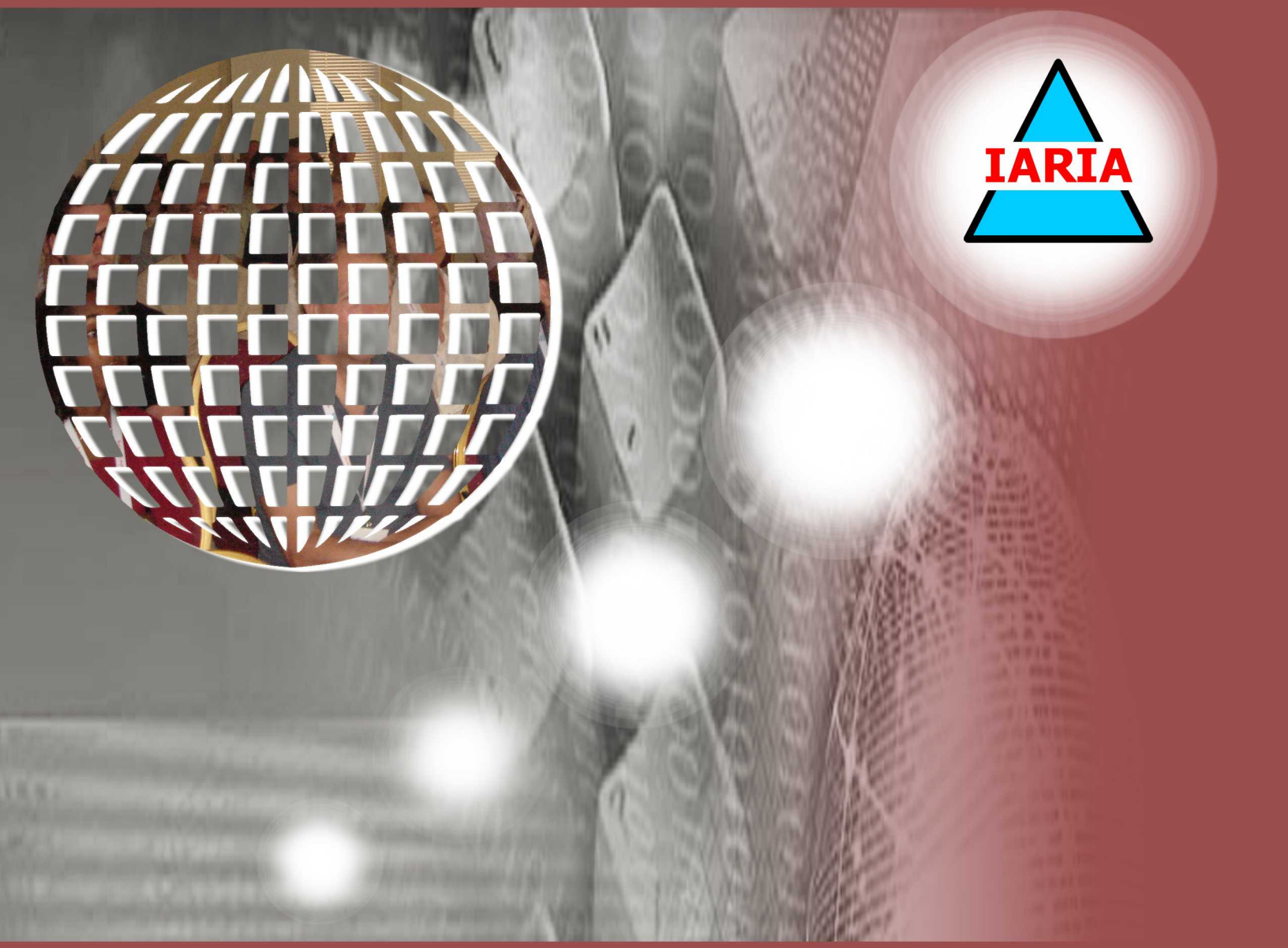
International Journal on Advances in Telecommunications



2008 vol. 1 nr. 1

The *International Journal On Advances in Telecommunications* is Published by IARIA.
ISSN: 1942-2601
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal On Advances in Telecommunications
Volume 1, Number 1, 2008

Editorial Board

First Issue Coordinators

Jaime Lloret, Universidad Politécnica de Valencia, Spain
Pascal Lorenz, Université de Haute Alsace, France
Petre Dini, Cisco Systems, Inc., USA / Concordia University, Canada

Advanced Telecommunications

- Tulin Atmaca, IT/Telecom&Management SudParis, France
- Rui L.A. Aguiar, Universidade de Aveiro, Portugal
- Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania
- Symeon Chatzinotas, University of Surrey, UK
- Denis Collange, Orange-ftgroup, France
- Todor Cooklev, Indiana-Purdue University - Fort Wayne, USA
- Jose Neuman De Souza, Federal University of Ceara, Brazil
- Sorin Georgescu, Ericsson Research, Canada
- Paul J. Geraci, Technology Survey Group, USA
- Christos Grecos, University of Central Lancashir-Preston, UK
- Manish Jain, Microsoft Research – Redmond
- Michael D. Logothetis, University of Patras, Greece
- Natarajan Meghanathan, Jackson State University, USA
- Masaya Okada, ATR Knowledge Science Laboratories - Kyoto, Japan
- Jacques Palicot, SUPELEC- Rennes, France
- Maciej Piechowiak, Kazimierz Wielki University - Bydgoszcz, Poland
- Dusan Radovic, TES Electronic Solutions - Stuttgart, Germany
- Matthew Roughan, University of Adelaide, Australia
- Sergei Semenov, Nokia Corporation, Finland
- Carlos Becker Westphal, Federal University of Santa Catarina, Brazil
- Rong Zhao, Detecon International GmbH - Bonn, Germany
- Piotr Zwierzykowski, Poznan University of Technology, Poland

Digital Telecommunications

- Bilal Al Momani, Cisco Systems, Ireland
- Tulin Atmaca, IT/Telecom&Management SudParis, France
- Claus Bauer, Dolby Systems, USA
- Claude Chaudet, ENST, France
- Gerard Damm, Alcatel-Lucent, France

- Michael Grottke, Universitat Erlangen-Nurnberg, Germany
- Yuri Ivanov, Movidia Ltd. – Dublin, Ireland
- Ousmane Kone, UPPA - University of Bordeaux, France
- Wen-hsing Lai, National Kaohsiung First University of Science and Technology, Taiwan
- Pascal Lorenz, University of Haute Alsace, France
- Jan Lucenius, Helsinki University of Technology, Finland
- Dario Maggiorini, University of Milano, Italy
- Pubudu Pathirana, Deakin University, Australia
- Mei-Ling Shyu, University of Miami, USA

Communication Theory, QoS and Reliability

- Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania
- Piotr Cholda, AGH University of Science and Technology - Krakow, Poland
- Michel Diaz, LAAS, France
- Ivan Gojmerac, Telecommunications Research Center Vienna (FTW), Austria
- Patrick Gratz, University of Luxembourg, Luxembourg
- Axel Kupper, Ludwig Maximilians University Munich, Germany
- Michael Menth, University of Wuerzburg, Germany
- Gianluca Reali, University of Perugia, Italy
- Joel Rodriques, University of Beira Interior, Portugal
- Zary Segall, University of Maryland, USA

Wireless and Mobile Communications

- Tommi Aihkisalo, VTT Technical Research Center of Finland - Oulu, Finland
- Zhiquan Bai, Shandong University - Jinan , P. R. China
- David Boyle, University of Limerick, Ireland
- Xiang Gui, Massey University-Palmerston North, New Zealand
- David Lozano, Telefonica Investigacion y Desarrollo (R&D), Spain
- D. Manivannan (Mani), University of Kentucky - Lexington, USA
- 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
- Ossama Younis, Telcordia Technologies, USA

Systems and Network Communications

- Fernando Boronat, Integrated Management Coastal Research Institute, Spain
- Anne-Marie Bosneag, Ericsson Ireland Research Centre, Ireland
- Huaqun Guo, Institute for Infocomm Research, A*STAR, Singapore
- Jong-Hyouk Lee, Sungkyunkwan University, Korea
- Elizabeth I. Leonard, Naval Research Laboratory – Washington DC, USA
- Sjouke Mauw, University of Luxembourg, Luxembourg

- Reijo Savola, VTT, Finland

Multimedia

- Dumitru Dan Burdescu, University of Craiova, Romania
- Noel Crespi, Institut TELECOM SudParis-Evry, France
- Mislav Grgic, University of Zagreb, Croatia
- Atsushi Koike, KDDI R&D Labs, Japan
- Polychronis Koutsakis, McMaster University, Canada
- Chung-Sheng Li, IBM Thomas J. Watson Research Center, USA
- Artur R. Lugmayr, Tampere University of Technology, Finland
- Parag S. Mogre, Technische Universitat Darmstadt, Germany
- Chong Wah Ngo, University of Hong Kong, Hong Kong
- Justin Zhan, Carnegie Mellon University, USA
- Yu Zheng, Microsoft Research Asia - Beijing, China

Space Communications

- Emmanuel Chaput, IRIT-CNRS, France
- Alban Duverdier, CNES (French Space Agency) Paris, France
- Istvan Frigyes, Budapest University of Technology and Economics, Hungary
- Michael Hadjitheodosiou ITT AES & University of Maryland, USA
- Mark A Johnson, The Aerospace Corporation, USA
  Massimiliano Laddomada, Texas A&M University-Texarkana, USA
- Haibin Liu, Aerospace Engineering Consultation Center-Beijing, China
- Elena-Simona Lohan, Tampere University of Technology, Finland
- Gerard Parr, University of Ulster-Coleraine, UK
- Cathryn Peoples, University of Ulster-Coleraine, UK
- Michael Sauer, Corning Incorporated/Corning R&D division, USA

International Journal On Advances in Telecommunications
Volume 1, Number 1, 2008

Foreword

Finally, we did it! It was a long exercise to have this inaugural number of the journal featuring extended versions of selected papers from the IARIA conferences.

With this 2008, Vol. 1 No.1, we open a long series of hopefully interesting and useful articles on advanced topics covering both industrial tendencies and academic trends. The publication is by-invitation-only and implies a second round of reviews, following the first round of reviews during the paper selection for the conferences.

Starting with 2009, quarterly issues are scheduled, so the outstanding papers presented in IARIA conferences can be enhanced and presented to a large scientific community. Their content is freely distributed from the www.iariajournals.org and will be indefinitely hosted and accessible to everybody from anywhere, with no password, membership, or other restrictive access.

We are grateful to the members of the Editorial Board that will take full responsibility starting with the 2009, Vol 2, No1. We thank all volunteers that contributed to review and validate the contributions for the very first issue, while the Board was getting born. Starting with 2009 issues, the Editor-in Chief will take this editorial role and handle through the Editorial Board the process of publishing the best selected papers.

Some issues may cover specific areas across many IARIA conferences or dedicated to a particular conference. The target is to offer a chance that an extended version of outstanding papers to be published in the journal. Additional efforts are assumed from the authors, as invitation doesn't necessarily imply immediate acceptance.

This particular issue covers papers invited from those presented in 2007 and early 2008 conferences. The papers cover a quite heterogeneous spectrum. One topic is referring to multicast transmission cost scheme. A related one is treating traffic streams and traffic engineering for modeling systems and services in enterprise and carrier networks. The third topic is covering open platforms and J2ME development experiences for DVB-H.3G and HTTP over Bluetooth, respectively.

We hope in a successful launching and expect your contributions via our events.

First Issue Coordinators,
Jaime Lloret, Universidad Politécnica de Valencia, Spain
Pascal Lorenz, Université de Haute Alsace, France
Petre Dini, Cisco Systems, Inc., USA / Concordia University, Canada

International Journal On Advances in Telecommunications
Volume 1, Number 1, 2008

CONTENTS

Fair Allocation of Multicast Transmission Costs

Patrik Österberg and Tingting Zhang
Department of Information Technology and Media
Mid Sweden University
SE-851 70 Sundsvall, Sweden
patrik.osterberg@miun.se, tingting.zhang@miun.se

Abstract

In scenarios where many receivers simultaneously are interested in the same data, multicast transmission is more bandwidth efficient than unicast. The reason is that the receivers of a multicast session share the resources through a common transmission tree. Since the resources are shared between the receivers, it is reasonable that the costs corresponding to these resources should be shared as well.

This paper deals with fair cost sharing among multicast receivers, and the work is based upon the assumption that costs should be shared according to the resource usage. However, it is not for certain that an optimally fair cost allocation is most beneficial for the receivers; receivers that cannot cover their fair share of the costs may nevertheless be able to contribute to the cost sharing to some extent. We propose a cost-allocation mechanism that strives to allocate the costs fairly, but gives discount to poor receivers who at least manage to cover the additional cost of providing them with the service.

Keywords: *multicast, fairness, cost allocation*

1. Introduction

Video-streaming services are rapidly gaining in popularity, and the quality of these services is also increasing. Internet video already has attracted a large crowd, but the quality leaves more to wish for. *Internet protocol television* (IPTV) is being deployed on a wider extent and the transition to *high definition television* (HDTV) resolution is ongoing. In the longer run, 3D video and *free-viewpoint video* (FVV) services will also be offered. This development produces challenges for computer networks of all sizes, from small LANs to the whole Internet.

The employment of multicast transmission can reduce the resource demands of services where some content is simultaneously transmitted to a number of users. The reason is that the receivers of a multicast session share the resources through a common transmission tree, where data are only transmitted once along each branch. Nevertheless, multicast transmission is not deployed to its full extent.

In [13], we therefore aimed at creating an incentive for the use of multicast transmission. The proposal was a general definition of how the bandwidth should be distributed fairly between competing multicast and unicast sessions. In short, the definition takes the number of receivers into consideration, which is beneficial for multicast sessions.

If the transmission costs for multicast sessions also were favorable when compared to those of unicast, this would create another incentive for the employment of multicast. In this paper, we therefore study how the transmission costs of multicast sessions should be allocated to achieve this goal. This work is an extension of that presented at the IARIA ICDT 2007 conference [15] and in [14].

Henceforth, *costs* always refers to the costs associated with the actual transmission, i.e. costs for network resources such as links and routers, or in reality, the fees that the *Internet service providers* (ISPs) are charging. The cost of the delivered content is strictly excluded throughout this work.

To begin with, we adopt the fundamental assumption made by Herzog *et al.* in [8], that the cost of a multicast tree should be assigned to the receivers and not to the source. The reason is that multicast transmission is receiver initiated and that the service primarily is of use to the receivers, since the sources typically are streaming servers. The three basic requirements; no positive transfers, voluntary participation, and consumer sovereignty, are also sustained.

Further, we believe that fair cost allocation should be based on resource usage. This is likely to make the resource utilization more effective. With a flat-rate policy, there are no incentives for limiting the resource usage, as long as it is maintained within the postulated limit.

