capacity and are able to accept congestion signals delegations from the Free connection. Figure 8 (right) shows the results of a different experiment where the Free flow is turned on and off (we show the trace only in the on period of the Free flow). The elephant flows have a mean burst time of 10 seconds and idle time of 5 seconds. The mice flows have a mean burst time of 300 msec and 13 seconds. In this experiment, there were 10 elephant connections (only 4 were used as Stunts) and 50 mice connections. Again the parameters were chosen to produce the 80%-20% rule between elephant and mice. Utilizing few elephant flows as Stunts results in a good matching.

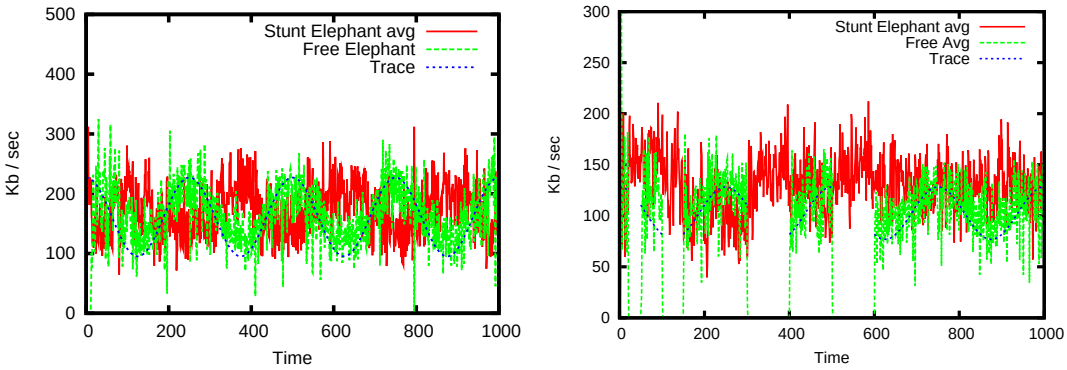


Figure 8. Simulation results with Pareto distribution. Elephants and mice consume around 80% and 20% of the bottleneck's capacity, respectively. Right plot considers the Free connection turning on and off as well.

VI. Implementation Experiments

We have implemented the Free and the Stunts architecture as a pluggable module for the 2.6.20 Linux kernel. We have chosen to experiment with the On/Off controller since the Linux kernel does not natively support floating point operations (implementing the PI controller is harder since it requires changing all math operations to fixed-point ones).

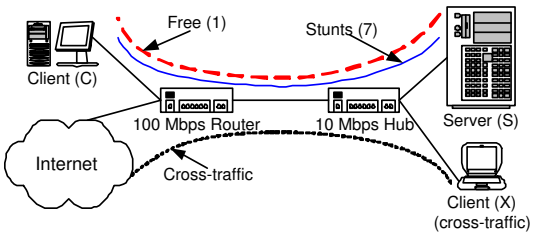


Figure 9. The experimental setup used in our implementation experiments.

The Setup: Figure 9 shows the experimental setup used in our implementation experiments. It is composed of a Server (S) and two Clients (C and X). The Server runs Linux and implements the Free and Stunts pluggable module. The server runs an application that accepts TCP connections from clients and serves them continuous flows of data. We have created a bottleneck link of 10 Mbps at the Server's first hop. All other links are 100 Mbps. All experiments were composed of 1 Free connection and 7 Stunt connections, from Client (C) to the Server. To simulate cross-traffic on the bottleneck link, we have manually opened and closed connections between Client (X) and different Internet web-servers.

A. Matching the Target Throughput

Our first set of experiments illustrates the performance of the free connection in matching the requested target. Figure 10 shows two different traces. Each plot shows the average throughput over four independent runs. We also plot the requested target as well as the adjusted target (1.33 of the requested target). One can observe that on average there is a very close matching between the throughput and the requested target.

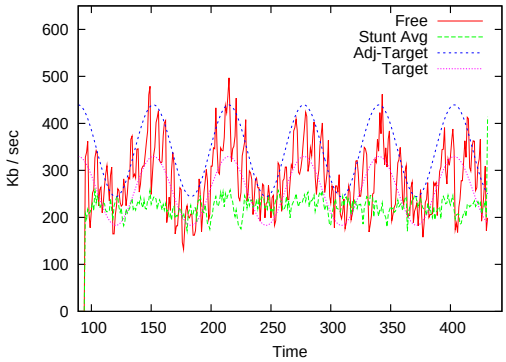


Figure 11. Free connection Performance with cross traffic introduced at time 330.

Figure 11 shows the performance of a single Free connection in matching the target throughput. One can see that the implementation does relatively well in matching the target. At around 330, we manually opened several connections from Client (X) to www.youtube.com over a period of 50 seconds and downloaded some videos in order to introduce cross traffic on the bottleneck link. One can see the effect of those cross-traffic connections on the performance on the Free connection as indicated by few misses in matching the target throughput.

B. Improving the Reverse Delegation

To improve the matching between the throughput of the Free connection and the target rate, we have modified the "reverse" delegation so that the Free connection does not drop its congestion window more than necessary. Recall that delegating a loss from a stunt connection to the free connection would cause the free connection to drop is congestion window by half. Here we modified such adjustment so that

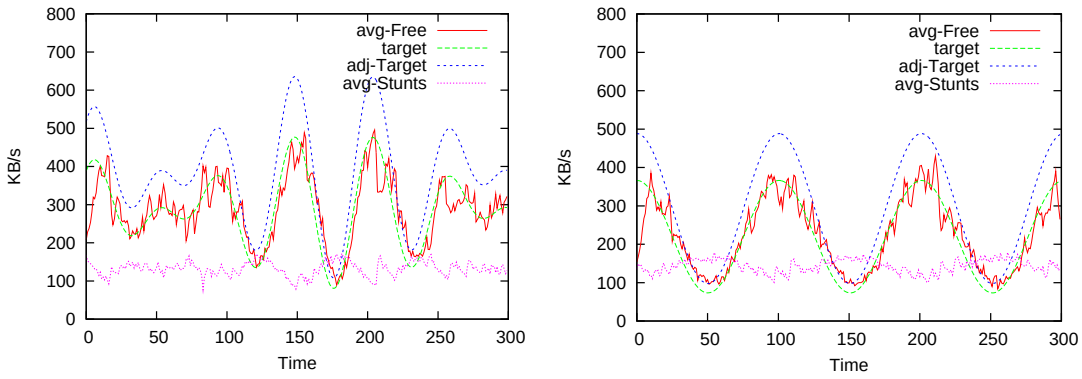


Figure 10. Implementation results for target matching.

the free connection would drop its congestion window by *the exact value* the delegating stunt would have experienced, if it were to pick up this congestion signal.

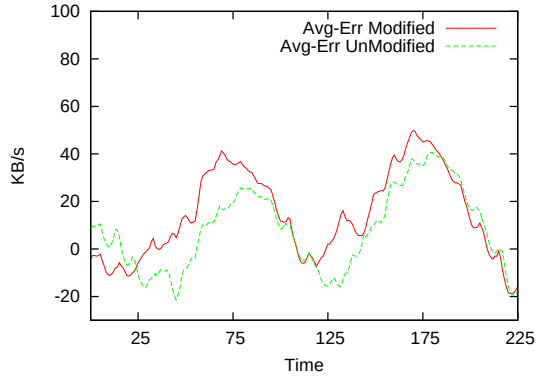


Figure 12. Improving the reverse delegation.

Figure 12 depicts the trending behaviors of the modified reverse delegation method versus the usual method, on the error signal obtained as the difference between throughput achieved and the target. The plotted lines are the average error values over several runs.

One can observe from Figure 12 that the trending of the targeting error is higher in certain areas of the plot. These areas are predictably located where reverse delegation has a higher probability of occurring (for example, the throughput of the Free flow needs to be reduced to meet the target). The increased error means that the Free flows throughput is trending closer to the adjusted target, rather than the requested target. We have experimented with different target traces and they show the same types of trends. This behavior was expected since the 1.33 target adjustment was computed based on the exact halving of the congestion window, due to normal TCP Congestion Control AIMD behavior. One can mitigate this effect by having a closer adjusted target to the requested target.

VII. Conclusion

This paper describes an architecture that enables a set of clients to delegate and trade congestion signals between them in order to shape traffic based on demands from the applications. The architecture strictly adheres to TCP rules and does not affect network congestion on the bottleneck links. Moreover, no modifications is required to any devices along the communication path (unless an ISP decides to implement this architecture to manage flows). We have shown that this service architecture is capable of providing a reasonably accurate targeting between the achieved rate and the target rate with a small number of Stunts. We have assessed the performance of our proposed architecture through new metrics, using numerical solutions, extensive simulation experiments and real implementation in the Linux kernel.

References

[1] J. Valdez and M. Guirguis, “Liberating TCP: The Free and the Stunts,” in *Proceedings of the 7th International Conference on Networking (ICN 2008)*, Cancun, Mexico, April 2008.

[2] V. Cerf and L. Kahn, “A Protocol for Packet Network Interconnections,” *IEEE Transactions on Communications*, vol. 22, no. 5, June 1974.

[3] V. Jacobson, “Congestion Avoidance and Control,” in *Proceedings of ACM SIGCOMM*, Stanford, CA, August 1988.

[4] K. Thompson, G. Miller, and R. Wilder, “Wide-Area Internet Traffic Patterns and Characteristics,” *IEEE Networks*, vol. 11, no. 6, 1997.

[5] S. Fred, T. Bonald, A. Proutiere, G. Régnié, and J. Roberts, “Statistical Bandwidth Sharing: A Study of Congestion at Flow Level,” in *Proceedings of ACM SIGCOMM*, San Diego, CA, August 2001.

[6] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and S. Diot, “Packet-level Traffic Measurements from the Sprint IP Backbone,” *Network, IEEE*, vol. 17, no. 6, pp. 6–16, 2003.

[7] L. Guo and I. Matta, "The War between Mice and Elephants," in *Proceedings of IEEE ICNP*, Mission Inn, Riverside, November 2001.

[8] C. Barakat and E. Altman, "Performance of Short TCP Transfers," *Lecture Notes in Computer Science*, pp. 567–579, 2000.

[9] M. Mellia, H. Zhang, and D. di Elettronica, "TCP Model for Short Lived Flows," *IEEE Communications Letters*, vol. 6, no. 2, pp. 85–87, 2002.

[10] J. Londono, A. Bestavros, and N. Laoutaris, "Trade and Cap: A Customer-Managed, Market-Based System for Trading Bandwidth Allowances at a Shared Link," *Technical Report BUCS-TR-2009-025, CS Department, Boston University*, 2009.

[11] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An Overview," *RFC 1633, 1994*, June 1994.

[12] S. Blake, D. Black, M. Carlson, E. Davies, Z.Wang, and W. Weiss, "An Architecture for Differentiated Services," *IETF RFC 2475, 1998*, December 1998.

[13] H. Balakrishnan, H. Rahul, and S. Seshan, "An Integrated Congestion Management Architecture for Internet Hosts," in *Proceedings of ACM SIGCOMM*, Cambridge, MA, August 1999.

[14] D. Ott and K. Mayer-Patel, "An Open Architecture for Transport-level Protocol Coordination in Distributed Multimedia Applications," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), 2007*, August 2007.

[15] Z. Rosberga, J. Matthewsa, and M. Zukerman, "A Network Rate Management Protocol with TCP Congestion Control and Fairness for All," *Elsevier Computer Networks*, vol. 54, no. 9, June 2010.

[16] Z. Rosberg, J. Matthews, C. Russell, and S. Dealy, "Fair and End-to-End QoS Controlled Internet," in *Proceedings of 3rd International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ)*, Athens, Greece, June 2010.

[17] M. Guirguis, A. Bestavros, I. Matta, N. Riga, G. Diamant, and Y. Zhang, "Providing Soft Bandwidth Guarantees Using Elastic TCP-based Tunnels," in *Proceedings of the 9th IEEE Symposium on Computer and Communications*, Alexandria, Egypt, July 2004.

[18] T. Tsugawa, N. Fujita, T. Hama, H. Shimonishi, and T. Murase, "TCP-AFEC: An Adaptive FEC Code Control for End-to-End Bandwidth Guarantee," in *Proceedings of 16th International Packet Video Workshop*, Lausanne, Switzerland, November 2007.

[19] P. Siripongwutikorn, S. Banerjee, and D. Tipper, "A Survey of Adaptive Bandwidth Control Algorithms," *IEEE Communications Surveys and Tutorials*, vol. 5, no. 1, pp. 14–26, 2009.

[20] H. Wei and Y. Lin, "A Survey and Measurement-based Comparison of Bandwidth Management Techniques," *IEEE Communications Surveys and Tutorials*, vol. 5, no. 2, 2009.

[21] C. Hollot, V. Misra, D. Towsley, and W. Gong, "A Control Theoretic Analysis of RED," in *Proceedings of IEEE INFOCOM 2001*, Anchorage, AL, April 2001.

[22] F. Kelly, "Mathematical Modelling of the Internet," *Mathematics Unlimited and Beyond, 2001*, pp. 685–702, 2001.

[23] S. Low, F. Paganini, J. Wang, S. Adlakha, and J. Doyle, "Dynamics of TCP/RED and a Scalable Control," in *Proceedings of IEEE INFOCOM*, New York, NY, June 2002.

[24] S. Shenker, "A Theoretical Analysis of Feedback Flow Control," in *Proceedings of ACM SIGCOMM*, Philadelphia, PA, September 1990.

[25] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *Transactions on Networking*, vol. 1(4), pp. 397–413, August 1993.

[26] E. Amir and et al., "UCB/LBNL/VINT Network Simulator - ns (version 2)," available at http://www.isi.edu/nsnam/ns/.

[27] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing Router Buffers," in *Proceedings of ACM SIGCOMM*, Portland, Oregon, August 2004.

[28] L. Zhang, S. Shenker, and D. Clark, "Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two-Way Traffic," in *Proceedings of ACM SIGCOMM*, Zurich, Switzerland, September 1991.

[29] K. Avrachenkovt, U. Ayesta, P. Brown, E. Nyberg, and F. INRIA, "Differentiation between Short and Long TCP Flows: Predictability of the Response Time," in *Proceedings of IEEE INFOCOM*, vol. 2, 2004.

A Further Look at the Distance-Availability Weighted Piece Selection Method

A BitTorrent Piece Selection Method for On-Demand Media Streaming

Petter Sandvik and Mats Neovius
Department of Information Technologies
Åbo Akademi University
and
Turku Centre for Computer Science
Turku, Finland
e-mail: {petter.sandvik, mats.neovius}@abo.fi

Abstract—**During the last few years, BitTorrent has become a popular way of transferring large files over the Internet. However, the original out-of-order nature of the BitTorrent protocol has made it difficult to enable playback of media files that have not yet been fully transferred. In this paper we describe a piece selection method which we believe will enable simultaneous playback of the transferred media file without impacting on the speed and quality of the transfer. The distance-availability weighted method compromises between selecting rare pieces and pieces which are soon to be played back, making playback possible before the transfer is complete. In our simulations, we have compared our piece selection method with other proposals for on-demand streaming media using a BitTorrent-like setup, with our method giving similar or better results.**

Keywords-media, on-demand, peer-to-peer, streaming, BitTorrent, simulation

I. INTRODUCTION

In [1], we presented the distance-availability weighted method; a BitTorrent piece selection method for on-demand streaming. BitTorrent is originally a peer-to-peer file sharing protocol and an application, designed by Bram Cohen and first released in July 2001 [2]. During the following years BitTorrent evolved into one of the most popular peer-to-peer protocols [3][4]. Unlike earlier popular peer-to-peer file sharing applications such as Napster and Kazaa, the use of BitTorrent usually starts by clicking a link in a web browser, and Cohen suggests that "ease of use has contributed greatly to BitTorrent's adoption, and may even be more important than [...] the performance and cost redistribution features" [5]. Another major difference between the earlier peer-to-peer file sharing applications and BitTorrent is the lack of central server in the latter, in the sense that BitTorrent users are not all connected to each other through one server.

While BitTorrent is popular, there are also other reasons for choosing it as the basis for a peer-to-peer based media streaming system such as the one we have in mind. Several applications using the protocol, as well as the protocol itself, are open, which makes understanding and modifying more easily possible than if the staring point was a proprietary product. For our purposes an even more important aspect is that all the logic involved in the file transfer is contained on the client side, which makes it possible, at least in theory, to have an on-demand content streaming BitTorrent client participate in content transfer with regular, non-streaming, BitTorrent clients.

While the original BitTorrent protocol was not designed for streaming, it has already been argued [6] that with some modifications, it would be possible to create a streaming media solution based on BitTorrent. Indeed, there are proprietary and commercial efforts to create exactly such a thing, for instance "to turn BitTorrent into a point-click-watch experience much more similar to YouTube" [7], and in [1] we described our proposal for a solution that would enable file sharing in such a way that content could be played back while downloading. In this paper we will further describe our proposal and show that it is a modification to the BitTorrent protocol, which would allow it to function as an on-demand media streaming solution.