As an example, in everyday life, the expectation is that a train ticket will cost less than an air ticket. In addition, short domestic flights are expected to cost less than longer international flights. Furthermore, a shared cab is cheaper

per capita than a private one. The higher costs involved in more exclusive services together with a limited budget, probably accounts for the most common reason why people do not travel more, further, and faster, etc. A season ticket or the like, i.e. a flat rate policy, works against this incentive. Although, there might exist other motives, such as environmental awareness etc.

For data transmission over computer networks, the two major resource-related factors, which might differ between receivers, also relate to distance and quality. Namely the transmission path and the *quality of service* (QoS) requirements. As an example, choosing a server that is geographically close and settling for a low quality service would reduce the resource usage. This also holds for multicast receivers, but here the "shared-cab" aspect comes into play as well. Connecting to a multicast tree with many receivers in the vicinity will also save resources.

In Section 2 and 3 we describe existing cost-allocation mechanisms for multicast traffic. These mechanisms are then studied in Section 4, and the finding is that none take all of the aforementioned factors into consideration. A terminology for cost-allocation mechanisms that targets multi-rate multicast sessions is then introduced in Section 5, whereupon two new cost-allocation mechanisms are proposed in Section 6. The conclusions are presented in Section 7 together with some possible future research topics.

2. Existing cost-allocation mechanisms

In this section, a number of cost-allocation mechanisms for cost sharing among multicast receivers are outlined. These are a selection of existing mechanisms, other proposals for example include [5] and [3]. However, some of the terminology associated with cost sharing among multicast receivers is firstly introduced.

2.1. Terminology for multicast cost sharing

This section outlines the notations for cost sharing among multicast receivers, originally introduced in [8].

The number of receivers upstream and downstream respectively for a particular link are denoted by n_u and n_d. The receivers downstream of a link are those receivers whose transmission paths from the source traverse that link. The receivers upstream of a link are somewhat less intuitively defined as the receivers who are not located downstream of that link. In the multicast tree of Figure 1, where t is the transmitter, receivers r_1, r_2 and r_3 are located downstream of link l, whereas receivers r_4 through r_7 are upstream of link l. The part of the cost of the link allocated to the upstream receivers is described by the function $\mathrm{F}_u(n_u, n_d)$, whereas $\mathrm{F}_d(n_u, n_d)$ represents the part of the cost that is allocated to the downstream receivers.

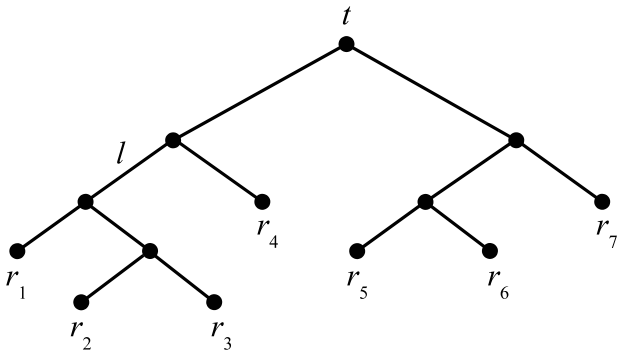


Figure 1. A multicast transmission tree with seven receivers.

Multicast sessions that support multiple *quality of service* (QoS) levels are also covered in [8]. The shares of the total cost allocated to the upstream and downstream receivers requesting QoS level i, are denoted by $\mathrm{F}_u^i(z_u, z_d)$ and $\mathrm{F}_d^i(z_u, z_d)$ respectively. However, the terms z_u and z_d are not defined.

2.2. The edge-pricing paradigm

Pricing and cost allocation in computer networks are treated extensively by Shenker *et al.* in [12]. They initiate their discussion with pricing based on estimated congestion conditions. The reason being the high complexity associated with the computation of the actual prevailing congestion conditions and the consequence is basically QoS-sensitive time-of-day pricing. They then claim that differentiated pricing based on estimated congestion conditions can be exchanged for differentially priced QoS classes. When the estimated congestion probability is low, even cheaper QoS classes will perform well. Users can therefore adapt their costs by monitoring and changing QoS classes.

Shenker *et al.* further propose that the pricing, aside from the QoS class, only should depend on the locations of the source and destination. The costs of the actual transmission path are approximated using the costs of the expected path. Consequently, the prices are based upon the estimated congestion conditions along the expected transmission path from the source to the destination. If information about congestion conditions is gathered at the edges of the network of an ISP, it should be possible to determine the price of a session at the access point. For connections that traverse the borders between different ISPs, the ISPs must purchase the service from each other in the same manner that regular users purchase service. This solution is called the *edge-pricing paradigm*.

Multicast traffic causes a challenge for the edge-pricing

paradigm, because a multicast destination address is merely a logical name and does not identify the individual receivers of the multicast group. The only information about multicast sessions that is present in a router node is regarding the next hop(s). It is therefore impossible to estimate the multicast tree at the access points. Shenker *et al.* propose control messages to be sent when new receivers join a multicast group. These messages should be forwarded along the reverse multicast tree to the access point of the source, where the cost of the tree may be approximated. The ISPs would process the control messages at the edges of their network and thereby extract adequate information. An alternative solution is to record the cost of each link within the control messages.

Shenker *et al.* also have a general discussion relating to cost sharing among multicast receivers. However, they do not propose any cost-allocation mechanism.

2.3. Single QoS cost allocation

In [8], Herzog *et al.* present an extensive work regarding how the costs of multicast trees should be split among the receivers. They present a number of cost-allocation mechanisms, of which the *equal tree split* (ETS) and *equal link split downstream* (ELSD) mechanisms are given the most attention.

The ELSD cost-allocation mechanism splits the cost of each link in the tree evenly between the downstream receivers. Using the notations introduced in subsection 2.1, the part of the cost of the link allocated to the upstream receivers can be described as

$$\mathrm{F}_u(n_u, n_d) = 0, \tag{1}$$

whereas the part of the cost allocated to each downstream receivers becomes

$$\mathrm{F}_d(n_u, n_d) = \frac{1}{n_d}. \tag{2}$$

The ETS cost-allocation mechanism splits the cost of the entire transmission tree uniformly amongst all the receivers. Using the same notations, we obtain

$$\mathrm{F}_u(n_u, n_d) = \mathrm{F}_d(n_u, n_d) = \frac{1}{n_u + n_d}. \tag{3}$$

2.4. QoS-based cost allocation

If the transmitted data are hierarchically encoded and marked and the router nodes employ priority dropping, users may choose to subscribe to a service although they cannot utilize the entire data rate transmitted by the source. The most obvious reason behind such limitations are network connections with low capacity. When the transmitted content is real-time video, another limiting factor might be the rendering capacity of the receiving device. In either case, these users do not utilize the entire bandwidth allocated to a multicast session, at least not on all of the links along their transmission path.

In [8], Herzog *et al.* observe that this should affect the cost allocation of multicast sessions, but they do not propose any specific cost-allocation mechanism for these scenarios. Using the terminology of subsection 2.1, they do however point out that if the cost-allocation functions fulfill the following condition,

$$\sum_{i=1}^{I} \left(z_u^i \cdot \mathrm{F}_u^i(z_u, z_d) + z_d^i \cdot \mathrm{F}_d^i(z_u, z_d) \right) = 1, \tag{4}$$

the costs associated with the link in question are fully allocated among the receivers.

Liu *et al.* study usage-based pricing and cost sharing of multicast traffic in [9]. They propose a cost-allocation mechanism, whose cost sharing they state "is proportional to individual members resource requirements, should a unicast service be used". The receivers are divided into categories depending on their requested QoS level. The costs associated with a particular category are then aggregated over the entire multicast tree, but only split among receivers obtaining that QoS level or higher, in an ETS fashion. Henceforth, this cost-allocation mechanism is therefore referred to as *QoS-dependent ETS* (QoS-D ETS).

3. Game-theoretic cost-allocation mechanisms

Many researchers have considered the bandwidth-allocation and pricing process from a game-theoretic perspective. Somewhat simplified, this implies that potential users place bids which reflect what the service is worth to them. The ISP then allocates the resources according to these bids. Some basic notions of game theory that are introduced in [11] are outlined in 3.1, followed by two game-theoretic cost-allocation mechanisms. Other works on the same subject are [4] and [2].

3.1. Game-theoretic notions

A cost-allocation mechanism in which the costs allocated to the users exactly match the cost of the service, is called *budget balanced*. A user's *welfare* can be described as the satisfaction after obtaining a service for a certain cost. An *efficient* cost-allocation mechanism chooses to serve the set of users that maximizes the aggregated welfare of all the users.

Assume that a user is part of a user set that is a subset of a larger set of users. Then a cost-allocation mechanism is *cross-monotonic* if for all such user sets, the cost allocated

to the user when the larger set is served, is lower or equal in comparison to when the smaller set is served.

It is reasonable to assume that users are selfish and place bids that maximize their probable welfare. A cost-sharing mechanism is *strategyproof* if users maximize their welfare by placing bids that truthfully correspond to how much the service is worth to them. *Group strategyproof* is a harder criterion that requires the cost-allocation mechanism to be resistant against groups of users who jointly place their bids in an attempt to increase their welfares.

Another contribution of [11], is the establishing of the following three basic requirements:

- *no positive transfers* – no user is paid to obtain a service
- *voluntary participation* – no user is forced to obtain a service
- *consumer sovereignty* – no user is refused a service if their bid is sufficiently high

According to [6], there are two cost-allocation mechanisms that are naturally strategyproof and adhere to these basic requirements, the *marginal-cost* (MC) and *Shapley-value* (SH) mechanisms. Further, it is stated that these are the two most appropriate mechanisms for cost sharing among multicast receivers.

3.2. The Shapley-value mechanism

The SH cost-allocation mechanism is the game-theoretical equivalent to ELSD. It splits the cost of a network link equally between all receivers that are located downstream [6]. The SH mechanism is group strategyproof and budget balanced. However, it is not efficient but has the smallest maximum loss of welfare among the budget-balanced mechanisms.

3.3. The marginal-cost mechanism

As described in [11], the MC mechanism essentially charges the marginal cost to the users, that is the cost of providing the service to all users minus the cost of providing the service to all but the user in question. It therefore has the characteristic that it treats equals equally, that is if two receivers give rise to the same marginal cost and place identical bids, they are allocated the same amount of resources and are charged the same cost. Further, the MC mechanism is efficient but not budget balanced nor group strategyproof.

In [1], the MC mechanism is applied to multicast sessions that support multiple rates. The *split session* and *layered* paradigms are studied, but only the layered paradigm is somewhat relevant here, since a split session basically implies separate transmissions of different QoS levels, i.e. the problem associated with multiple QoS levels is divided into a number of problems, each with a single QoS level.

The layered paradigm, thoroughly described in [10], utilizes hierarchically encoded data, which is divided into QoS layers that are transmitted to individual multicast groups. The receivers consequently join multicast groups with QoS layers that can be combined into the desired QoS level. The layered paradigm therefore inherently implies that costs are separated according to QoS requirements.

3.4. Comparison of SH and MC mechanisms

In [7], both the SH and MC cost-allocation mechanisms are implemented and experiments are carried out. The MC is shown to generate a smaller revenue, which is not surprising since it is not budget balanced. On the other hand, the MC mechanism is faster than the SH mechanism.

In [6], it is observed that the MC mechanism only requires two messages per link in the multicast tree, whereas the number of messages required for the SH mechanism is of the order of the square of the number of links.

4. Evaluation of existing mechanisms

In this section, the cost-allocation mechanisms outlined in Section 2 and 3 are evaluated based on their attractiveness to the receivers. Important parameters are the magnitude of the costs and how fairly the costs are distributed.

4.1. The edge-pricing paradigm

The edge-pricing paradigm [12], briefly described in subsection 2.2, possesses some attractive properties, and it appears to be based upon sound approximations. However, the authors do not specify the pricing policy to be used. This decision is left to the individual ISPs. There are two main classes of pricing policies; usage-based policies where users are charged based on their actual usage, and capacity-based or flat-rate policies, where the users pay for the desired capacity. The choice, in this case, was to focus on usage-based pricing policies, since they are more favorable to multicast sessions and also might be considered to be fairer.

4.2. Single QoS cost allocation

For usage-based pricing policies, the cost of a multicast session should be divided among the receivers. The receivers in a multicast group have unique transmission paths per definition, otherwise they would have been positioned at the same location. As outlined in subsection 2.3, Herzog *et al.* propose a couple of cost-allocation mechanisms that are based upon the individual receivers' transmission paths [8]. However, there is a second factor that might affect

the amount of resources that are utilized by the individual receivers, namely the QoS requirements.

4.3. QoS-based cost allocation

As stated in subsection 2.4, users may choose to subscribe to a service although they cannot utilize the entire data rate transmitted by the source. These users do not use the entire bandwidth allocated to a multicast session, and should therefore, from a usage-based pricing perspective, be allocated a smaller share of the costs.

Although the work of Herzog *et al.* presented in [8] is extensive, the case involving individual receivers of a multicast group requesting different levels of QoS is covered on less than half a page. The discussion is very general and no specific cost-allocation mechanism is proposed for these scenarios.

The QoS-D ETS cost-allocation mechanism described by Liu *et al.* in [9] does however represent this approach. The costs corresponding to each QoS level are aggregated over the entire multicast tree, and divided uniformly among the receivers obtaining that level or higher. Thus, the lengths of the individual transmission paths are not taken into consideration. The statement in [9] concerning the cost sharing being proportional to the individual receivers' resource requirements, if unicast had been used, is therefore not strictly true.

4.4. Game-theoretic approaches

In game-theoretic approaches, the bandwidth allocation is incorporated with the pricing procedure. However, we aim for a cost-allocation mechanism that can fairly distribute the costs of any bandwidth allocation. The game-theoretic mechanisms are therefore ruled out.

4.5. Section summary

The game-theoretic approaches do not support cost-allocation of arbitrary bandwidth allocations, and none of the pure cost-allocation mechanisms takes both the transmission path and the QoS requirements into consideration. Hence, the mechanisms do not fully reflect the resource usage, and consequently there is room for improvements.

5. Terminology for multicast cost sharing

As mentioned in subsection 2.4, the notations for cost-allocation functions targeting multicast sessions with differentiated QoS levels, introduced by Herzog *et al.* in [8] and outlined in subsection 2.1, are not well defined. Thus, the decision was made to interpret and extend the terminology, in order to better suit multicast sessions that provide multiple QoS levels. This will prove to be useful in the following section, where two new cost-allocation mechanisms are proposed.