The rest of this paper is organised as follows: in Section II, we describe BitTorrent and its terminology, as used throughout this paper. Section III consists of the requirements of an on-demand streaming media application in general, and what assumptions we will make for a BitTorrent streaming application to be feasible. In Section IV we present our proposed piece selection method, and in Section V, we compare it to existing piece selection methods that could be used for on-demand streaming. In Section VI, we list a selection of related works. The paper is concluded in section VII with a discussion about our findings and possible future work on this subject.

II. BITTORRENT

Since BitTorrent was introduced by Bram Cohen in 2001 it has evolved. While the terminology used in [5] could be seen as a standard, the terminology used in this paper will be based on that of [8]. This terminology is close to that of the application Vuze, formerly known as Azureus, a BitTorrent client often used as a basis for research applications [9][10][11].

A. BitTorrent Terminology

- **Torrent**. A torrent consists of a single file, or a collection of files, to be shared, and the associated metadata. The metadata, and therefore the torrent itself, is uniquely identified by its info-hash [12].

- **Tracker**. A tracker is a piece of software that is involved in keeping track of which peers are involved in the transfer of a particular torrent, using the info-hash of the torrent. Each torrent can be associated with many trackers. Additions to the BitTorrent protocol have enabled peer discovery through other means, such as distributed hash tables and peer exchange, and thus the use of trackers is no longer required. Whether a tracker is used or not does not affect the file transfer, and this is of little importance to us.
- **Pieces** and **blocks**. The data of a torrent is divided into pieces, and each piece is divided into blocks, also called sub-pieces. A piece is typically 256 kilobytes in size [8][13] while a block is typically 16 kilobytes in size [5][8]. A piece must be complete, that is, all blocks of it must have been downloaded, before the piece can be transferred to another peer.
- **Torrent index file**. Also known as a ".torrent" [5], a torrent index file contains information about the torrent, such as the universal resource locator (URL) of the tracker (or trackers, if any), piece size of the torrent, names and sizes of the files in the torrent, as well as SHA-1 hashes of all the pieces. This file is generally hosted on a web server and downloading of a torrent starts by opening the file with a BitTorrent application.
- **Interested** and **to choke**. If peer B has pieces, which peer A does not have, peer A is interested in peer B, otherwise peer A is uninterested in peer B. If peer B decides not to send data to peer A, peer B chokes peer A, and if peer B decides to send data to peer A, peer B unchokes peer A.
- **Peer set** and **active peer set**. A peer set consists of all the other peers one peer is connected to, and the active peer set consists of those peers it is currently sending data to, i.e., its unchoked peers.
- **Seed**. A peer that has all pieces of a torrent and therefore only sends data is called a seed.
- **Availability**. We define availability of a piece as the number of peers in the peer set who have that specific piece.

B. How BitTorrent Data Transfer Works

When transferring data, BitTorrent needs to decide what data to request (piece selection) and which peers to choke or unchoke (peer selection). The reference BitTorrent implementation begins by selecting pieces to download at random, until one complete piece has been downloaded. BitTorrent then switches to a rarest-first piece selection method. The rarest-first selection method selects the piece that the fewest peers have, i.e., the piece with the lowest availability, as the first piece to request. This has the effect of reducing the possibility that one piece may become unavailable, as a lower number of peers having a piece makes other peers more likely to request that particular piece, thereby increasing the number of peers that will have that piece. When a single block from a piece has been downloaded, other blocks from that piece are given highest priority, in order to have as few incomplete pieces transferred as possible. BitTorrent then keeps selecting the rarest pieces first, until all remaining blocks in all remaining pieces have been requested, at which time all remaining blocks are requested from all peers in the active peer set. This is done so that one slow peer cannot prevent the whole download from completing.

To create an incentive for peers to upload as well as download, the reference BitTorrent implementation uses a tit-for-tat (TFT) peer selection strategy. Every ten seconds, which peers to unchoke is evaluated based on the rate of data sent, with the fastest peers chosen as the peers to unchoke. Additionally, there is also one interested peer unchoked at random, re-evaluated every thirty seconds. This randomly chosen peer is called the optimistic unchoke [5][8] and exists for two reasons: to allow new peers a chance to enter the TFT game, and to potentially discover faster peers that could become regular, non-optimistic, unchokes [8][9]. This approach is not necessarily the optimal way of peer selection, and alternative approaches have been suggested, such as [9]. However, the basic TFT strategy remains an essential part of the BitTorrent protocol as used today.

After a download is finished, the peer may continue to participate in sending data to other peers, and in many cases the peer is actually encouraged to do so. In this case choosing peers based on how much they send is of course not possible, and thus the reference BitTorrent implementation then switches to sending to the peers that can receive data the fastest [5]. However, this feature could be exploited, and later clients have switched to choosing peers to send to randomly [9].

III. REQUIREMENTS FOR STREAMING

We define streaming as the transport of data in a continuous flow, in which the data can be used before it has been received in its entirety. In this context, on-demand streaming is essentially playback, as a stream, of pre-recorded content, at the request of a user. This is in contrast to live streaming, which is playback, as a stream, of content, which is not pre-recorded but rather created and transmitted practically simultaneously. Concerning an on-demand peer-to-peer media streaming system, we make the following initial observations:

Each peer must have a download bandwidth at least as large as the playback bit rate of the media. Unlike a peer-to-peer file sharing system, the media is played while being received, and therefore cannot be received slower than it should be played back. Furthermore, if the media is to be received faster than real-time, it must also be sent faster than real-time. Therefore, the average upload bandwidth of all peers must be larger than the playback bit rate of the media, although an individual peer may have an upload bandwidth smaller than that.

We will make the following assumptions regarding a BitTorrent-based on-demand media streaming system: The torrent will consist of only one complete media file, unlike in file sharing where several files can be combined in one torrent. Additionally, in a file sharing system the time an

individual peer participates is difficult to estimate and not dependent on the content received. In an on-demand peer-to-peer streaming system it can be estimated more easily, and although peers may of course leave the system at any time, our assumption is that a typical peer will enter the system when starting playback and leave the system some time after playback is complete, which means some time after all pieces of the content have been received. We will also assume that for each piece there will always be at least one peer holding it, that is, we assume that we do not encounter the situation where the availability of any piece is zero, because that would lead to a situation where complete playback is not possible.

Internet connections are typically symmetric, with the same bandwidth available for sending and receiving data, or asymmetric with a higher download bandwidth than upload bandwidth. Therefore, a typical peer will be able to receive data at least as fast as it can send data. Additionally, all pieces must be requested, resulting in a complete file once the transfer is complete, unless the peer leaves before finishing playback. Furthermore, each peer will keep and make available all the pieces it has received, for as long as the peer is participating. Each peer must therefore have enough space to store the entire media file.

Ideally, the piece selection algorithm in our BitTorrent-based streaming system should comprise the following behaviours: When the ratio of seeds to peers is high, our piece selection method should prefer pieces close to being played back rather than rare pieces, because with a large number of complete sources there is no need for downloading rare pieces to ensure the future availability of all pieces. In the extreme case, where our client has the only incomplete copy of the content, there is no obvious downside to requesting pieces sequentially. On the other hand, when the ratio of seeds to other peers is low, our piece selection method should also choose rare pieces to improve the overall availability of the pieces, while still requesting enough pieces in sequence to make continuous playback possible.

Although we specified earlier that peers should be able to download faster than the playback rate of the media, there will be variations in how much faster the peers will be able to receive data. Ideally, our piece selection method will be able to adapt to these kinds of differing conditions. For instance, if a peer can download data only slightly faster than the playback rate, our piece selection method should ensure that the peer requests data mostly sequentially, while a peer that can download the media much faster than its playback speed should also frequently request rare pieces in order to improve the overall availability of the pieces. With that in mind, we present our proposal for a piece selection method for on-demand streaming.

IV. THE DISTANCE-AVAILABILITY WEIGHTED METHOD

The idea behind the distance-availability weighted method for piece selection (DAW) is to strike a balance between lots of consecutive pieces, which is good for playback, and requesting rarest pieces first, which is good for piece availability. In other words, we want to balance requesting between distance (in the sequence of pieces) and availability.

We start by having a small, fixed buffer of size k. The priority for requesting the pieces in the buffer will be the highest, here represented as 1. Outside the buffer we will calculate the priority for each not yet requested piece as

$$\text{Priority} = 1/((P_r - P_c) * m_r)$$

where P_r is the sequence number of a particular piece, m_r is the number of peers who hold that piece, and P_c is the sequence number of the current last piece in the buffer. In other words, $P_r - P_c$ is the distance to a particular piece and m_r is the availability of that piece. The priority for a piece outside the buffer is therefore never more than 1, with a priority of 1 occurring only in the rare situation that the piece immediately outside the buffer is held by exactly one peer. Pieces that are further from being played back or held by more peers are given lower priorities, while a short distance from the buffer or a low availability increases the priority.

The availability of a particular piece and its distance from the last piece in the buffer are here equally weighted when determining the priority. However, we do not claim that giving equal weights to distance and availability is in any way the optimal solution. We have not extensively tested different weights, but it seems likely that factors such as the total number of pieces, the number of peers, and the network speed of the peers, will have an effect on how the distance and availability should be weighted for optimal performance. As it would not be possible to test all different combinations of weights under all circumstances, we choose here to weigh them equally for the sake of simplicity.

It should be noted that when we talk about availability of a piece we do not mean how many peers in total are involved in the torrent and hold that particular piece. What we actually look at is how many peers in one peer's active peer set hold that particular piece. As one peer need not necessarily be connected to all other peers, especially if the total number of peers is very large, we must by necessity look at the system from one peer's point of view. Another point worth mentioning is that the above priority calculation holds even if we allow playback to start somewhere else than from the beginning. Not yet requested pieces earlier in sequence than the last piece of our playback buffer will get negative priorities, and therefore will not be requested until all pieces higher in sequence have been requested.

V. COMPARISON WITH OTHER PIECE SELECTION METHODS

We have done two separate sets of simulations. The first set, the results of which also appeared in [1], is less exhaustive and compares our DAW piece selection method to two others:

- **Sequential Method**. This represents how a straightforward streaming would work, by always requesting pieces in the same order as they appear in the torrent.

- **Rarest-First Method with Buffer (RFB)**. This is the original BitTorrent piece selection method, slightly modified to better support streaming media. This is done as follows: we add a small buffer of fixed size, so that *k* pieces after the currently playing one are requested with the highest priority. If all *k* buffer pieces have been requested, we use the rarest-first method on the remaining part of the media. Another small modification is that if more than one piece have the same availability the original rarest-first method chooses between them randomly [12], while we always choose the one closest to being played back, as in the rarest-first method mentioned in [6].

In our second set of simulations, we add another piece selection method to the comparison:

- **BitTorrent Streaming (BiToS)**. This piece selection method is described in [6]. Pieces not yet downloaded are divided between a small high-priority set, with pieces close to being played back, and a larger remaining pieces set, with lower priority. The probability of choosing a piece from the high-priority set is *p* and the probability of choosing a piece from the remaining pieces set is similarly 1-*p*. Within each set, pieces are chosen rarest-first, with the aforementioned modification that if there is more than one piece with equal availability, the one closest to being played back is chosen. In [6], *p* was chosen to be 0.8 and we have used the same number here.

A. Our First Set of Simulations

In these simulations, we have focused on the piece selection algorithms, and the results therefore do not reflect real-world performance of applications. Our main focus has been on two things: the percent of requests going to the original source over time, and the availability of the last piece (which is also the rarest piece, in these cases) over logical time. The first one of these we want to be as low as possible, because a good peer-to-peer system should distribute the load equally over as many peers as possible and therefore not have a proportionally high amount of requests directed to the original source. The second one we want to be high, as we want the availability of the rarest piece to increase, as that signifies redundancy and robustness of the system. In these simulations we also assume that peers have the ability to send data to as many other peers as necessary, effectively creating a situation where a peer will always get the piece it requests.

The number of pieces was chosen to be 1000; large enough to study the behaviour of different piece selection methods over long periods of time. The buffer size for both DAW and RFB was set to 8 pieces; large enough for smooth playback but small enough that filling the buffer should not impact n overall performance. All simulations were done up to 800 logical time units; at the rate of one request per time unit we therefore never reach the situation that any peers have finished downloading, instead focusing on what happens from the start onwards.

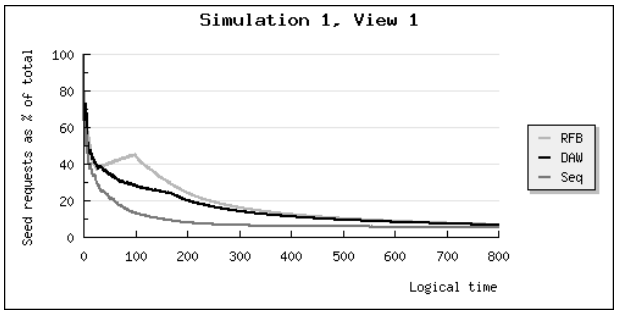


Figure 1. Seed requests as percentage of total requests over logical time, with peers joining at regular intervals.

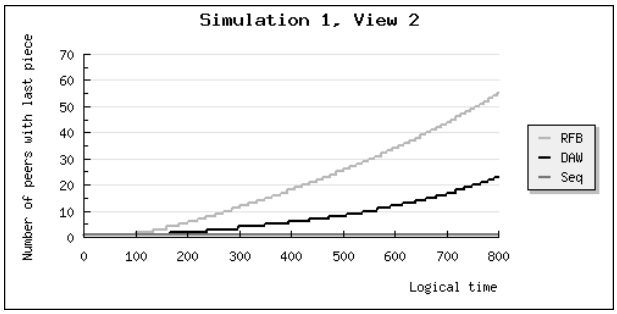


Figure 2. Availability of last piece over logical time, with peers joining at regular intervals.

In the first simulation, to stress the system we chose to have one seed and 100 regular peers, the latter arriving at regular intervals (one new peer every two logical time units). Fig. 1 shows that DAW here results in a lesser burden for the seed than RFB, although it is not as good as the sequential method. However, as Fig. 2 shows, the DAW does not increase the availability of the last piece as much as RFB. With RFB many rare pieces are requested early. These pieces are only available from the seed, and as seen in Fig. 1 the burden on the seed drops only after 100 logical time units. This corresponds to the point where there is no piece left where the seed is the only source, as can be seen in Fig. 2. The same drop can be seen, less dramatically, for DAW around logical time 170, with the same explanation.

The second simulation is identical to the first one, except that all regular peers join simultaneously. As can be seen in Fig. 3, the sequential method does not work in this theoretical situation, while DAW is the better suited one of the other two. Fig. 4 shows the situation as very similar to the one in Fig. 2, except that with more peers from the start it takes less time to increase the availability of the last (rarest) piece with DAW and RFB.

For our third simulation, we chose a similar setup to the first one, but with ten seeds instead of one. Fig. 5 shows the average percentage of requests over logical time to each one of the ten seeds, and DAW is again a less taxing choice than RFB. A possible reason for this can be seen in Fig. 6;

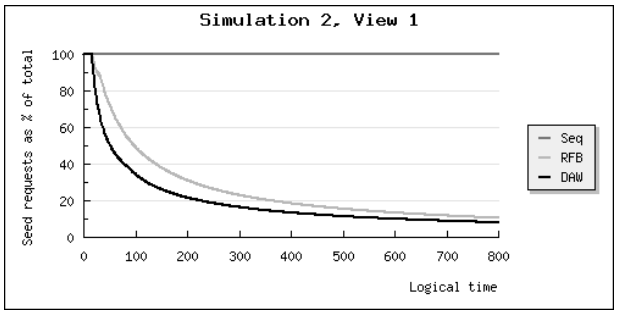


Figure 3. Seed requests as percentage of total requests over logical time, with peers joining simultaneously.

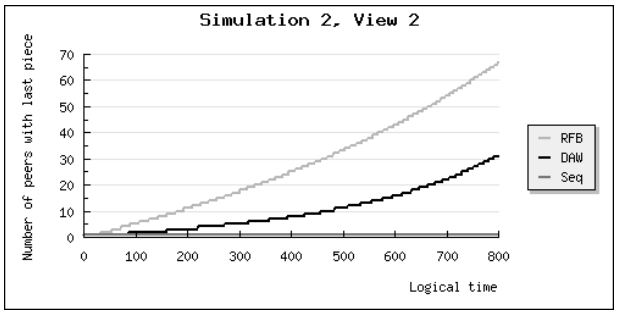


Figure 4. Availability of last piece over logical time, with peers joining simultaneously.

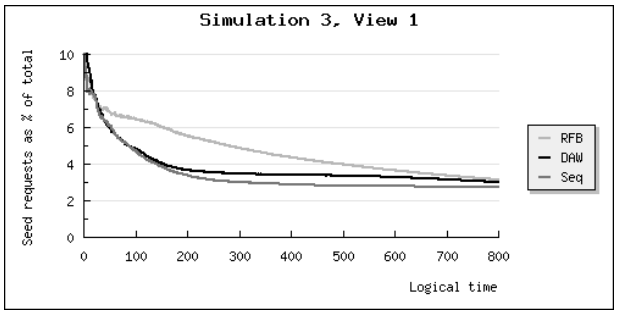


Figure 5. Average requests to a seed as percentage of total requests over logical time, with peers joining at regular intervals.