We define n_u^q and n_d^q to be the number of upstream and downstream receivers of the q^{th} QoS level (QoS^q), and let z_u^q and z_d^q denote the total number of upstream and downstream receivers utilizing the information corresponding to QoS^q. That is,

$$z_u^q = \sum_{x=q}^{Q} n_u^x$$

and

$$z_d^q = \sum_{x=q}^{Q} n_d^x,$$

given that there are Q available QoS levels. We also define the vectors

$$\mathbf{z}_u = \left\{z_u^1, z_u^2, \ldots, z_u^Q\right\}$$

and

$$\mathbf{z}_d = \left\{z_d^1, z_d^2, \ldots, z_d^Q\right\}.$$

Further, Herzog *et al.* not only allow the cost-allocation functions to control the division of the costs between receivers requesting the same QoS level, but also the distribution of the total cost among the different QoS levels. On the contrary, our opinion is that the cost-allocation functions should be general and not influence the distribution of the cost among the QoS levels. This distribution should instead fully reflect the resource requirements of each QoS level and the corresponding pricing made by the ISP in question.

Consequently, the cost vector

$$\mathbf{c} = \left\{c^1, c^2, \ldots, c^Q\right\}$$

is introduced, where the additional costs for supporting QoS^q on a particular link during a specific period of time, when compared to those of QoS^{q-1}, are denoted by c^q. These costs should reasonably be split among the receivers requiring QoS^q or higher, and two cost-allocation subfunctions, $\mathrm{f}_u^q(z_u^q, z_d^q)$ and $\mathrm{f}_d^q(z_u^q, z_d^q)$, are introduced for this purpose. These subfunctions describe the shares of the additional costs, for supporting QoS^q level, that should be allocated to the receivers of QoS^q or higher, both upstream and downstream of the link in question. The total cost that is to be allocated to the upstream and downstream receivers of QoS^q may now be written as

$$\mathrm{C}_u^q(\mathbf{z}_u, \mathbf{z}_d, \mathbf{c}) = \sum_{x=1}^{q} \mathrm{f}_u^x(z_u^x, z_d^x)c^x \tag{5}$$

and

$$\mathrm{C}_d^q(\mathbf{z}_u, \mathbf{z}_d, \mathbf{c}) = \sum_{x=1}^{q} \mathrm{f}_d^x(z_u^x, z_d^x)c^x, \tag{6}$$

respectively.

The two cost-allocation functions $\mathrm{C}_u^q(\mathbf{z}_u, \mathbf{z}_d, \mathbf{c})$ and $\mathrm{C}_d^q(\mathbf{z}_u, \mathbf{z}_d, \mathbf{c})$ represent the actual cost, whereas the original cost-allocation functions $\mathrm{F}_u(z_u, z_d)$ and $\mathrm{F}_d(z_u, z_d)$ described the fraction of the total cost to be allocated to the users. The condition (4), regarding full cost allocation, is therefore no longer valid. Instead, for the costs corresponding to each QoS level to be fully allocated, the following equation

$$z_u^q \cdot \mathrm{f}_u^q(z_u^q, z_d^q) + z_d^q \cdot \mathrm{f}_d^q(z_u^q, z_d^q) \geq 1, \tag{7}$$

must be fulfilled for all integers q between one and Q, where Q is the highest QoS level with a receiver downstream of the link in question.

If equation (7) is an equality for all integers q between one and Q, this guarantees that the sum of all allocated costs equals the sum of the costs according to equation (8), which means that the cost-allocation mechanism is budget balanced.

As an example, consider the QoS-D ETS cost-allocation mechanism described in subsection 2.4. Using the terminology introduced in this section, it is represented by cost-allocation subfunctions corresponding to the cost-allocation functions of the ETS mechanism (3)

$$\mathrm{f}_u^q(z_u^q, z_d^q) = \mathrm{f}_d^q(z_u^q, z_d^q) = \frac{1}{z_u^q + z_d^q}.$$

Consequently

$$\begin{aligned} &z_u^q \cdot \mathrm{f}_u^q(z_u^q, z_d^q) + z_d^q \cdot \mathrm{f}_d^q(z_u^q, z_d^q) \\ &= z_u^q \frac{1}{z_u^q + z_d^q} + z_d^q \frac{1}{z_u^q + z_d^q} = \frac{z_u^q + z_d^q}{z_u^q + z_d^q} = 1, \end{aligned}$$

and the QoS-D ETS mechanism is therefore budget balanced according to equation (8).

6. Fair cost-allocation strategies

The evaluation of existing cost-allocation mechanisms in Section 4 was concluded with the realization that none of them were satisfactorily fair. The reason was that, at most, they consider one of the two main factors affecting the resource usage, i.e. the transmission path and the QoS requirements. Using the terminology introduced in Section 5, a new cost-allocation mechanism, which takes both these factors into consideration, is proposed in subsection 6.1.

Although the aim of this mechanism is to achieve optimum fairness, it might have one, possibly severe, shortcoming: Optimum fairness may not be the primary interest of the receivers, if it occurs at the expense of higher costs. If poor and greedy receivers get a discount on the service, it may actually become cheaper for the rest of the receivers. An alternative mechanism is therefore proposed in subsection 6.2.

6.1. QoS-differentiated link split downstream

The first proposal is designed to perform perfectly fair cost allocations, taking into consideration both the transmission path and the QoS requirements. It builds on the ELSD cost-allocation mechanism, presented by Herzog *et al.* in [8], but is enhanced to support differentiated QoS levels.

The cost-allocation subfunctions therefore correspond to equations (1) and (2), and become

$$\mathrm{f}_u^q(z_u^q, z_d^q) = 0 \tag{9}$$

and

$$\mathrm{f}_d^q(z_u^q, z_d^q) = \frac{1}{z_d^q}, \tag{10}$$

respectively. This gives that

$$z_u^q \cdot \mathrm{f}_u^q(z_u^q, z_d^q) + z_d^q \cdot \mathrm{f}_d^q(z_u^q, z_d^q) = z_u^q \cdot 0 + z_d^q \frac{1}{z_d^q} = \frac{z_d^q}{z_d^q} = 1,$$

and the cost-allocation mechanism is consequently budget balanced according to equation (8).

Substituting equations (9) and (10) into (5) and (6), the main cost-allocation functions for receivers of QoS^q become

$$\mathrm{C}_u^q(\mathbf{z}_u, \mathbf{z}_d, \mathbf{c}) = 0 \tag{11}$$

and

$$\mathrm{C}_d^q(\mathbf{z}_u, \mathbf{z}_d, \mathbf{c}) = \sum_{x=1}^{q} \frac{c^x}{z_d^x}. \tag{12}$$

We call the cost-allocation mechanism described by equations (11) and (12), the *QoS-differentiated link split downstream* (QoS-D LSD) mechanism.

6.1.1. Bandwidth-differentiated link split downstream. As observed in [8], in the extreme case, each receiver will have a QoS level of its own. This can be taken one step further, by assuming the bandwidth to be the predominant cost factor and considering the bandwidth consumption as a direct function of the QoS level. Let us also assume that the bandwidth is uniformly priced and costs c *monetary units* (MU) per *bitrate unit* (BU) and *time unit* (TU).

Let $\mathbf{b}$ be a vector whose first element $\mathbf{b}[0]$ is 0 and the n_d following elements are the receiving rates of the receivers downstream of the link in question, sorted in ascending order. The total cost per TU, allocated to the downstream receiver obtaining the q^{th} smallest bandwidth, may now be rewritten as

$$\mathrm{C}_d^q(n_d, \mathbf{b}) = c \sum_{x=1}^{q} \frac{\mathbf{b}[x] - \mathbf{b}[x-1]}{n_d - x + 1}. \tag{13}$$

$$
\begin{aligned}
&\sum_{q=1}^{Q}\left(n_u^q\cdot \mathrm{C}_u^q(\mathbf{z}_u,\mathbf{z}_d,\mathbf{c})+n_d^q\cdot \mathrm{C}_d^q(\mathbf{z}_u,\mathbf{z}_d,\mathbf{c})\right)\\
&=\sum_{q=1}^{Q}\left(n_u^q\cdot\sum_{x=1}^{q}\mathrm{f}_u^x(z_u^x,z_d^x)c^x+n_d^q\cdot\sum_{x=1}^{q}\mathrm{f}_d^x(z_u^x,z_d^x)c^x\right)\\
&=\left(n_u^1\cdot\mathrm{f}_u^1(z_u^1,z_d^1)c^1+n_d^1\cdot\mathrm{f}_d^1(z_u^1,z_d^1)c^1\right)\\
&+\left(n_u^2\cdot\left(\mathrm{f}_u^1(z_u^1,z_d^1)c^1+\mathrm{f}_u^2(z_u^2,z_d^2)c^2\right)+n_d^2\cdot\left(\mathrm{f}_d^1(z_u^1,z_d^1)c^1+\mathrm{f}_d^2(z_u^2,z_d^2)c^2\right)\right)+\cdots\\
&\cdots+\left(n_u^Q\cdot\left(\mathrm{f}_u^1(z_u^1,z_d^1)c^1+\mathrm{f}_u^2(z_u^2,z_d^2)c^2+\cdots+\mathrm{f}_u^Q(z_u^Q,z_d^Q)c^Q\right)\right.\\
&\left.+n_d^Q\cdot\left(\mathrm{f}_d^1(z_u^1,z_d^1)c^1+\mathrm{f}_d^2(z_u^2,z_d^2)c^2+\cdots+\mathrm{f}_d^Q(z_u^Q,z_d^Q)c^Q\right)\right)\\
&=\left(\mathrm{f}_u^1(z_u^1,z_d^1)c^1\cdot\left(n_u^1+n_u^2+\cdots+n_u^Q\right)+\mathrm{f}_d^1(z_u^1,z_d^1)c^1\cdot\left(n_d^1+n_d^2+\cdots+n_d^Q\right)\right)\\
&+\left(\mathrm{f}_u^2(z_u^2,z_d^2)c^2\cdot\left(n_u^2+n_u^3+\cdots+n_u^Q\right)+\mathrm{f}_d^2(z_u^2,z_d^2)c^2\cdot\left(n_d^2+n_d^3+\cdots+n_d^Q\right)\right)+\cdots\\
&\cdots+\left(\mathrm{f}_u^Q(z_u^Q,z_d^Q)c^Q\cdot n_u^Q+\mathrm{f}_d^Q(z_u^Q,z_d^Q)c^Q\cdot n_d^Q\right)\\
&=\sum_{q=1}^{Q}\left(\mathrm{f}_u^q(z_u^q,z_d^q)c^q\cdot\sum_{x=q}^{Q}n_u^x+\mathrm{f}_d^q(z_u^q,z_d^q)c^q\cdot\sum_{x=q}^{Q}n_d^x\right)\\
&=\sum_{q=1}^{Q}c^q\cdot\left(z_u^q\cdot\mathrm{f}_u^q(z_u^q,z_d^q)+z_d^q\cdot\mathrm{f}_d^q(z_u^q,z_d^q)\right)=\sum_{q=1}^{Q}c^q
\end{aligned}
\tag{8}
$$

The *bandwidth-differentiated link split downstream* cost allocation performed by equation (13) is only a special case of the QoS-D LSD mechanism.

6.1.2. A cost-allocation example. As a small example of the QoS-D LSD mechanism, let us study how equation (13) allocates the cost of link l in Figure 1, where t is the transmitter and r_1 through r_7 are the receivers. For simplicity, we assume that receiver r_i obtains i BU for one TU, and that the bandwidth on link l costs one MU per BU and TU. Now we have

$$
\begin{aligned}
c &= 1\\
n_d &= 3\\
\mathbf{b} &= \{0,1,2,3\},
\end{aligned}
$$

which when substituted into equation (13) give the cost of link l being allocated to receiver r_1, r_2, and r_3 as follows,

$$
\mathrm{C}_d^1(n_d,\mathbf{b})=\sum_{x=1}^{1}\frac{\mathbf{b}[x]-\mathbf{b}[x-1]}{4-x}=\tfrac{1}{3}\text{ MU},
$$

$$
\mathrm{C}_d^2(n_d,\mathbf{b})=\sum_{x=1}^{2}\frac{\mathbf{b}[x]-\mathbf{b}[x-1]}{4-x}=\tfrac{1}{3}+\tfrac{1}{2}=\tfrac{5}{6}\text{ MU},
$$

and

$$
\begin{aligned}
\mathrm{C}_d^3(n_d,\mathbf{b})&=\sum_{x=1}^{3}\frac{\mathbf{b}[x]-\mathbf{b}[x-1]}{4-x}=\tfrac{1}{3}+\tfrac{1}{2}+\tfrac{1}{1}\\
&=\tfrac{11}{6}\text{ MU}.
\end{aligned}
$$

If we, similarly, calculate the total costs allocated to receiver r_1, r_2, and r_3, link by link from the source, they become

$$
\left(\tfrac{1}{4}\right)+\left(\tfrac{1}{3}\right)+\left(\tfrac{1}{1}\right)=\tfrac{19}{12}\text{ MU},
$$

$$
\left(\tfrac{1}{4}+\tfrac{1}{3}\right)+\left(\tfrac{1}{3}+\tfrac{1}{2}\right)+\left(\tfrac{2}{2}\right)+\left(\tfrac{2}{1}\right)=\tfrac{53}{12}\text{ MU},
$$

and

$$
\begin{aligned}
&\left(\tfrac{1}{4}+\tfrac{1}{3}+\tfrac{1}{2}\right)+\left(\tfrac{1}{3}+\tfrac{1}{2}+\tfrac{1}{1}\right)+\left(\tfrac{2}{2}+\tfrac{1}{1}\right)+\left(\tfrac{3}{1}\right)\\
&=\tfrac{95}{12}\text{ MU},
\end{aligned}
$$

respectively. To make the calculations easier to follow, the costs are presented for every bandwidth interval, and costs arising from the same link are grouped together by parentheses.

The costs allocated to all the seven receivers in the multicast tree are presented in Table 1, together with the corresponding costs produced by the ETS, ELSD, and QoS-D ETS cost-allocation mechanisms.

The ETS and ELSD mechanisms were not designed with differentiated QoS demands in mind. Both these mechanisms will therefore generally allocate disproportionately large parts of the cost to receivers with low QoS demands. The ETS mechanism simply splits the aggregated cost of the entire multicast tree equally among all the receivers, and is therefore also unfair towards receivers with short transmission paths. The ELSD mechanism only splits the link costs among downstream receivers, and the receivers that are treated most unfairly are consequently those with low QoS demands, compared to the receivers with whom they share the links. Examples of such mistreated receivers are consequently r_1, r_2, and r_5.

Table 1. The obtained bitrates in BUs of the seven receivers in the example, together with the costs in MUs, allocated by the ETS, ELSD, QoS-D ETS, and QoS-D LSD cost-allocation mechanisms.

receiver	rate	ETS	ELSD	QoS-D ETS	QoS-D LSD
r_1	1	7.29	3.00	1.71	1.58
r_2	2	7.29	5.50	3.55	4.42
r_3	3	7.29	6.50	5.55	7.92
r_4	4	7.29	5.00	7.30	6.08
r_5	5	7.29	10.3	8.96	9.17
r_6	6	7.29	11.3	11.0	11.7
r_7	7	7.29	9.33	13.0	10.2

The QoS-D ETS mechanism performs differently, as it is now the receivers with short transmission paths, such as r_4 and r_7, that are treated unfairly. The situation is worst for r_7, which obtains the highest QoS level, and therefore has to share the costs of the entire multicast tree.