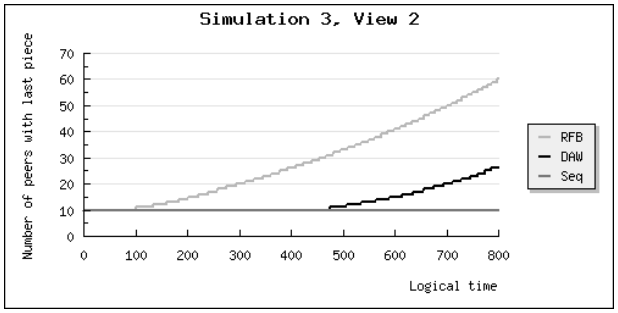


Figure 6. Availability of last piece over logical time, with peers joining at regular intervals.

compared to DAW, RFB spends a lot of time requesting rare pieces although there is no immediate reason for doing so.

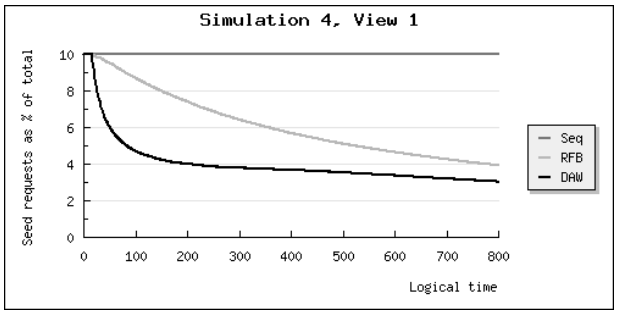


Figure 7. Average requests to a seed as percentage of total requests over logical time, with peers joining simultaneously.

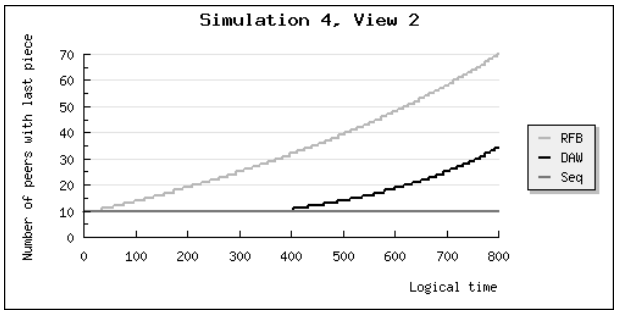


Figure 8. Availability of last piece over logical time, with peers joining simultaneously.

Our fourth simulation is a combination of the second and third ones, with ten seeds and all the regular peers starting at the same time. In Fig. 7 we see that the DAW piece selection method is again less demanding on the seeds than the RFB method, with the sequential method making on average a constant ten percent of the requests to each seed in this theoretical case. Fig. 8 is very much like Fig. 6, showing that the distance-availability weighted method does not spend a lot of time requesting the rarest piece until fairly late.

B. Our Second Set of Simulations

Compared to our first set of simulations, our second set is a step or two closer to reality. Unlike our first set of simulations, where a peer would always get the piece it requested, we have here introduced limits to how many other peers the seeds and regular peers can send to. This introduces the possibility that a peer will not have the piece that it is supposed to be playing back. In our simulations we have dealt with that situation in three different ways:

- **Skip**. We ignore the piece we should be playing back and just note that that piece has been skipped. This is the method favoured by BiToS [6], and it should be noted that for this to work in practise the format of the media content must be tolerant of missing data.
- **Stop**. If we are missing the piece that should be played back, we stop playback until we have been able to receive all pieces in our buffer (or high-priority set), after which we resume playback. From an end user point of view this is similar to how many current on-demand streaming media systems work,

in that not receiving data fast enough means that playback is paused until an amount of consecutive data has been received.

- **Skip and stop**. If we are missing the piece we are supposed to play back we skip it, but if our buffer (or high-priority set) is completely empty we stop playback until we have received all pieces in it. This should in theory generate less complete stops than by always stopping, and possibly also less skips than by just skipping.

What we actually measure in these simulations is three things: firstly, the playback position of the peers after a certain time; secondly, the number of skips and/or stops encountered before that time; and thirdly, the percentage of pieces that should have been transferred that were actually skipped and/or the number of stops per 100 pieces played back, respectively. What we are looking for is thereby a high number for the playback position, but low numbers for the amount of skips and stops.

In all these simulations, we have 10 seeds and 90 regular peers. The seeds can upload to 8 peers simultaneously and the regular peers to 2 peers simultaneously, at the rate of half the playback speed for each peer. The regular peers request new pieces twice as fast as the playback speed. In all the following figures we look at the situation after a logical time of 800. In simulations 5, 6 and 7 the regular peers join simultaneously, while in simulations 8, 9 and 10 they join at regular intervals; similarly to simulations 1 and 3. All simulations were run multiple times and the maximum reported here is the maximum for any peer during any run, the minimum reported is the minimum for any one peer duing any run, and the average reported is the average for all peers over all runs.

Our simulation number five concerns the situation where any pieces not transferred are skipped completely. Fig. 9 shows the maximum, average and minimum playback positions of peers using the BiToS, DAW, RFB and sequential piece selection methods. BiToS seems worse than the others in this case, but it must be pointed out that in BiToS we always spend about 20% of our time downloading pieces not close to being played back, and therefore it is reasonable to expect it to take longer for playback to start. In a best-case scenario this would lead to a lower number of pieces skipped later on, but as we see in Fig. 10, there is not a significant difference in the absolute number of skipped pieces, and if we look at the relative numbers, i.e., the percentage of pieces that should have been played back that were skipped, as in Fig. 11, we find that arguably BiToS is worse than the other three, although the differences are small.

Our simulation number six concerns the situation where the piece to be played back not being available leads to a complete stop of the playback. Fig. 12 shows that the distance-availability weighted method here leads to the furthest playback position. In Fig. 13 we see that despite the good results for the playback position, DAW also does pretty well in the absolute number of stops by coming in second. Fig. 14 shows the relative number of stops, i.e., the number of stops per 100 pieces played back, and there is not a big difference between the four piece selection methods.

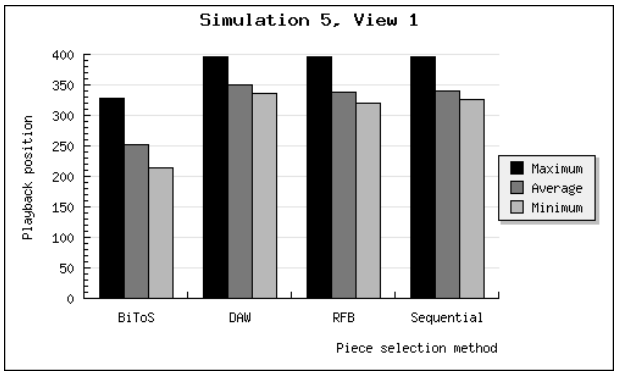


Figure 9. Playback positions of peers when playback stops for missing pieces, with peers joining simultaneously.

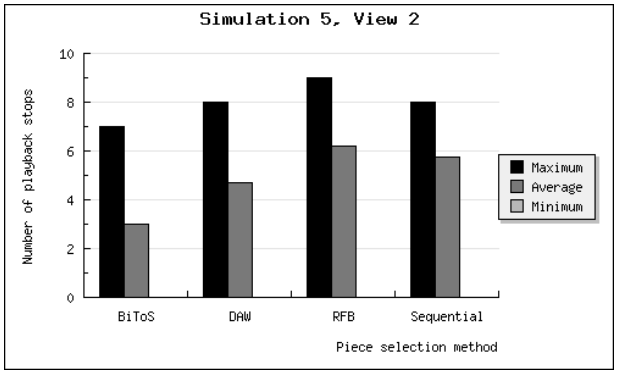


Figure 10. Number of playback stops for each peer when playback stops for missing pieces, with peers joining simultaneously.

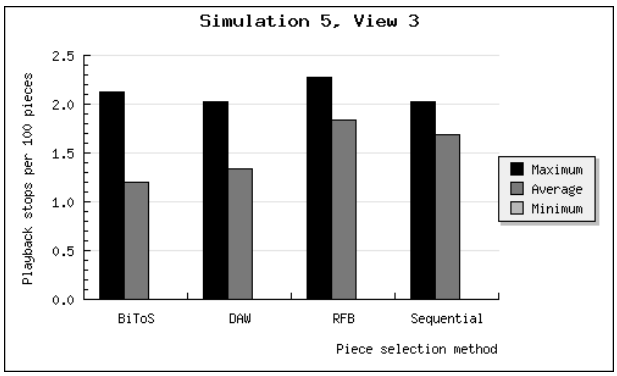


Figure 11. Playback stops per 100 pieces when playback stops for missing pieces, with peers joining simultaneously.

Simulation number seven seems to be the one with the most diverse results so far. Here we have the situation that if the piece to be played back is not received, we skip it, but if we do not have any of the pieces in our buffer or high-priority set, we stop. Fig. 15 shows the playback positions of the peers, and the situation is not very different from in the two preceding simulations, with DAW slightly ahead of the others and BiToS slightly behind. However, in Fig. 16 we notice that BiToS seems to skip a lot more than the others, and in Fig. 17 we notice that BiToS stops a lot less often.

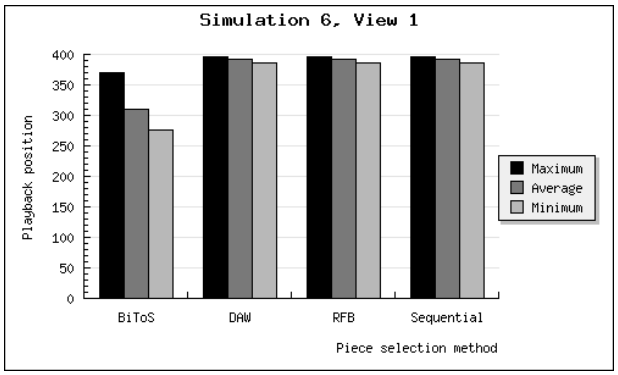


Figure 12. Playback positions of peers when playback skips missing pieces, with peers joining simultaneously.

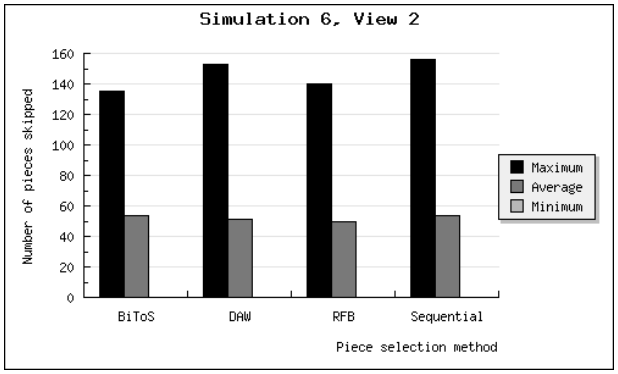


Figure 13. Number of pieces skipped for each peer when playback skips missing pieces, with peers joining simultaneously.

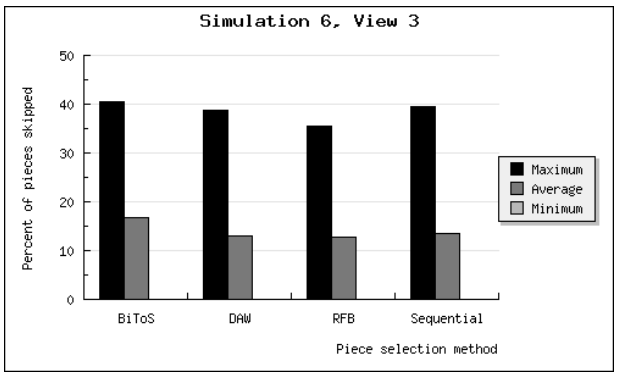


Figure 14. Percent of pieces skipped for each peer when playback skips missing pieces, with peers joining simultaneously.

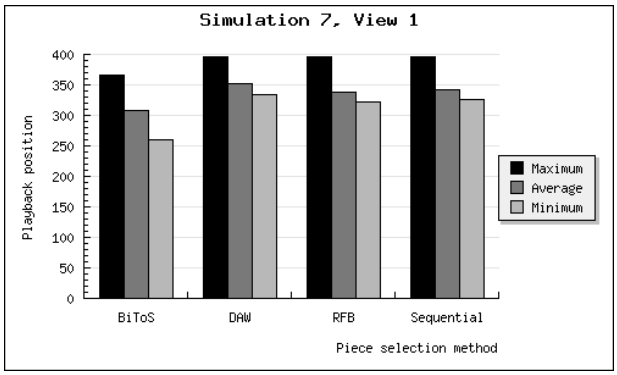


Figure 15. Playback positions of peers when playback skips and stops, with peers joining simultaneously.

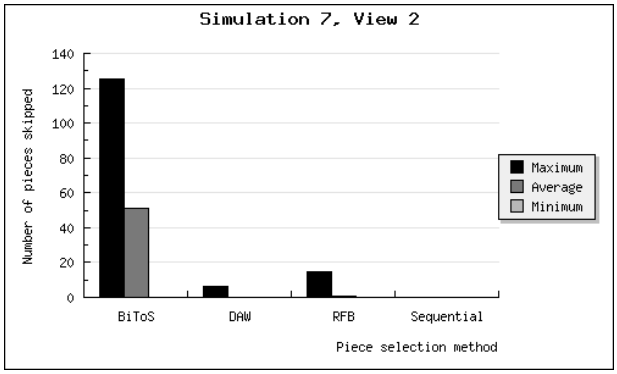


Figure 16. Number of pieces skipped for each peer when playback skips and stops, with peers joining simultaneously.

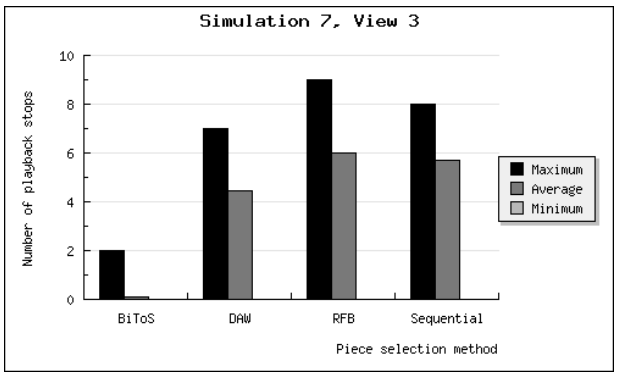


Figure 17. Number of playback stops for each peer when playback skips and stops, with peers joining simultaneously.

This can be explained by how these piece selection methods work. BiToS always requests pieces in a rarest-first, out-of-order fashion, and therefore it is likely that even if the piece to be played back is missing, the high-priority set is not empty, and therefore we just skip. DAW and RFB as described in this paper both have a small buffer in which pieces are requested in-order, and therefore it is likely that if the piece to be played back is missing, the whole buffer is empty, and therefore we stop. In the case of the sequential method, we always request sequentially, and therefore if the piece to be played back is missing, the following k pieces are also missing and we stop.

The following three simulations are similar to the previous ones, except that the regular peers do not join simultaneously. Figures 18, 19 and 20 show the result of simulation 8, where playback stops if the piece to be played back is not available, and the regular peers join at intervals. Because peers do not join at once, the difference between the maximum playback position and the minimum playback position is larger than in simulation 5 (compare Figures 9 and 18). Fig. 19 shows the absolute number of playback stops and Fig. 20 the number of playback stops per 100 pieces, and we see that the DAW and sequential piece selection methods are almost equal in this case, with both

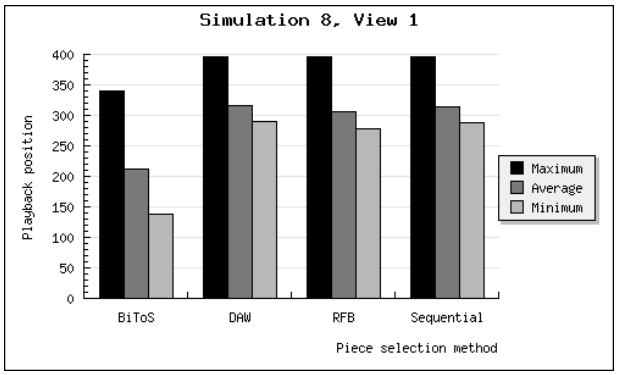


Figure 18. Playback positions of peers when playback stops for missing pieces, with peers joining at regular intervals.

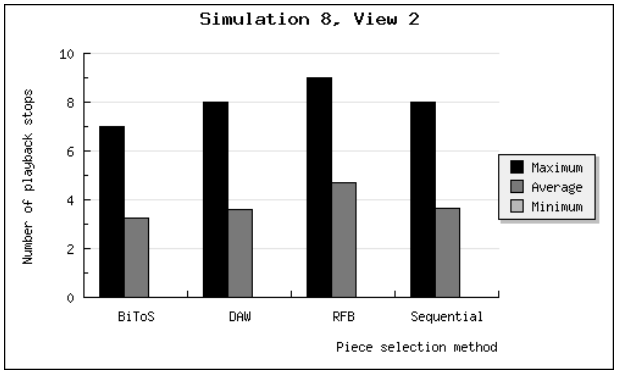


Figure 19. Number of playback stops for each peer when playback stops for missing pieces, with peers joining at regular intervals.