6.2. Bid-based link split downstream

As mentioned previously, the proposed QoS-D LSD cost-allocation mechanism attempts to achieve optimum fairness, but it has one possibly severe shortcoming: Optimum fairness may not be the primary interest of the receivers if it is at the expense of higher costs. If poor and greedy receivers get a discount on the service, it may actually become cheaper for the rest of the receivers. Here we further investigate this issue and propose an alternative cost-allocation mechanism that solves the shortcoming.

We start by drawing a parallel to an everyday situation. Children and/or retired people often receive a discount on the entrance fee to sport events, festivals, and museums etc. Most people are willing to accept this since it typically does not negatively affect their fees. As long as the events are not sold out, the economy of the organizers might actually benefit from this, and thereby allow them to also lower the standard fees[1].

However, if the scenario was the opposite and the attendance of discounted groups had a negative influence on standard fees, i.e. forcing the regular visitors to subsidize those on discounted rates, few would be happy about accepting such a system. Consumer goods are seldom discounted in this manner, since they are associated with specific material and production costs.

[1]If any organizers actually do this in reality is a completely different question.

Table 2. Possible outcomes of a placed bid, with a certain maximum cost, for the BB LSD cost-allocation mechanism.

relative size of the maximum cost	served	allocated cost
max cost $<$ additional cost	no	–
additional cost $\leq$ max cost $<$ fair share	yes	max cost
fair share $\leq$ max cost	yes	fair share

If we look at the game-theoretic approaches of Section 3, the SV mechanism allocates the costs in a LSD manner, and therefore shares the aforementioned shortcoming. The MC mechanism on the other hand does not require the receivers to cover more than their marginal cost. It is consequently not budget balanced, and may thereby produce a financial deficit for the ISPs.

We propose a bid-based cost-allocation mechanism, where fair cost allocation according to the QoS-D LSD mechanism is retained as the target. However, bids that do not cover the receivers' fair shares of the costs, but do cover at least the *additional cost* associated with receivers' requests, are also accepted. That is, the additional cost for providing the receiver with the requested service, compared to the cost of providing the service to the existing set of receivers.

The main difference between marginal cost and additional cost is that the latter is dependent upon the order of the arrival of the bids which, in turn, guarantees that the proposed mechanism is budget balanced. However, although an expansion of the user set never causes increased costs for users within the original set, the mechanism is not cross monotonic, since it only applies to ordered sets of users.

A placed bid consequently leads to one of the outcomes described in Table 2. The fair share is the cost calculated according to the principles of the QoS-D LSD mechanism, with the addendum that if some poor receivers are discounted, these costs have to be carried by the wealthier receivers. The costs not covered by a receiver are distributed between the affected links and QoS levels of the existing transmission tree, proportional to that receiver's fair cost shares, and are split among the higher-bidding receivers utilizing these resources. The proposed mechanism is called *bid-based link split downstream* (BB LSD).

6.2.1. Bid structure. There are a number of mandatory parameters that a bid must contain to make the BB LSD possible, namely:

- the maximum acceptable cost of the transmission
- the requested duration of the transmission

Table 3. The bids of the receivers in the example in subsection 6.3. The maximum cost is measured in MU.

receiver	requested QoS	maximum cost
r_1	QoS^1	25
r_2	QoS^2	25
r_3	QoS^3	80
r_4	QoS^4	100

- the requested QoS level of the transmission
- the *time to live* (TTL) of the bid

It is insufficient to replace the maximum cost and requested duration with a maximum cost per TU. This would prevent the calculation of other receivers' maximum costs, since these are affected by receivers who leave the service prematurely. There is also a possibility of non-recurrent costs associated with setting up the service. The bid TTL is required since most users are only interested in a particular service if it can be started within a given amount of time.

A receiver may request a service at a particular price, but be willing to settle for a poorer QoS level at a lower price if the main bid cannot be accepted. The main bid could then possibly remain effective during its TTL, in case the costs associated with it were to be reduced. We observe that there may be as many subbids as there are QoS levels, but do not discuss these composite bids any further.

6.2.2. Strategyproofness. The BB LSD mechanism is not strategy proof. There is an obvious risk that users place dishonestly low bids, i.e. bids that do not correspond to their estimated value of the service, in an attempt to find the minimum cost of the service. To avoid this destructive behavior for the system, we propose an upper limit on the bid frequency of any particular receiver. This might not make the mechanism strategy proof, but it should make users more honest, since a lower bid equals a higher risk of missing out on the service for a particular amount of time.

The problem of finding a sufficient maximum bid frequency is a weighing of the honesty of the bids against the adaptability of the mechanism. It is possible that the economic prerequisites of a receiver change for the better after a low bid has been placed. An alternative to a fixed maximum bid frequency, is to exponentially increase the period of time until a new bid might be placed or considered.

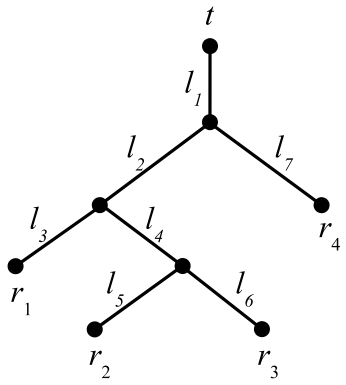


Figure 2. The multicast transmission tree of the example in subsection 6.3.

6.3. A cost-allocation example

The transmission tree in Figure 2 is used as an example in order to shed some light on the possible advantages of the BB LSD cost-allocation mechanism. The requested QoS levels are outlined in Table 3, together with the maximum total cost that the receivers are willing to pay for the service. For simplicity, assume that all requests concern the same duration, say 10 TU, and that the bandwidth on all links cost one MU per BU and TU. Further assume that the bitrate is the predominant cost factor and that QoS^q constantly requires q BU. The incremental cost of transmitting QoS^q, when compared to that of QoS^{q-1}, is consequently one MU/TU per link.

In the two first subsections, the QoS-D LSD and MC cost-allocation mechanisms are utilized to allocate the bandwidth and costs, and in the third subsection, these parameters are calculated according to the proposed BB LSD mechanism. For the latter mechanism, the order of arrival of the bids is essential. For simplicity, we base the order on the receiver numbers, and assume the arrivals of the bids to be sufficiently closely spaced in time for the requested transmissions to be considered simultaneously from a cost-sharing perspective. The results of the cost-allocation mechanisms are compared in the last subsection.

6.3.1. Allocation according to QoS-D LSD. We start by studying how the QoS-D LSD mechanism would allocate the cost of link l_2, under the assumption that all receivers are able to obtain the requested service at prices not exceeding their maximum costs. According to equation (12), receiver r_1 will be charged

$$\tfrac{10}{3} \approx 3.33 \text{ MU}$$

for receiving QoS^1, since there are three receivers utilizing

this information. In the same manner, the cost of link l_2 allocated to receivers r_2 and r_3, which are requesting QoS^2 respectively QoS^3, become

$$\tfrac{10}{3} + \tfrac{10}{2} \approx 8.33 \text{ MU}$$

and

$$\tfrac{10}{3} + \tfrac{10}{2} + \tfrac{10}{1} \approx 18.33 \text{ MU}.$$

The cost of each receiver can be calculated link by link from the source. The total costs of receivers r_1 through r_4 then become

$$\begin{gathered}\left(\tfrac{10}{4}\right) + \left(\tfrac{10}{3}\right) + \left(\tfrac{10}{1}\right) \approx 15.83 \text{ MU}, \\ \left(\tfrac{10}{4} + \tfrac{10}{3}\right) + \left(\tfrac{10}{3} + \tfrac{10}{2}\right) + \left(\tfrac{10}{2} + \tfrac{10}{2}\right) + \\ \left(\tfrac{10}{1} + \tfrac{10}{1}\right) \approx 44.17 \text{ MU}, \end{gathered} \tag{14}$$

$$\begin{gathered}\left(\tfrac{10}{4} + \tfrac{10}{3} + \tfrac{10}{2}\right) + \left(\tfrac{10}{3} + \tfrac{10}{2} + \tfrac{10}{1}\right) + \\ \left(\tfrac{10}{2} + \tfrac{10}{2} + \tfrac{10}{1}\right) + \left(\tfrac{10}{1} + \tfrac{10}{1} + \tfrac{10}{1}\right) \approx 79.17 \text{ MU}, \end{gathered} \tag{15}$$

and

$$\begin{gathered}\left(\tfrac{10}{4} + \tfrac{10}{3} + \tfrac{10}{2} + \tfrac{10}{1}\right) + \left(\tfrac{10}{1} + \tfrac{10}{1} + \tfrac{10}{1} + \tfrac{10}{1}\right) \\ \approx 60.83 \text{ MU},\end{gathered}$$

respectively. To make the calculations easier to follow, the costs arising from the same link are grouped by parentheses.

Apparently, the assumption that all receivers are able to obtain the service, at a cost not exceeding their maximum limits, was false. Receiver r_2 is only willing to pay 25 MU, but would be charged over 44 MU. It will therefore not obtain the service, and the rest of the receivers will consequently have to cover a larger part of the costs on the shared links. Receivers r_1, r_3, and r_4 will now be charged

$$\left(\tfrac{10}{3}\right) + \left(\tfrac{10}{2}\right) + \left(\tfrac{10}{1}\right) \approx 18.33 \text{ MU},$$

$$\begin{gathered}\left(\tfrac{10}{3} + \tfrac{10}{2} + \tfrac{10}{2}\right) + \left(\tfrac{10}{2} + \tfrac{10}{1} + \tfrac{10}{1}\right) + \\ \left(\tfrac{10}{1} + \tfrac{10}{1} + \tfrac{10}{1}\right) + \left(\tfrac{10}{1} + \tfrac{10}{1} + \tfrac{10}{1}\right) \approx 103.33 \text{ MU},\end{gathered}$$

respectively

$$\begin{gathered}\left(\tfrac{10}{3} + \tfrac{10}{2} + \tfrac{10}{2} + \tfrac{10}{1}\right) + \left(\tfrac{10}{1} + \tfrac{10}{1} + \tfrac{10}{1} + \tfrac{10}{1}\right) \\ \approx 63.33 \text{ MU}.\end{gathered}$$

Hence, the cost allocated to receiver r_3 exceeds its bid of 80 MU, and it will also fail to obtain the requested service. The costs of receivers r_1 and r_4 are increased accordingly to

$$\left(\tfrac{10}{2}\right) + \left(\tfrac{10}{1}\right) + \left(\tfrac{10}{1}\right) = 25.00 \text{ MU}$$

and

$$\begin{gathered}\left(\tfrac{10}{2} + \tfrac{10}{1} + \tfrac{10}{1} + \tfrac{10}{1}\right) + \left(\tfrac{10}{1} + \tfrac{10}{1} + \tfrac{10}{1} + \tfrac{10}{1}\right) \\ = 75.00 \text{ MU},\end{gathered}$$

respectively. Finally, all the costs are covered by the receivers' bids.

6.3.2. Allocation according to MC. The MC cost-allocation mechanism has received its name because it allocates the marginal cost to each user. The marginal cost of a user is the additional cost of providing the service to that user, when compared to the cost of providing the service to the remaining set of users.

In this example the marginal cost of receiver r_1 corresponds to that of QoS^1 on link l_3, i.e. 10 MU, since r_2 and r_3 also utilize QoS^1 on the rest of the transmission path from the source to r_1. On link l_1, QoS^1 is also utilized by receiver r_4.

In the same manner, the marginal cost of receiver r_2 is derived from the provision of QoS^2 on link l_5, that is 20 MU. On the rest of the transmission path from the source to r_2, QoS^2 is shared by receiver r_3.

Receiver r_3 is allocated the total cost for QoS^3 on its last hop link l_6, which corresponds to 30 MU. Further, on links l_2 and l_4, r_3 is the only receiver that utilizes QoS^3. It therefore has to cover the additional cost of QoS^3, when compared to that of QoS^2, on these links. This implies a cost of 10 MU per link. However, r_3 does not have to contribute to the costs of l_1, since QoS^3 is shared with receiver r_4 on that link. The aggregated cost allocated to receiver r_3 is consequently 50 MU.

Finally, receiver r_4 is charged with the total cost of QoS^4 on link l_7 and the additional cost of QoS^4 on link l_1. This adds up to a total of 50 MU, and all receivers will therefore be served since the maximum costs of their bids cover the allocated costs.

6.3.3. Allocation according to BB LSD. Now the proposed BB LSD cost-allocation mechanism is applied to the same example.

When the bid of receiver r_1 is placed, its maximum cost of 25 MU is insufficient to cover the cost of the requested QoS^1, which is calculated to 30 MU over the three-link transmission path from the source. The bid is therefore not accepted, but remains effective, pending other bids that may share the costs.

When the bid of receiver r_2 arrives, the costs associated with its request for QoS^2 is 80 MU. The bid is on 25 MU, and can therefore not be accepted either, not even when considered jointly with the bid of r_1.

Then the bid of receiver r_3 is placed. It concerns QoS^3 and is worth 80 MU, whereas the cost for offering the service is 120 MU. The total cost for serving r_1, r_2, and r_3 would be 150 MU, whereas their joint means are calculated as being 130 MU. Separately considering r_1 and r_3, or r_2 and r_3, does not make the situation more favorable.

Finally, the bid of receiver r_4 is placed. The costs for

the requested transmission to r_4 is 80 MU, and the bid on 100 MU can therefore be accepted on its own. However, to decide what costs will actually be allocated to r_4, the bids of the other receivers must first be reconsidered.

Let us start by considering receiver r_2. The costs of the resources that r_2 must cover in total, i.e. those of link l_5, are 20 MU according to the last parenthesis of equation (14). It therefore has 5 MU left to contribute to the cost sharing on the upstream links. These 5 MU will be split uniformly according to r_2's fair shares of the costs on these links, which corresponds to the remaining first three parenthesis of (14). This results in

$$5 \cdot \frac{\frac{10}{4}}{\left(\frac{10}{4}+\frac{10}{3}\right)+\left(\frac{10}{3}+\frac{10}{2}\right)+\left(\frac{10}{2}+\frac{10}{2}\right)} \approx 0.52 \text{ MU}$$

for QoS^1 on l_1, and in the same manner approximately 0.69 MU for QoS^2 on l_1 and QoS^1 on l_2, and 1.03 MU for QoS^2 on l_2, and QoS^1 and QoS^2 on l_4.