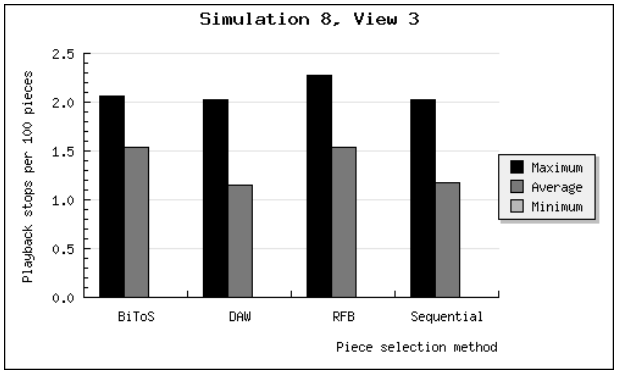


Figure 20. Playback stops per 100 pieces when playback stops for missing pieces, with peers joining at regular intervals.

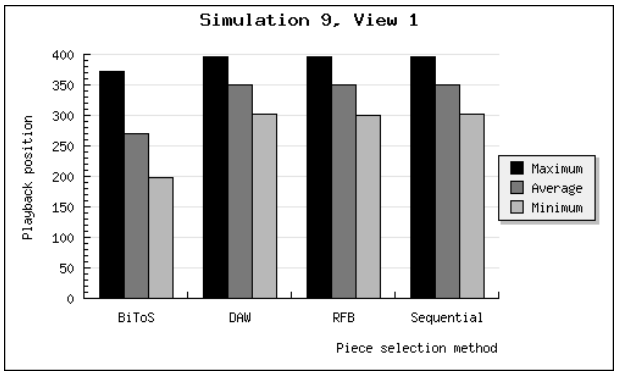


Figure 21. Playback positions of peers when playback skips missing pieces, with peers joining at regular intervals.

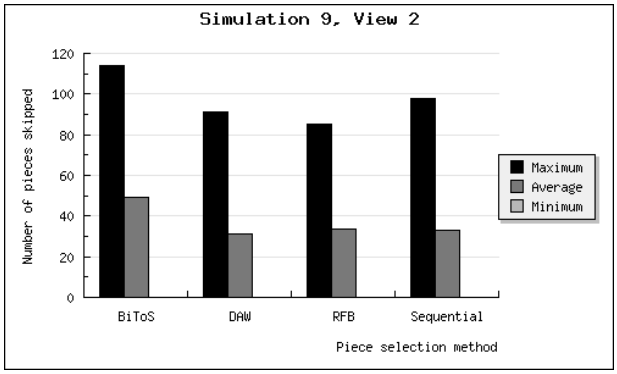


Figure 22. Number of pieces skipped for each peer when playback skips missing pieces, with peers joining at regular intervals.

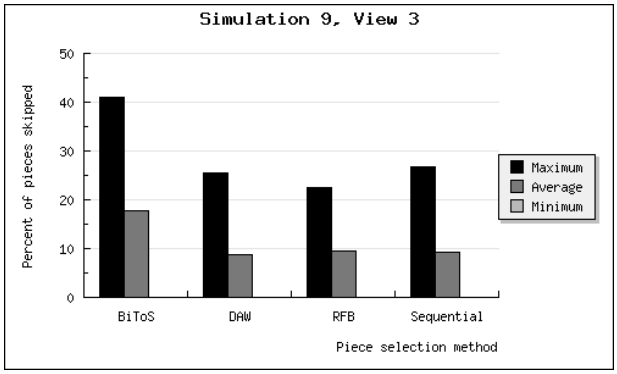


Figure 23. Percent of pieces skipped for each peer when playback skips missing pieces, with peers joining at regular intervals.

being better than the RFB method and in the relative case also better on average than BiToS.

Fig. 21 is comparable to Fig. 12, and again the larger difference between the maximum and minimum playback positions is caused by peers joining at intervals, instead of simultaneously. Comparing Fig. 22 and Fig. 13 we notice that not having all peers joining at once improves the result for the piece selection method utilising some form of sequential requests, leaving BiToS behind. This is emphasised in Fig. 23 where we see the percent of skipped pieces instead of the absolute amount.

Simulation 10 is comparable to simulation 7, and comparing Fig. 24 with Fig. 15 shows that also in this case the peers joining at intervals has the effect of putting BiToS further behind when it comes to playback position. Fig. 25 and Fig. 26 show that as in simulation 7, the nature of BiToS makes it skip more often than stop, while the sequential buffer used in the DAW and RFB piece selection methods make them behave more like the sequential method.

So far, we have only compared the piece selection methods to each other but not discussed the actual figures. While this is of course a very theoretical simulation, the bandwidth figures we have used suggest that playback

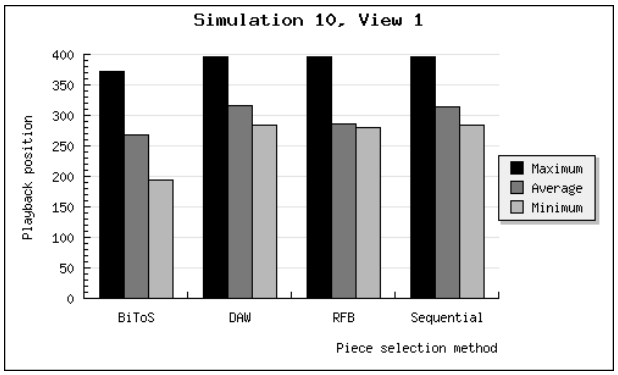


Figure 24. Playback positions of peers when playback skips and stops, with peers joining at regular intervals.

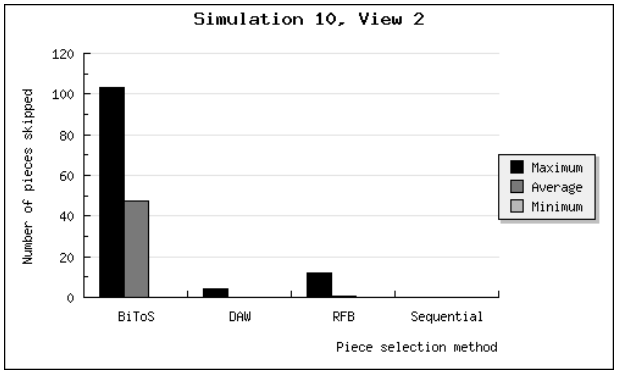


Figure 25. Number of pieces skipped for each peer when playback skips and stops, with peers joining at regular intervals.

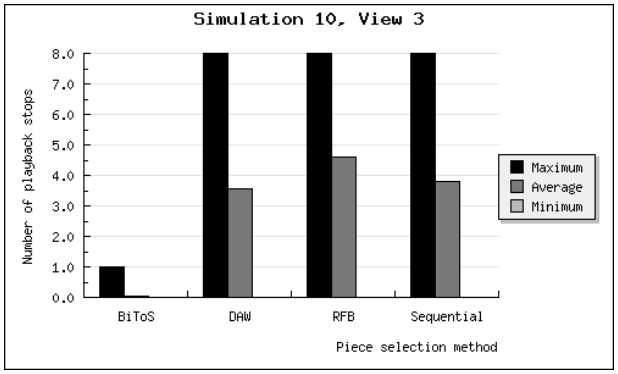


Figure 26. Number of playback stops for each peer when playback skips and stops, with peers joining at regular intervals.

should theoretically be possible for all peers without interruption. Instead, we get figures up to above 40% of all pieces skipped (for instance in Fig. 10) with average levels of more than 1/8 of all pieces skipped (for instance in Fig. 10 and 16). The situation for stops is not very good either, with an average of at least one stop for every 100 pieces played back (Fig. 14). Neither of these results is very good from an end-user point of view. The conclusion we can draw from this is that we need a better way of determining when to start, or restart, playback, as the methods we have used here do not seem to work in that regard. As for comparing the performance of DAW to the others, we note that in there three simulations DAW always comes up as the method that gives the best results with regard to playback position, without ever being the worst in any other regard.

One major concern regarding piece selection methods is whether they work well with the peer selection methods, in other words, whether the method of selecting pieces to request is compatible in practise with the methods used to determine which peers to send data to. The original tit-for-tat method from BitTorrent requires that data is exchanged both ways between peers in order to function well, and therefore is not a good solution when combined with the sequential method, where data is received from peers with more pieces and sent to peers with fewer pieces, exclusively. RFB would obviously work rather well in this fashion also, and the division of pieces into a high-priority set and a remaining pieces set as in BiToS is because the acquisition of pieces from the latter is "beneficial due to the Tit-for-Tat policy" [6]. We have not done any testing of the distance-availability weighted method on this subject, but our estimate is that it would be compatible. When the ratio of seeds to regular peers is low, the availability of pieces is important and the distance-availability weighted method selects pieces in a manner reminiscent of the rarest-first method, where the tit-for-tat policy has been proved as working. Conversely, when the ratio of seeds to regular peers is high, the distance-availability weighted method behaves similarly to the sequential method, but as the seeds do not require data to be sent to them, the need for out-of-order transfers in order to get the tit-for-tat policy working diminishes. We therefore believe our method would work in such a case as well.