Receiver r_3 must cover the entire 30 MU for link l_6 and the remaining costs on l_4. Further, it also has to cover the additional cost of QoS^3 on l_2 together with the remaining cost for QoS^2. Consequently, there are approximately

$$80 - 30 - (30 - 2 \cdot 1.03) - (20 - 1.03) \approx 3.09 \text{ MU}$$

left on the bid of r_3. Split uniformly according to r_3's remaining costs shares, which can be found in equation (15), this yields

$$3.09 \cdot \frac{\frac{10}{4}}{\left(\frac{10}{4}+\frac{10}{3}+\frac{10}{2}\right)+\left(\frac{10}{3}\right)} \approx 0.55 \text{ MU}$$

for QoS^1 on link l_1, and in the same manner approximately 0.73 MU for QoS^2 on l_1, 1.09 MU for QoS^3 on l_1, and 0.73 MU for QoS^1 on l_2.

Consequently, receiver r_1 that only requested QoS^1, has to cover 10 MU on link l_3 and the remaining costs on l_2, which is approximately

$$10 - 0.69 - 0.73 = 8.58 \text{ MU}.$$

On link l_1, r_1 will be charged with its own fair share of the costs, plus its share of the costs for QoS^1 that are not covered by r_2 and r_3. This adds up to

$$\frac{10}{4} + \frac{\frac{10}{4} - 0.52}{2} + \frac{\frac{10}{4} - 0.55}{2} \approx 4.47 \text{ MU}.$$

The total cost allocated to r_1 thereby aggregates into approximately

$$10 + 8.58 + 4.47 = 23.05 \text{ MU}.$$

The remaining costs, which are allocated to receiver r_4, are calculated as being 40 MU for link l_7, and approximately

$$(10 - 4.47 - 0.52 - 0.52) + (10 - 0.69 - 0.73) + (10 - 1.09) + 10 = 31.98 \text{ MU}$$

Table 4. The outcomes for the receivers with the QoS-D LSD, MC and BB LSD cost-allocation mechanisms. The costs are measured in MU.

	QoS-D LSD		MC		BB LSD	
receiver	served	cost	served	cost	served	cost
r_1	yes	25.0	yes	10.0	yes	23.0
r_2	no	–	yes	20.0	yes	25.0
r_3	no	–	yes	50.0	yes	80.0
r_4	yes	75.0	yes	50.0	yes	72.0

Table 5. The announced costs of the provided services and the generated incomes, both measured in MU, with the QoS-D LSD, MC and BB LSD cost-allocation mechanisms.

	QoS-D LSD	MC	BB LSD
announced service costs	100	200	200
generated incomes	100	130	200

for link l_1, where each QoS level is accounted for separately. This gives a total cost for receiver r_4 of approximately 71.98 MU.

6.3.4. Comparison of results. In Table 4, the outcomes for the receivers with the proposed BB LSD cost-allocation mechanism are presented together with them of MC and QoS-D LSD.

The most obvious difference between the BB LSD and QoS-D LSD mechanisms is that receivers r_2 and r_3 are served by BB LSD but not by QoS-D LSD, since they cannot fully cover their fair shares of the costs. As a consequence, the costs allocated to receivers r_1 and r_4 are somewhat lower for the BB LSD mechanism, where receiver r_3 contributes to the cost sharing on links l_1 and l_2. Another, more significant effect, which is apparent in Table 5, is that the income of the ISP is doubled through the use of the BB LSD mechanism.

The BB LSD and MC mechanisms serve the same user sets. However, all the receivers are allocated lower costs by using the MC mechanism, since it only charges the marginal costs. As can be seen in Table 5, the result is, if not a financial deficit, at least a 70 MU reduction of the ISP's revenue, when compared to the budget-balanced BB LSD mechanism.

7. Conclusion

This paper has aimed at more efficient usage of bandwidth in IP networks. The area that has been targeted is the slow deployment of multicast transmission. The proposal was to reduce the costs for users of multicast sessions. The cost reduction is brought about by the resource savings offered by the bandwidth sharing.

Fair cost sharing among multicast receivers has been addressed. This would favor the multicast receivers under the assumption that fair cost sharing should be based upon resource usage. Two major resource-related factors were observed; the transmission path and the bandwidth or QoS requirements. Existing cost-allocation mechanisms for multicast were evaluated, but none took both these parameters into consideration. The QoS-D LSD cost-allocation mechanism was therefore proposed. It considers both the transmission path and the QoS requirements, in order to achieve optimum fairness.

However, optimum fairness might not be in the best interest of the users, when it is at the expense of higher costs. An alternative cost-allocation mechanism, BB LSD, was therefore proposed. The BB LSD mechanism enables the users to place bids for a requested service, revealing their maximum acceptable cost. A bid that does not cover the user's fair share of the costs for the requested service is nevertheless accepted if it does cover at least the additional cost associated with the request. This guarantees that the BB LSD mechanism is budget balanced. The result is not only a possible reduction in the costs for the rest of the users, but also an increase in revenue for the ISPs, which are able to serve more users.

Unfortunately, the BB LSD mechanism is not strategy proof. To avoid users seeking the minimum cost by placing dishonestly low bids, an upper limit on the bid frequency of any particular receiver was therefore proposed. Another alternative would be an exponentially growing time out in the case of a rejected bid. This should make the users more honest, i.e. to bid closer to what the service is worth to them, since a lower bid equals a higher risk of missing out on the service.

7.1. Future work

Future research about cost-allocation mechanisms may involve the problem of finding a sufficient maximum bid frequency, or other procedures to mitigate the fact that the BB LSD mechanism is not strategy proof. Another alternative might be the search for a completely new mechanism that is naturally strategy proof and still possesses as many of the BB LSD mechanism's attractive properties as possible.

Further research topics are the implementation of the QoS-D LSD and BB LSD cost-allocation mechanisms, and the process of actually charging the receivers with the allocated costs.

Acknowledgment

The work was financed in part by the Regional Fund of the European Union and the County Administrative Board of Västernorrland.

References

[1] M. Adler and D. Rubenstein. Pricing multicasting in more flexible network models. *ACM Transactions on Algorithms*, 1(1):48–73, July 2005.

[2] M. Bläser. Approximate budget balanced mechanisms with low communication costs for the multicast cost-sharing problem. In *Proceedings of 15th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 625–195, New Orleans, LA, USA, January 2004.

[3] A. Bueno, P. Vila, and R. Fabregat. Multicast extension of unicast charging for QoS services. In *Proceedings of 4th IEEE European Conference on Universal Multiservice Networks (ECUMN)*, pages 119–126, Toulouse, France, February 2007.

[4] S. Chawla, D. Kitchin, U. Rajan, R. Ravi, and A. Sinha. Profit guaranteeing mechanisms for multicast networks. In *Proceedings of 4th ACM Conference on Electronic Commerce (EC)*, pages 190–191, San Diego, CA, USA, June 2003.

[5] H. J. Einsiedler, P. Hurley, B. Stiller, and T. Braun. Charging multicast communications based on a tree metric. In *Proceedings of GI Multicast Workshop*, Braunschweig, Germany, May 1999.

[6] J. Feigenbaum, C. H. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *Elsevier Journal of Computer and System Sciences*, 63(1):21–41, August 2001.

[7] N. Garg and D. Grosu. Performance evaluation of multicast cost sharing mechanisms. In *Proceedings of 21st IEEE International Conference on Advanced Networking and Applications (AINA)*, pages 901–908, Niagara Falls, Canada, May 2007.

[8] S. Herzog, S. Shenker, and D. Estrin. Sharing the "cost" of multicast trees: An axiomatic analysis. *IEEE/ACM Transactions on Networking*, 5(6):847–860, December 1997.

[9] C.-C. Liu, S.-C. Chang, and H.-H. Cheng. Pricing and fee sharing for point to multipoint and quality guaranteed multicast services. In *Proceedings of 7th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pages 255–260, Iwate, Japan, July 2000.

[10] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. *ACM SIGCOMM Computer Communication Review*, 26(4):117–130, October 1996.

[11] H. Moulin and S. Shenker. Strategyproof sharing of submodular access costs: Budget balance versus efficiency. *Springer Journal on Economic Theory*, 18(3):511–533, 2001.

[12] S. Shenker, D. Clark, D. Estrin, and S. Herzog. Pricing in computer networks: Reshaping the research agenda. *ACM SIGCOMM Computer Communication Review*, 26(2):19–43, April 1996.

[13] P. Österberg and T. Zhang. Revised definition of multicast-favorable max-min fairness. In *Proceedings of 3^{rd} IASTED International Conference on Communications and Computer Networks (CCN)*, pages 63–68, Lima, Peru, October 2006.

[14] P. Österberg and T. Zhang. Bid-based cost sharing among multicast receivers. In *Proceedings of 4^{th} ACM International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine)*, Vancouver, BC, Canada, August 2007.

[15] P. Österberg and T. Zhang. Fair cost sharing among multicast receivers. In *Proceedings of 2^{nd} IARIA International Conference on Digital Telecommunications (ICDT)*, Silicon Valley, CA, USA, July 2007.

Modeling Systems with Multi-service Overflow Erlang and Engset Traffic Streams

Mariusz Głąbowski
Poznań University of Technology
ul. Piotrowo 3a, 60-965 Poznań
Email: mariusz.glabowski@et.put.poznan.pl

Abstract

The article proposes analytical methods for determining traffic characteristics of hierarchically organised telecommunication networks which are offered multi-service traffic streams. The article proposes a method for determining occupancy distribution in the group servicing multi-service overflow traffic. This method is based on modification of the Kaufman-Roberts recursion – elaborated for the full-availability group with Poisson calls streams – and uses Fredericks & Hayward approximation. Additionally, a method for determining parameters of the traffic overflowing from primary groups servicing PCT1[1] and PCT2[2] traffic streams is also presented.

Keywords: *overflow traffic, PCT1, PCT2, multi-rate traffic*

1. Introduction

Modeling telecommunication networks employing the strategy of redirecting traffic via alternative routes, i.e. systems with traffic overflow is a complex issue. This problem comes down to resolving the two following basic problems, namely: to a determination of traffic characteristics of traffic that overflows from direct (primary) groups (with high loss coefficients usually), and a determination of the number the so-called Basic Bandwidth Units (or channels) in alternative groups (with low loss coefficients usually), where the loss coefficients will not exceed the assigned value.

Systems with overflow traffic have been widely discussed e.g., in [8,22,35]. The above mentioned works, however, have dealt with single-rate traffic only, i.e. with traditional single-service telephone networks. There have been developed both exact [4,14,25,36] and approximate [15,35] models of the full-availability group with overflow traffic assuming Poisson distribution of calls streams and the exponential distribution of holding time for calls offered to the primary groups. The problem of modeling the groups with overflow traffic under assumption of hyper-exponential distribution of the holding time has been described in [27] while single-rate traffic systems with overflow traffic and finite number of traffic sources (PCT2) have been considered e.g., in [26].

The basic method for determining traffic characteristics of multi-service systems employs the so-called Kaufman-Roberts formulas (KR) [19, 24]. These equations allow to reliably model systems with PCT1 streams that are offered directly to the primary groups of telecommunication networks. The traffic that is not serviced in such groups is overflowed to an alternative group. This part of traffic is called the overflow traffic. However, even if the streams that are offered directly to the primary groups are of type PCT1, the calls stream overflowing from the primary group does not agree with the Poison distribution [35].

Overflow calls can appear only in the occupancy time of all Basic Bandwidth Units of the primary group. This means that the overflow stream is more "concentrated" in certain time periods, i.e. is characterized by greater "peakedness" as compared with PCT1 traffic. If identical values of offered traffic and the congestion are assumed, then a greater number of Basic Bandwidth Units (BBUs) is required for servicing overflow traffic than that required for servicing PCT1 traffic.

The following parameters can be used for statistical evaluation of the overflow stream: the mean value R of overflow traffic (the first moment of the probability distribution of the number of calls) and the second moment with the corresponding variance σ^2. With the help of those two parameters it is possible to determine "unevenness" of the overflow stream by the introduction of the concept of the peakedness

[1] PCT1 – Pure Chance Traffic Type One – type of traffic in which we assume that the service times are exponentially distributed and the arrival process is a Poisson process. This type of traffic is known as Erlang traffic.

[2] PCT2 – Pure Chance Traffic type Two – type of traffic in which we assume that the service times are exponentially distributed and the arrival process is formed by the limited number of sources. This type of traffic is known as Engset traffic.

coefficient Z that is equal to the ratio of the variance σ^2 to the mean value of overflow traffic R:

$$Z = \sigma^2/R. \quad (1)$$

The "unevenness" of the overflow stream can also be evaluated by the application of the parameter D that is the difference between the variance and the mean value of overflow traffic:

$$D = \sigma^2 - R. \quad (2)$$

It is noticeable that the parameters Z and D take the following values for the offered traffic, serviced traffic and the overflowed traffic:

- for offered traffic: $Z = 1$ and $D = 0$,
- for serviced traffic on the primary group (smooth traffic): $Z < 1$ and $D < 0$,
- for overflow traffic: $Z > 1$ and $D > 0$.

The service process of a Poisson calls stream in a full-availability group can be thus characterized by four parameters A, V, R, σ^2 (σ^2 can be replaced by Z or D). The stream offered to the group is here determined by one parameter A – the mean value of the offered traffic, whereas the overflow traffic stream by two: the mean value of the overflow traffic R and its variance σ^2.

Having the above in mind, we can come to a conclusion that the KR equations in their basic form (devised with the assumption of the exponential distribution of time gaps between calls) cannot be applied to determining call blocking coefficients in multi-service traffic in the alternative group. The problem of modeling the full-availability group with overflow traffic with known value of parameter Z was taken in [7], and then in [20, 34]. The methods for modeling the systems with multi-service overflow traffic (under the assumption of infinite number of traffic sources) including the methods for determining parameters of overflow traffic, an occupancy distribution in alternative groups and dimensioning systems with multi-service overflow traffic was presented in [10, 11, 13].

The other group of methods, enabling modeling the systems with overflow traffic, are the methods based on Markov-Modulated Poisson Processes, published in [6, 17, 21]. Among this group of methods, the highest accuracy, in case of multi-service systems, assures the method proposed in [6]. The accuracy of this method is related to high computational complexity of the process of calculating the variance of overflow traffic based on analysis of multidimensional Markov process in the system composed of two groups, i.e. the primary group and the alternative group. Exponential order of computational complexity (in function of number of classes of calls) makes practical application of this method very difficult.

The purpose of the article is the proposition of a consistent methodology for determining traffic characteristics of systems which are offered overflow multi-service traffic streams, generated both by finite and infinite source population. On the basis of author's earliest results [10–13]), the method for determining occupancy distribution in the group servicing multi-service overflow traffic will be presented. The proposed method is based on the appropriate modification of the Kaufman-Roberts recursion [19, 24] – elaborated for the full-availability group with Poisson traffic – and uses the idea of Fredericks & Hayward approximation.

In order to keep consistency of the considered problems, we start considerations from presentation of basic analytical dependencies for systems with single-rate overflow traffic in Section 2. In Section 3 it is presented the method for determining occupancy distribution in groups servicing multi-service overflow traffic. Section 4 includes the description of the method for determining parameters of the traffic overflowing from primary groups servicing multi-service PCT1 and PCT2 traffic streams. Comparison of analytical and simulation results of blocking probability in alternative groups servicing multi-service overflow traffic is performed in Section 5. Section 6 concludes the paper.

2. Modeling systems with overflow single-rate traffic

2.1. Overflow traffic parameters

The traffic that overflows from the direct group which is offered PCT1 traffic can be characterized with the help of the following two parameters: the mean value of overflow traffic R and its variance σ^2 (or the coefficient Z or the coefficient D). In order to evaluate analytically these parameters we will consider the following model: a full-availability group with the capacity of V Basic Bandwidth Units (the primary group) is offered traffic of the type PCT1 with the mean intensity A:

$$A = \frac{\lambda}{\mu}. \quad (3)$$

The next assumption is that the traffic that is not carried because of the occupancy of all the BBUs of the considered group overflows to a next full-availability group (the alternative group) with an unlimited number of BBUs. The values to be determined are: the average number of busy BBUs R in the alternative group (mean value of overflow traffic) and its variance σ^2 (variance of overflow traffic).

The process going on in the system presented in Figure 1, composed of two full-availability groups, is determined by the the two-dimensional discrete Markov chain: $\{\omega(t), \rho(t)\}$, where $\omega(t)$ is the number of busy BBUs in

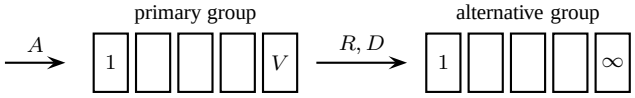


Figure 1. Model of a system with overflow traffic

the original group at the point of time t, whereas $\rho(t)$ is the number of busy BBUs in the alternative group at the point of time t. The state probabilities of the system under consideration are denoted with the symbols $[p_{\omega,\rho}]_{V,\infty}$ and are defined in the following way:

$$[p_{\omega,\rho}]_{V,\infty} = \lim_{t\to\infty} P\{\omega(t) = \omega, \rho(t) = \rho\}, \qquad (4)$$

where: $(0 \leqslant \omega \leqslant V)$ and $(0 \leqslant \rho \leqslant \infty)$. The probabilities $[p_{\omega,\rho}]_{V,\infty}$ can be determined on the basis of the system of state equations that, for the considered process, takes the following form:

$$\begin{gathered}
\dots \\
-(\lambda+\rho\mu)[p_{0,\rho}]_{V,\infty} + \mu[p_{1,\rho}]_{V,\infty} + \\
+(\rho+1)\mu[p_{0,\rho+1}]_{V,\infty} = 0 \dots \\
-(\lambda+\omega\mu+\rho\mu)[p_{\omega,\rho}]_{V,\infty} + \lambda[p_{\omega-1,\rho}]_{V,\infty} + \\
+(\omega+1)\mu[p_{\omega+1,\rho}]_{V,\infty} + (\rho+1)\mu[p_{\omega,\rho+1}]_{V,\infty} = 0 \\
\dots \\
-(\lambda+V\mu+\rho\mu)[p_{V,\rho}]_{V,\infty} + \lambda[p_{V-1,\rho}]_{V,\infty} + \\
+\lambda[p_{V,\rho-1}]_{V,\infty} + (\rho+1)\mu[p_{V,\rho+1}]_{V,\infty} = 0 \\
\dots \\
\sum_{\rho=0}^{\infty}\sum_{\omega=0}^{V}[p_{\omega,\rho}]_{V,\infty} = 1
\end{gathered} \qquad (5)$$

Once the system of equations (5) has been solved, it is possible to determine all essential properties of the system with traffic overflow. A determination of the parameters R and σ^2, related to the alternative group with unlimited capacity, can be, however, simplified as compared to the system (5). This possibility of simplification is connected with the fact that for a determination of parameters R and σ^2 the knowledge of all probabilities $[g_\rho]_\infty$ is not necessary, but it is sufficient to know only those probabilities $[g_\rho]_\infty$ that relate to the alternative group only, regardless the occupancy state of the primary group, i.e.:

$$[g_\rho]_\infty = \sum_{\omega=0}^{V}[p_{\omega,\rho}]_{V,\infty}. \qquad (6)$$

Knowing the occupancy $[g_\rho]_\infty$, it is possible to determine the parameters to be found, i.e. R and σ^2:

$$R = \sum_{\rho=0}^{V}\rho[g_\rho]_\infty, \quad \sigma^2 = \sum_{\rho=0}^{V}\rho^2[g_\rho]_\infty - R^2. \qquad (7)$$

Derivations of Equation (7) will be omitted here (they are to be found in, for example, [1, 4, 35]), by giving the final result derived by J. Riordan [35]:

$$R = AE_V(A), \qquad (8)$$

$$\sigma^2 = R[A/(V+1-A+R)+1-R]. \qquad (9)$$

In calculational practice, instead of the variance σ^2 the parameter D is often used. Hence, on the basis of Equation (2), (8) and (9) we obtain:

$$D = R[A/(V+1-A+R)-R]. \qquad (10)$$

Formula (8) is intuitively self-evident since it is only traffic lost in the original group that can be the offered traffic and, at the same time, be carried by the infinite alternative group. It should be noted that, quite predictably, for $V = 0$ (zero capacity of the original group), $R = \sigma^2 = A$, since all the PCT1 traffic is directed to the alternative group. Generally, for each value of the parameters A and V of the full-availability group, the parameters of overflow traffic R and σ^2, or R and D can be unequivocally determined.

In telecommunications networks, calls streams from several high-usage full-availability groups most frequently overflow to one alternative path. If we assume that PCT1 streams offered to high-usage primary groups are statistically independent, then the streams that overflow from these groups will also be independent. In such a case, the parameters of the total overflow traffic offered to the alternative path are determined by the following formulas [31]:

$$R = \sum_{s=0}^{v}R_s, \quad \sigma^2 = \sum_{s=0}^{v}\sigma_s^2, \quad D = \sum_{s=0}^{v}D_s, \qquad (11)$$

where: v – number of primary group, R_s – mean value of overflow traffic from s-th group, σ_s^2 – variance of overflow traffic from s-th group.

2.2. Method of equivalent random traffic

Analysing Formulas (8) and (10) we can notice that the parameters A and V determine unequivocally the parameters of the overflow traffic R and D of a given group. Consequently, these formulas can be used to solve a reverse problem, i.e. to determine unequivocally the parameters of the original group A and V on the basis of the parameters of the traffic that overflows from this group: R and D [31]. This conclusion has been applied to the ERT method (Equivalent

Random Traffic), which has been worked out independently by R. I. Wilkinson [35] and G. Bretschneider [4].

The ERT method consists in finding such an equivalent PCT1 traffic with the mean value A^*, that when offered to a fictitious equivalent group with the equivalent capacity of V^*, will cause an overflow of traffic with identical mean value and variance as the actual traffic offered to a given alternative group [31]. In this way, the traffic initially defined by the pairs of parameters: A_s and V_s (the alternative group usually services traffic overflowing from a few high-usage primary groups), will be described by one pair of parameters only (A^*, V^*).

The parameters A^* and V^* of the equivalent group can be determined on the basis of the obtained values R and D, solving the set of Riordan equations [35]:

$$R = A^* E_{V^*}(A^*), \tag{12}$$

$$D = R\left[A^*/\left(V^* + 1 - A^* + R\right) - R\right]. \tag{13}$$

Such equivalent traffic, determined by the pair of the parameters (A^*, V^*), requires $V^* + V_{\text{alt}}$ BBUs for servicing calls with assigned quality B. The required capacity of the alternative group can be obtained on the basis of Erlang-B formula, written in the following form:

$$E = B = E_{(V^* + V_{\text{alt}})}(A^*), \tag{14}$$

where E is the blocking probability, and B is the loss probability in the alternative group.

Summing up, the ERT method, presented graphically in Figure 2, can be written in the form of the following algorithm:

Algorithm 1 ERT Method

1. Determination of the mean value R_s and the parameter D_s of each of υ $(s = 1, \ldots, \upsilon)$ traffic streams that overflow to the alternative group (Equations (8) and (10));
2. Determination of the parameters of the total stream that overflows to the considered alternative group, assuming statistical independence of overflow streams (Equation (11));
3. Determination of the parameters A^* and V^* of the equivalent group on the basis of the obtained parameters R and D; these parameters can be determined by providing solution to the Riordan system of equations (Equation (13));
4. Determination of the required capacity of the alternative group for the assigned quality of service in the system equal to B (Equation (14)).

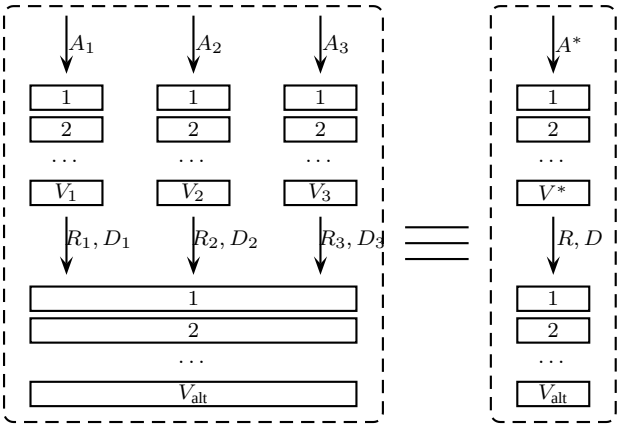


Figure 2. Graphical representation of the ERT method

The determination of the parameters of the equivalent group (A^*, V^*) is a complex issue and requires the application of complex, iterative computational programs [23, 31]. Therefore, to simplify the calculations, special nomograms have been developed [28] that present in graphic form dependencies between pairs of parameters (A^*, V^*) and (R, D). If, however, the above graphic dependencies are unavailable, then to determine the parameters (A^*, V^*) one can use the approximate solution of the system of equations (12) and (13), proposed by G. Rapp [22]:

$$A^* = \sigma^2 + 3\frac{\sigma^2}{R}\left(\frac{\sigma^2}{R} - 1\right), \tag{15}$$

$$V^* = A^* \frac{(R^2 + \sigma^2)}{R^2 + \sigma^2 - R} - R - 1. \tag{16}$$

It should be stressed that the determined values of parameters A^* and V^* obtained after the application of Rapp formulas are approximate, with the accuracy of calculations being the lowest within the area of low loss probability values [33]. With values of this probability lower than 1%, the approximation error can exceed 20%. Therefore, for $B < 0.01$ (which happens rarely in high-usage primary groups in real networks) it is more convenient to use the cited above nomograms [28]. A detailed analysis of the accuracy of this method has been worked out by J. M. Holtzmann and presented in [16] which shows the dependency between the error of loss probability, determined by the ERT method, and the number of BBUs of the alternative group V_{alt} and the overflow traffic parameters R and σ^2. On the basis of these dependencies it is possible to find that the error increases with the increase of the variance of overflow traffic σ^2, while it diminishes along with the increase in the number of BBUs in the high-usage primary group [31].

2.3. Fredericks-Hayward Method

Let us consider a full-availability group with the capacity of V BBUs which is offered overflow traffic with the mean value R and variance σ^2. The peakedness coefficient of the offered traffic is then:

$$Z = \frac{\sigma^2(R)}{R}. \tag{17}$$

Let us perform the following transformation, presented in Figure 3. Let us divide the group into Z identical full-availability groups (subsystems), each one with the capacity:

$$V_e = \frac{V}{Z}. \tag{18}$$

Each group is offered then traffic with the mean value:

$$R_e = \frac{R}{Z}. \tag{19}$$

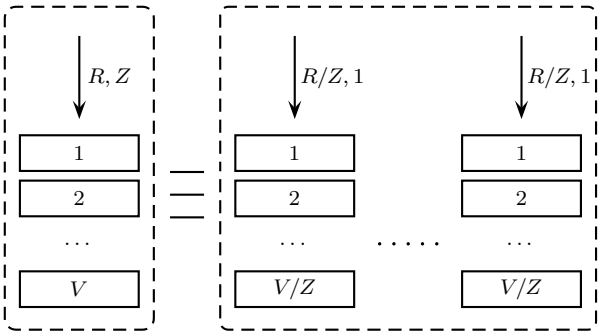


Figure 3. Transformation of the system (V, R, Z) into Z subsystems $(V/Z, R/Z, 1)$

Taking into consideration the property of variance, variance σ_e^2 can be determined in the following way:

$$\sigma_e^2 = \sigma^2\left(\frac{1}{Z}R\right) = \left(\frac{1}{Z}\right)^2 \sigma^2(R). \tag{20}$$

Now we can determine the peakedness coefficient of traffic offered to an individual subsystem. Taking into account (19) and (20), we get:

$$Z_e = \frac{\sigma_e^2}{R_e} = \frac{\sigma^2(R)}{RZ} = 1. \tag{21}$$

The peakedness coefficient equal to one means that traffic R_e is a PCT1 traffic. Thus, we have made a transformation of the full-availability group – described by the parameters (R, V, Z) – which is offered overflow traffic into Z subsystems (full-availability groups) – described by the parameters $(R/Z, V/Z, 1)$ – which is offered PCT1 traffic. Since all groups are identical, blocking probabilities in all groups will be also identical. In work [8] it is assumed that blocking probability in the group $(R/Z, V/Z, 1)$ will be the same as in the initial group (R, V, Z). Therefore, we can write:

$$E(R, V, Z) \approx E(R/Z, V/Z, 1) \approx E_{\frac{V}{Z}}\left(\frac{R}{Z}\right). \tag{22}$$

Formula (22) is a modified Erlang-B formula that takes into consideration non-Poisson nature of the calls stream offered to the group. In teletraffic theory, this formula is called Fredericks-Hayward formula.

The presented reasoning for Equation (22) assumes mutual independence of traffic offered to the subsystems. In real world, a distribution of the traffic stream into several identical streams without an application of an appropriate call assignment mechanism is not possible. The introduction of such a mechanism is, however, tantamount to the introduction of mutual correlation between the streams, which, in turn, can be interpreted as a lack of independence of the traffic streams offered to the subsystems. This phenomenon makes the formula (22) an approximated formula. It should be stressed, though that it is characterized by high accuracy [8, 18].

Equation (22) forms the basis for Fredericks-Hayward method [8] and can be described in the form of the following algorithm:

Algorithm 2 Fredericks-Hayward Algorithm

1. Determination of the mean value and the variance of each of υ traffic streams that overflows to an alternative group based on the formulas (8) and (9);
2. Determination of the parameters of the total overflow traffic (Equation (11)) offered to the alternative group and the peakedness coefficient (Equation (1)) of the traffic, assuming statistical independence of the overflow streams;
3. Determination of the number of BBUs of the alternative group (with the assigned quality of service, equal to B) on the basis of Fredericks-Hayward formula (22).