VI. Related Work

This paper is an extended version of [1], which is a further development of a concept introduced in [14]. Whether due to its popularity or some other factor, such as its lack proprietarity, BitTorrent has been the subject of several other research projects during the last few years; some of which are directly relevant to ours.

The BitTorrent protocol has been analysed in real-world usage and found to be an efficient and viable solution for file sharing [8][15]. When it comes to the idea of on-demand streaming with BitTorrent as a basis, one of the more interesting propositions is the previously mentioned BiToS [6]. The main difference between BiToS and our distance-availability weighted method is that BiToS seems designed to maximise the amount of received pieces in a situation where the media bit rate is the same as the download rate of the client, while our solution is designed for a situation where the download bit rate is higher than the media bit rate, and all pieces should be received is such a way that playback is possible before all of the file has been transferred.

The approach used in BiToS is further expanded in [16], which also adds modifications to how the peers choose which peers to send data to, effectively replacing the tit-for-tat peer selection method used in BitTorrent with a method called Give-to-Get. The application Tribler uses Give-to-Get [17], while its protocol also contains other additions such as social networking. Another project combining social networking with BitTorrent and media streaming is

OneSwarm [10], adding "friend-to-friend" file sharing and, as of version 0.6 and later, the ability to choose between "streaming" media files (downloading sequentially) or not (downloading using rarest-first) [18]. As mentioned in Section I, there is also a project underway to "turn BitTorrent into a point-click-watch experience much more similar to YouTube", based on the popular, closed source BitTorrent application µTorrent [7].

Another BitTorrent-based service is LiveBT [19], which replaces the rarest-first method with Most-wanted-Block-Download-First (MBDF). In MBDF, a peer has a set of most wanted blocks (pieces), which is a fixed number of undownloaded pieces. Each peer also knows the most wanted blocks of its peers. With a probability *p* the peer's own most wanted piece is selected, and with probability 1-*p* the most wanted piece of its peers.

The idea of using a weighted priority is not unique to the DAW method. Wu et al [20] propose a weighted piece selection method, but for downloading instead of streaming. Their idea is that peers which hold many pieces are given greater weight than peers with few pieces when computing priorities. Whether this could improve performance also in streaming remains to be seen.

Besides BitTorrent-based solutions, there are also other projects underway to use peer-to-peer networks for on-demand streaming. Peer-to-peer networks serve as the backbone of both Spotify [21] and Voddler [22], the former a platform for on-demand streaming music, while the latter enables on-demand movies. Both these services distribute commercial content and therefore have limitations on usage as well as do not provide technical details on how their networks function.

VII. Conclusion and Future Work

In our first set of simulations, the distance-availability weighted piece selection method seems to be a better solution for on-demand streaming than either a straightforward sequential method or a modified version of the rarest-first method. Our second set of simulations, where in addition to the previously mentioned piece selection methods we also include the one presented in [6], do not show results contradicting the first set of simulations. However, the differences seem to be smaller than we previously thought, and none of the piece selection methods simulated seemed to be sufficiently good to work perfectly in the conditions given. However, as we have still only simulated the theoretical performance of the piece selection methods, we cannot comment on how they would work in a real-life environment. Future work on the subject could include practical implementation and real-world testing of the distance-availability weighted piece selection method, as well as comparison to other piece selection methods for streaming than the ones used for comparison here.

Acknowledgement

We thank Professor Kaisa Sere who has been very helpful during this project.

References

[1] P. Sandvik and M. Neovius, "The Distance-Availability Weighted Piece Selection Method for BitTorrent: A BitTorrent Piece Selection Method for On-Demand Streaming", In Proceedings of AP2PS '09, Sliema, Malta, October 2009

[2] B. Cohen, "BitTorrent - a new P2P app", Yahoo eGroups, http://finance.groups.yahoo.com/group/decentralization/message/3160 (Accessed August 2010)

[3] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy and M. Faloutsos, "File-sharing in the Internet: A characterization of P2P traffic in the backbone", Technical report, November 2003.

[4] Ipoque Internet Study 2008 / 2009. Available from http://www.ipoque.com/resources/internet-studies/internet-study-2008_2009 (Accessed August 2010)

[5] B. Cohen, "Incentives Build Robustness in BitTorrent", In Proc. of IPTPS, 2003.

[6] A. Vlavianos, M. Iliofotou and M. Faloutsos, "BiToS: Enhancing BitTorrent for Supporting Streaming Applications", 9th IEEE Global Internet Symposium 2006 (in Conjunction with IEEE INFOCOM 2006).

[7] µTorrent Labs: Project Falcon, http://www.utorrent.com/labs/falcon (Accessed August 2010)

[8] A. Legout, G. Urvoy-Keller and P. Michiard, "Rarest First and Choke Algorithms Are Enough", In Proceedings of ACM SIGCOMM/USENIX IMC'2006, Rio de Janeiro, Brazil, October 2006.

[9] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy and A. Venkataramani, "Do incentives build robustness in BitTorrent?", 4th USENIX Symposium on Networked Systems Design & Implementation (NSDI 2007).

[10] T. Isdal, M. Piatek, A. Krishnamurthy and T. Anderson, "Friend-to-friend data sharing with OneSwarm", Technical report, UW-CSE, February 2009.

[11] D. Choffnes and D. Bustamante, "Taming the Torrent: A practical approach to reducing cross-ISP traffic in P2P systems", In Proceedings of ACM SIGCOMM 2008, August 2008.

[12] A. Nordberg, "Introduction to BitTorrent", Umeå University, 2006. Available from http://www.rasterbar.com/products/libtorrent/bittorrent.pdf (Accessed August 2010)

[13] B. Cohen, "The BitTorrent Protocol Specification", January 2008, http://www.bittorrent.org/beps/bep_0003.html (Accessed August 2010)

[14] P. Sandvik, "Adapting Peer-to-Peer File Sharing for On-Demand Media Streaming", Master of Science Thesis, Åbo Akademi University, May 2008.

[15] A. Legout, G. Urvoy-Keller and P. Michiardi, "Understanding BitTorrent: An Experimental Perspective", Technical Report (inria-00000156, version 2 - 19 July 2005), INRIA, Sophia Antipolis, July 2005.

[16] J.J.D. Mol, J.A. Pouwelse, M. Meulpolder, D.H.J. Epema and H.J. Sips, "Give-to-Get: Free-riding-resilient Video-on-Demand in P2P Systems", Proc. of SPIE, Multimedia Computing and Networking Conference (MMCN), vol. 6818, article 681804, 2008.

[17] A. Bakker et al, "Tribler Protocol Specification v0.0.2", January 2009, Available from http://www.tribler.org (Accessed August 2010)

[18] OneSwarm Changelog, http://wiki.oneswarm.org/index.php/Changelog (Accessed August 2010)

[19] J. Lv, X. Cheng, Q. Jiang, J. Ye, T. Zhang, S. Lin and L. Wang, "LiveBT: Providing Video-on-demand Streaming Service over BitTorrent Systems", pdcat, pp.501-508, Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2007), 2007

[20] C. Wu, C. Li and J. Ho, "Improving the Download Time of BitTorrent-like Systems", IEEE International Conference on Communications 2007 (ICC 2007), Glasgow, Scotland, June 2007

[21] Spotify, http://www.spotify.com/fi/help/faq/tech/ (Accessed August 2010)

[22] Voddler, http://www.voddler.com/help/topic/2721821230584819787 (Accessed August 2010)

www.iariajournals.org

International Journal On Advances in Intelligent Systems
ICAS, ACHI, ICCGI, UBICOMM, ADVCOMP, CENTRIC, GEOProcessing, SEMAPRO, BIOSYSCOM, BIOINFO, BIOTECHNO, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS
issn: 1942-2679

International Journal On Advances in Internet Technology
ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING
issn: 1942-2652

International Journal On Advances in Life Sciences
eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO
issn: 1942-2660

International Journal On Advances in Networks and Services
ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION
issn: 1942-2644

International Journal On Advances in Security
ICQNM, SECURWARE, MESH, DEPEND, INTERNET, CYBERLAWS
issn: 1942-2636

International Journal On Advances in Software
ICSEA, ICCGI, ADVCOMP, GEOProcessing, DBKDA, INTENSIVE, VALID, SIMUL, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS
issn: 1942-2628

International Journal On Advances in Systems and Measurements
ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL
issn: 1942-261x

International Journal On Advances in Telecommunications
AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA
issn: 1942-2601