Fredericks-Hayward method is far more simple than the ERT method since it requires only calculations based on Erlang-B formula. The formula is used in two steps of the algorithm – with the determination of mean value of traffic that overflows to the alternative group (Formula (8)) and, in the form of Fredericks-Hayward formula, with the determination of the capacity of the alternative group (Formula (22)).

3. Modeling of full-availability groups with multi-service overflow traffic

3.1. Basic assumptions

Let us consider first a fragment of the network shown in Figure 4, servicing multi-service PCT1 traffic streams. It is assumed that each of primary groups is offered only one call class. The adopted assumption is to facilitate the understanding of the introduced analytical dependencies. Systems in which primary groups service many classes of traffic will be presented in Section 4.

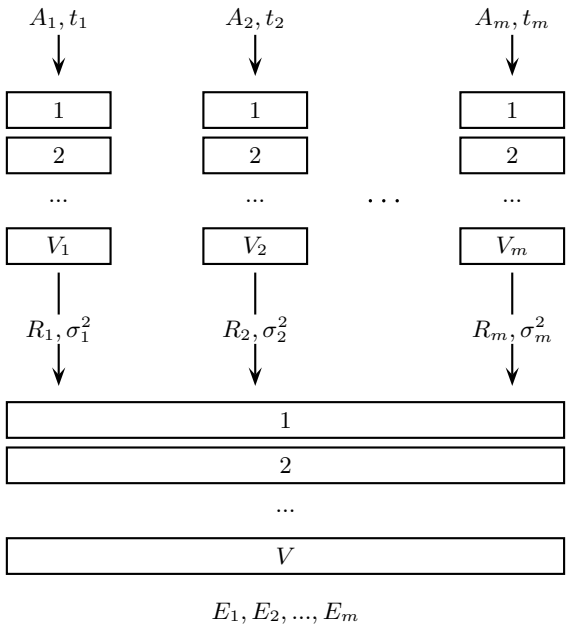


Figure 4. A fragment of the network with overflow traffic

There are $m = m_I$ high-usage primary groups in the considered system. The group designated by number i has the capacity equal to V_i BBUs. Each of the groups is offered a different calls stream characterized by the traffic intensity A_i. The calls of class i demand t_i BBUs to set up a connection.

3.2. Parameters of overflow traffic

As the result of occupying successive BBUs in primary groups, a situation ensues in which the groups get blocked and traffic overflows to an alternative group with the capacity V_{alt}. Blocking coefficients in primary groups can be calculated with the help of the Erlang-B formula. One has to take into consideration, however, that one call of class i occupies simultaneously t_i BBUs [10, 11].

Therefore, from the point of view of the Erlang model, it is tantamount to t_i-fold decrease of the capacity of the group with the real capacity of V_i BBUs. What it means is that before the substitution to Erlang-B formula, the group capacity should be divided by the number of BBUs demanded to set up a connection of a given class. With the case of non-integral values V_i/t_i, calculations of blocking probability can be performed using the interpolation method or the approximation of Erlang loss formula in the following form [32]:

$$E_{N+\delta} = \frac{AE_{N+\delta-1}(A)}{N+\delta+AE_{N+\delta-1}(A)}, \tag{23}$$

where $N+\delta$ is non-integral value of group's capacity (N is an integer part and δ is a fraction). To start the calculation process we need to use an approximate formula:

$$E_\delta \approx \frac{(2-\delta)A + A^2}{\delta + 2A + A^2}. \tag{24}$$

Another way to obtain the same values of blocking coefficients is to apply the Kaufman-Roberts formulas [19, 24]:

$$n\,[P_n]_V = \sum_{i=1}^{m} A_i t_i\,[P_{n-t_i}]_V, \tag{25}$$

$$B_i = E_i = \sum_{n=V-t_i+1}^{V} [P_n]_V, \tag{26}$$

where $[P_n]_V$ is the occupancy distribution, i.e. the probability of n BBUs being busy in the system. Equations (25) and (26) will take into consideration the group with the capacity of V_i which is offered one calls stream with Poisson distribution formed by the calls that demand t_i BBUs to set up a connection [10, 11].

Knowing the blocking coefficients in primary groups we are in position to calculate the parameters of overflow traffic of each of the classes, i.e the mean value R_i and the variance σ_i^2. For this purpose, the Riordan formulas (8) and (9) are used. Then, on the basis of the obtained parameters, we determine the unevenness of individual calls streams of overflow traffic by calculating the values of peakedness coefficients $Z_i = \sigma_i^2/R_i$.

It should be emphasised that the possibility of direct application of Riordan formulas, elaborated for systems with single-rate traffic, results from the assumption that each primary group is offered only one traffic class [10, 11]. In the case when all groups serve calls of several traffic classes, the determination of variance of overflow traffic becomes a complex problem [2, 3], despite the value of traffic intensity can be simply obtained on the Kaufman-Roberts formulas (25) and (26). An approximate method of elaboration of the variance of the traffic overflowing from primary group servicing mixture of multi-service traffic will be presented in Section 4.

3.3. Modeling overflow traffic in systems with infinite number of traffic sources

Calls lost in primary groups are offered to an alternative group and, successively, begin to occupy its resources. Thus, the group services m call classes. In order to determine blocking coefficients in such a group we apply the analogy to Hayword method, described in Section 2.3. Let us remind that the method was designed to determine the blocking coefficient in the group with the capacity of V BBUs with single-service traffic which was offered overflow traffic stream with the mean value R, additionally characterized by the peakedness Z. In this method the Fredericks-Hayword equation is used, i.e. the Erlang-B formula with appropriately modified parameters A and V. In the case of a group with multi-service traffic, we will apply the identical modification to Kaufman-Roberts formulas:

$$E_{\text{alt},1}, E_{\text{alt},2}, \ldots, E_{\text{alt},m} = \\ = KR\left(\frac{R_1}{Z_1}, \frac{R_2}{Z_2}, \ldots, \frac{R_m}{Z_m}; t_1, t_2, \ldots, t_m; \frac{V_{\text{alt}}}{Z}\right), \quad (27)$$

where $KR(\cdot)$ denotes the algorithm for determining blocking coefficients of calls of particular classes E_1, E_2, ..., E_M, on the basis of the Kaufman-Roberts equations (25) and (26) that take on the following form [10, 11]:

$$n\left[P_n\right]_{V_{\text{alt}}/Z} = \sum_{i=1}^{m} \frac{R_i}{Z_i} \cdot t_i \left[P_{n-t_i}\right]_{V_{\text{alt}}/Z}, \quad (28)$$

$$B_{\text{alt},i} = E_{\text{alt},i} = \sum_{n=\frac{V}{Z}-t_i+1}^{\frac{V_{\text{alt}}}{Z}} \left[P_n\right]_{V_{\text{alt}}/Z}. \quad (29)$$

The peakedness coefficient acts a normalization function. By dividing the mean values of overflow traffics of particular call classes by the corresponding values of the coefficients Z_i, we perform a transformation of the uneven overflow traffic stream into the Erlang stream. Similarly as in the dependence (22), we also divide the capacity of the alternative group V by the value of the peakedness coefficient. Let us notice that the capacity of the alternative group in the formulas (28) and (29) is divided by the so-called overall peakedness coefficient Z. The problem of definition of this coefficient, for m calls classes, where each can have individual value of the peakedness Z_i, was taken in [10]. According to these considerations, the relevant parameter will be approximated by the weighted mean of the coefficients Z_i of particular calls streams:

$$Z = \sum_{i=1}^{m} Z_i k_i, \quad (30)$$

where

$$k_i = \frac{R_i t_i}{\sum_{l=1}^{m} R_l t_l} \quad (31)$$

It is adopted in Equation (30) that the contribution of peakedness Z_i of a stream of class i in the overall peakedness coefficient Z is directly proportional to the value of traffic offered to the alternative group by class i calls. The plausibility of this assumption has been proved by simulation studies [13].

The formulas (28) and (29) are a generalization of the Kaufman-Roberts formulas for all kinds of groups servicing multi-service traffic, both non-Poisson calls streams (overflow traffic) and Poisson calls streams. For the Poisson distribution, the value of the peakedness is equal to one and then the formulas (28) and (29) will take on the form of the basic Kaufman-Roberts formulas (25) and (26).

3.4. Modeling of overflow traffic in systems with finite number of traffic sources

In this section it is presented an analytical method for determining the mean value and the variance in systems with multi-service traffic overflowing from primary groups servicing multi-service PCT2 traffic streams [12]. The presented method is based on the method elaborated in [5] for the networks servicing single-rate traffic. The basis of this method is the application of ERT method to convert the traffic stream generated by the finite population of sources (PCT2 traffic stream) to the equivalent traffic stream generated with the assumption of the infinite population of sources (PCT1 traffic streams) [29].

Let us consider a group with the capacity of V_j BBUs servicing a finite number of sources for each traffic class. Let N_j be a number of sources of class j requiring t_j BBUs to be serviced. The input calls stream of class j is built by the superposition of N_j two-state traffic sources which can alternate between the active (busy) state ON (the source requires t_j BBUs) and the inactive state OFF (the source is idle). When a source is busy, its call intensity is zero. Thus the arrival process is state-dependent. The class j arrival rate in the state of n BBUs being busy can be expressed by the following formula:

$$\lambda_j(n) = (N_j - n_j(n))\Lambda_j, \quad (32)$$

where $n_j(n)$ is a number of class j calls being serviced in state n (state of n BBUs being busy) and Λ_j is the mean arrival rate generated by an idle source of class j. In the considered model we assume additionally that the holding time for calls of particular classes has an exponential distribution. Thus, the class j traffic α_j offered by an idle source is equal to:

$$\alpha_i = \frac{\Lambda_i}{\mu_i}, \quad (33)$$

where $1/\mu_j$ is the mean holding (service) time of class j calls.

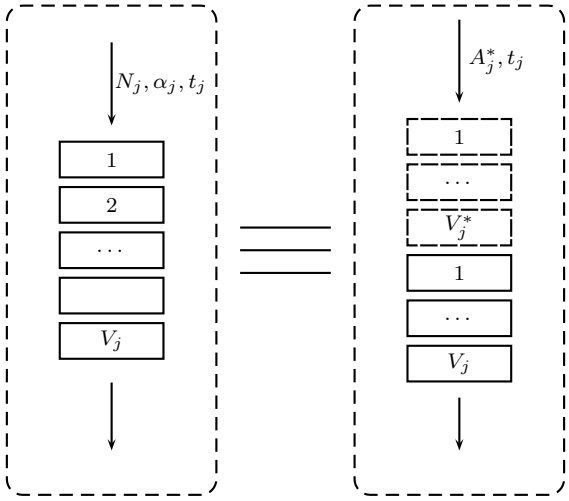


Figure 5. The idea of conversion of systems PCT2 to PCT1

Let us additionally assume, that $N_j > V_j$. Based on the results presented in [5] and [29] we can determine the mean value $R_{\text{PCT2},j}$, the variance $\sigma^2_{\text{PCT2},j}$ and the coefficient $D_{\text{PCT2},j}$ of the number of busy BBUs in considered group:

$$R_{\text{PCT2},j} = \frac{N_j \alpha_j}{1+\alpha_j}, \tag{34}$$

$$\sigma^2_{\text{PCT2},j} = \frac{N_j \alpha_j}{(1+\alpha_j)^2}, \tag{35}$$

$$D_{\text{PCT2},j} = \sigma^2_{\text{PCT2},j} - R_{\text{PCT2},j} = -N_j \frac{\alpha_j}{(1+\alpha_j)^2}. \tag{36}$$

The traffic described by Equations (34), (35) and (36) can be treated as an equivalent PCT1 stream with intensity A_j^* overflowing on the equivalent group with the capacity equal to V_j^* BBUs. The idea of this conversion is presented in Figure 5. We call A_j^* and V_j^* fictitious, and their values can be obtained as the solution of a set of Riordan formulas – according to ERT method (page 4):

$$R_{\text{PCT2},j} = A_j^* E_{V_j^*}\left(A_j^*\right), \tag{37}$$

$$D_{\text{PCT2},j} = \\ = R_{\text{PCT2},j}\left[\frac{A_j^*}{V_i^* + 1 - A_j^* + R_{\text{PCT2},j}} - R_{\text{PCT2},j}\right]. \tag{38}$$

The above equations have a solution if we use Erlang formula for negative values of link capacity [5, 32]. It is possible to obtain the occupancy distribution for $V < 0$ on the basis of the following recurrent formula:

$$E_{V-1}(A) = \frac{V E_V(A)}{A(1 - E_V(A))}, \tag{39}$$

where the initial solution, for $V = -1$, we can get on the basis of the following equation:

$$E_{-1}(A) = [-Ei(-A)A\mathrm{e}^A]^{-1}, \tag{40}$$

in which function $Ei(A)$ is defined as follows:

$$Ei(x) = -\int_x^\infty (At+A)^{-1}\mathrm{e}^{At+A}\mathrm{d}(At+A). \tag{41}$$

It is also possible to approximate the function (40) by the the following polynomial [29]:

$$E_{-1}(A) \approx \frac{b_0 + b_1 A + b_2 A^2 + b_3 A^3 + b_4 A^4}{a_0 + a_1 A + a_2 A^2 + a_3 A^3 + a_4 A^4}, \tag{42}$$

where:

$$\begin{aligned} a_0 &= 0,2677737343, & b_0 &= 3,9584969228, \\ a_1 &= 8,6347608925, & b_1 &= 21,0996530827, \\ a_2 &= 18,0590169730, & b_2 &= 25,6329561486, \\ a_3 &= 8,5733287401, & b_3 &= 9,5733223454, \\ a_4 &= 1, & b_4 &= 1. \end{aligned}$$

Having at our disposal the values of fictitious traffic A_j^* and the equivalent group capacity V_j^*, we can calculate on the basis of (8) and (9) the parameters of the traffic overflowing from the primary group servicing PCT2 traffic streams, i.e. the variance σ_j^2 and the mean value R_j:

$$R_j = A_j^* E_{(V_j/t_j)+V_j^*}(A_j^*), \tag{43}$$

$$\sigma_j^2 = R_j\left[A_j^*/(V_j/t_j + V_j^* + 1 - A_j^* + R_j) + 1 - R_j\right]. \tag{44}$$

Let us notice that in Equation (43) and (44) the real link capacity V_j is divided by t_j because in the process of obtaining the capacity of fictitious link V_j^* we consider single-rate traffic (calls of each traffic class can demand only one BBU).

Having at disposal the parameters of traffic overflowing from primary groups, we can determine the occupancy distribution in the alternative group on the basis of the modified Kaufman-Roberts recursion, described in Section 3.3.

4. Modeling of systems with overflow multi-service traffic

In the previous section we dealt with the determination of the occupancy distribution in the alternative full-availability groups in systems in which primary groups serviced only one calls stream. This was purely theoretical case and its main purpose was to facilitate understanding of the introduced analytical dependencies. In real systems,

primary groups carry multi-service traffic that is composed of several classes of calls.

The assumption that has been used so far allowed us to determine the variance of traffic that overflows from primary groups in a simple way through the application of Riordan formulas. With the case when the group carries multi-service traffic, direct application of the Riordan formulas is not possible. In this section we will present an approximate method for determining variances of different traffic streams that overflow from groups servicing multi-service traffic.

Let us consider the fragment of a multi-service network shown in Figure 6. The system is composed of υ primary high-usage groups. Each of the group $s = 1, \ldots, \upsilon$ is offered $m_{I,s}$ PCT1 traffic streams and $m_{J,s}$ PCT2 traffic streams ($m_s = m_{I,s} + m_{J,s}$). Calls of class c demand t_c BBUs to set up a connection[3]. The intensity of PCT1 traffic stream of class i offered to the group s is $A_{i,s}$. The intensity of PCT2 traffic offered by a single idle source of class j in the group s is $\alpha_{j,s}$, while the intensity of traffic $A_{j,s}(n)$ offered by all idle PCT2 sources of class j in the group s depends on the occupancy state n of the group in the following way:

$$A_{j,s}(n) = (N_{j,s} - n_{j,s}(n))\alpha_{j,s}, \tag{45}$$

where $n_{j,s}(n)$ is the number of in-service sources of class j in the state of n BBUs being busy.

The traffic of particular classes, which is blocked in primary groups overflows to the alternative group. The blocking coefficient for calls of class i (PCT1) in the direct group s ($E_{i,s}$) can be determined on the basis of the Kaufman-Roberts formulas (25) and (26).

In the case of the full-availability group with PCT2 traffic stream, the Kaufman-Roberts recursion (25) can be rewritten in the form that includes characteristics of Engset traffic streams, namely:

$$n[P_n]_{V_s} = \sum_{j=1}^{m_J,s} A_{j,s}(n - t_j)t_j[P_{n-t_k}]_{V_s}. \tag{46}$$

According to the considerations presented in [9], the parameter $n_{j,s}(n)$ in Equation (45) can be approximated by the so-called *reverse transition rate* and can be calculated on the basis of the local equations of equilibrium [19, 30]:

$$n_{j,s}(n) = \begin{cases} A_{j,s}(n - t_j)[P_{n-t_j}]_{V_s} / [P_n]_{V_s} & \text{for } n \leq V_s, \\ 0 & \text{for } n > V_s. \end{cases} \tag{47}$$

The reverse transition rate determines the average number of class j calls serviced in the state n. Let us note that to determine the parameter $n_{j,s}(n)$ the knowledge of the occupancy distribution $[P_n]_{V_s}$, is necessary. In order to determine the distribution $[P_n]_{V_s}$ in turn, it is necessary to know the value $n_{j,s}(n)$. Equations (47) and (46) form then a set of confounding equations that can be solved with the application of iterative methods. In line with [9], in the first iteration we assume that the parameters $\forall_{j \in m_j} \forall_{0 \leq n \leq V}\ n_{j,s}^{(0)}(n) = 0$. The adopted assumption means that the Engset streams – in the first iteration – can be treated as an equivalent Erlang streams generating the offered traffic with the intensity:

$$A_{j,s}(n) = A_{j,s} = N_{j,s}\alpha_j, \tag{48}$$

which is equal in value to the traffic offered by all free sources of class j Engset stream. The state probabilities, obtained on the basis of Eq. (46), constitute the input data for the next iteration l, where the parameters $n_{j,s}^{(l)}(n)$ and subsequently $A_j(n)$ are designated. The iterative process ends when the assumed accuracy ϵ is obtained:

$$\forall_{j \in \langle 1, m_J \rangle} \forall_{n \in \langle 0, V \rangle} \left(\left| \frac{n_{j,s}^{(l-1)}(n) - n_{j,s}^{(l)}(n)}{n_{j,s}^{(l)}(n)} \right| \leq \epsilon \right). \tag{49}$$

The obtained occupancy distribution $[P_n]_{V_s}$ in the group with Engset traffic streams allows us to calculate the blocking probability $E_{j,s}$ on the basis of Equation (26).

Knowing blocking probabilities for PCT1 and PCT2 streams we are in position to determine the mean value

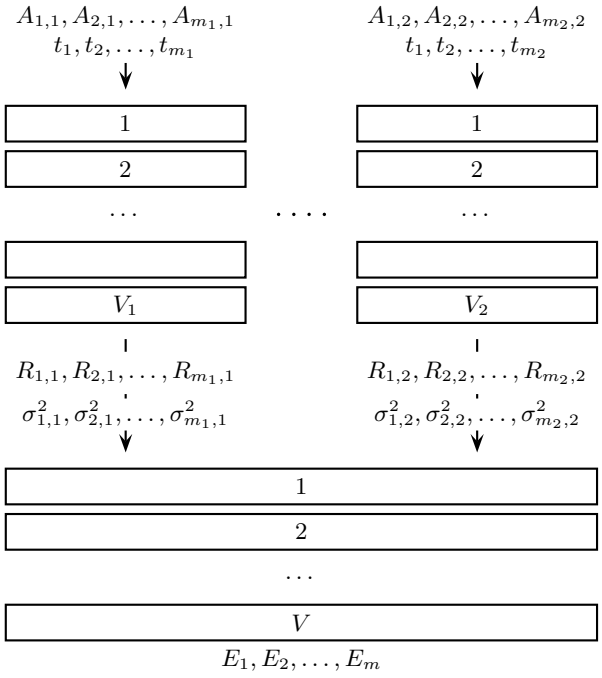


Figure 6. A fragment of telecommunications network with overflow multi-service traffic

[3]In the paper it is assumed that the letter "i" denotes a Poisson (Erlang) traffic class, the letter "j" – a Binomial (Engset) traffic class, and the letter "c" – an arbitrary traffic class, ($c = i|j$)

of the intensity of class c traffic that overflows from the group s:

$$R_{c,s} = A_{c,s}E_{c,s}. \tag{50}$$

To characterize overflow traffic fully it is necessary to determine the variance of each of calls streams. This parameter will be determined in an approximate way by carrying out a decomposition of each of the real groups into m_s fictitious component groups with the capacities $V_{c,s}$. Each fictitious group will be servicing exclusively calls of one class, which will make it possible to apply the Riordan formulas to determine the variance $\sigma^2_{c,s}$ of the traffic of class c that overflows from the group s. Let us determine then the capacities of the fictitious groups. For this purpose we first determine the carried traffic of class c in the group s:

$$Y_{c,s} = A_{c,s}(1 - E_{c,s}). \tag{51}$$

According to the definition, the value $Y_{c,s}$ defines the average number of calls of class c serviced in the group s. Therefore, the mean value of the intensity of class c traffic, expressed in BBUs, will be equal to $Y_{c,s}t_c$. The capacity of a fictitious component group $V_{c,s}$ will be defined as this part of the real group V_s which is not occupied by calls of the remaining classes (different from class c). Thus, we get [10–12]:

$$V_{c,s} = V_s - \sum_{l=1;l\neq c}^{m_{I,s}+m_{J,s}} Y_{l,s}t_l, \tag{52}$$

where V_s is the capacity of the primary group and the sum on the right side of Equation (52) determines the number of BBUs occupied by the calls of the remaining classes. The proposed decomposition allows us to use the method proposed in Section 3.4, to convert the system with PCT2 traffic streams to the equivalent PCT1 traffic streams.

Having all the parameters at our disposal for PCT1, i.e. $R_{i,s}$, $A_{i,s}$, $V_{i,s}$ and PCT2, i.e. $A^*_{j,s}$, $R_{j,s}$, $V^*_{j,s}$, $V_{j,s}$ we can – on the basis of the Riordan formula – determine the variance $\sigma^2_{i,j}$ for individual calls streams that overflow to the alternative group:

$$\sigma^2_{i,s} = R_{i,s}\left[\frac{A_{i,s}}{V_{i,s}/t_i + 1 - A_{i,s} + R_{i,s}} + 1 - R_{i,s}\right], \tag{53}$$

$$\sigma^2_{j,s} = R_{j,s}\left[\frac{A^*_{j,s}}{V_{j,s}/t_j + V^*_{j,s} + 1 - A^*_{j,s} + R_{j,s}} + 1 - R_{j,s}\right], \tag{54}$$

where the quotient $V_{c,s}/t_c$ normalizes the system to a single-service case. Such an operation is necessary since the Riordan formulas in their basic form are designed for determining overflow traffic parameters in single-service systems.

Since individual calls streams offered to the system are statistically independent, then the parameters of the total traffic of class c offered to the alternative group will be equal to:

$$R_c = \sum_{s=1}^{\upsilon} R_{c,s}, \quad \sigma^2_c = \sum_{s=1}^{\upsilon} \sigma^2_{c,s}. \tag{55}$$

At this point we have all the parameters that characterize m calls streams offered to the alternative group. Having at our disposal the dependencies (55), we can determine the occupancy distribution and the blocking probability in the system with overflow multi-service traffic shown in Figure 6. In order to do that, we can apply the formulas (28) and (29), where the overall coefficient Z is determined according to Equation (30).

Summing up our considerations, we can present the process of determining occupancy distribution in the alternative group of hierarchically organised networks with overflow traffic in the form of the Algorithm Overflow-MKRR.

Algorithm 3 Algorithm Overflow-MKRR

1. Determination of blocking probability of class $c = 1, \ldots, m$ calls stream in each of primary groups υ;
2. Determination of the mean value $R_{c,s}$ of class c traffic overflowing from the primary group $s = 1, \ldots, \upsilon$;
3. Decomposition of the primary group s (with the capacity of V_s BBUs), servicing m_s traffic classes, on the m_s groups where each has the capacity of $V_{c,s}$ BBUs (Equation (52));
4. Conversion of PCT2 traffic stream to the equivalent PCT1 traffic stream (Section 3.4);
5. Determination of the variance $\sigma^2_{c,s}$ of class c traffic stream overflowing from the primary group $V_{c,s}$ to the alternative group V_{alt} (Equation (53) and (54));
6. Determination of the parameters of class c overflow traffic offered to the alternative group (Equation (55));
7. Determination of the overall coefficient Z (Equation (30));
8. Determination of the occupancy distribution in the alternative group (Equation (28));
9. Determination of blocking probability for all traffic classes in the alternative group (Equation (29)).

5. Numerical examples

The presented methods for determining the parameters of overflow traffic, the occupancy distribution and the blocking probability in systems with overflow multi-service

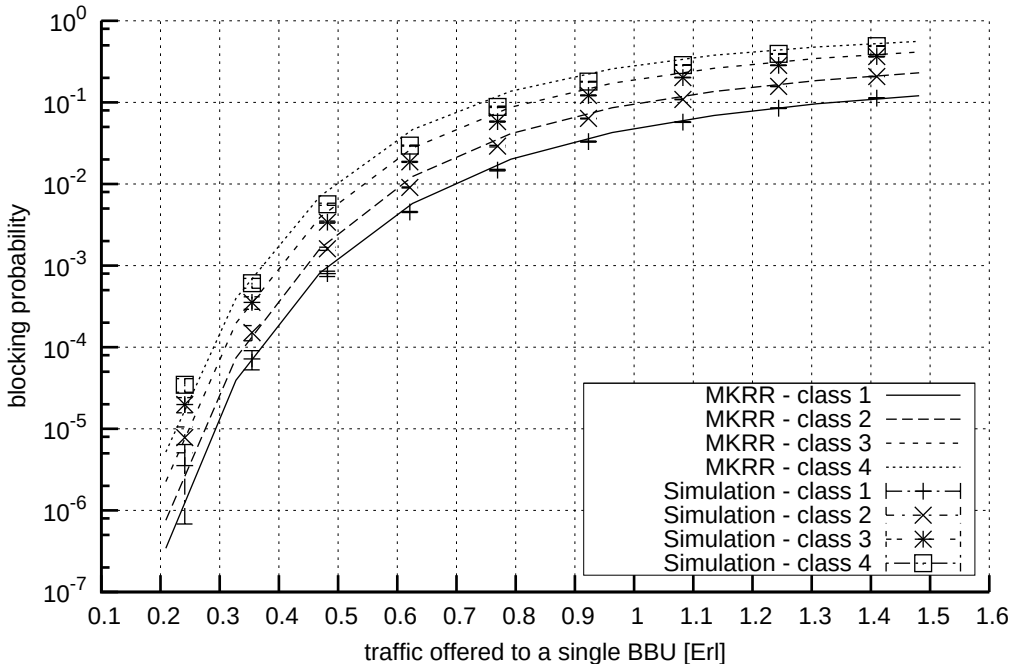


Figure 7. Blocking probability in the alternative group with overflow multi-service traffic with capacity equal to $V = 200$ BBUs; first and second primary groups: $V_1 = V_2 = 60$ BBUs, $t_1 = 2$ BBUs, $t_2 = 4$ BBUs, $t_3 = 8$ BBUs $A_{1,1}t_1 : A_{2,1}t_2 : A_{3,1}t_3 = 1 : 1 : 1$, $A_{1,2}t_1 : A_{2,2}t_2 : A_{3,2}t_3 = 1 : 1 : 1$; third and fourth primary groups: $V_3 = V_4 = 100$ BBUs, $t_1 = 2$ BBUs, $t_2 = 4$ BBUs, $t_3 = 8$ BBUs, $t_4 = 12$ BBUs, $A_{1,3}t_1 : A_{2,3}t_2 : A_{3,3}t_3 : A_{4,2}t_4 = 1 : 1 : 1 : 1$, $A_{1,4}t_1 : A_{2,4}t_2 : A_{3,4}t_3 : A_{4,4}t_4 = 1 : 1 : 1 : 1$; fifth primary group: $V_5 = 40$ BBUs, $t_2 = 4$ BBUs

traffic are the approximate methods. To determine the precision of the proposed solution, results of analytical calculations were compared with the simulation data. The research was carried out for two networks. The first network was composed of five primary groups servicing multi-service PCT1 (Erlang) traffic streams and one alternative group (with the capacity of 200 BBUs) servicing the traffic overflowing from the primary groups. The second network was composed of three primary groups servicing multi-service PCT2 (Engset) traffic streams and one alternative group (with the capacity of 100 BBUs) servicing the overflowed traffic.

The parameters of the offered traffic and the capacities of individual groups are given in the captions to Figures 7 and 8 presenting the obtained blocking probability results in the alternative group – both analytical and simulation results. The value of the blocking probability is expressed in the function of normalized traffic a offered to a single BBU of the alternative group:

$$a = \frac{\sum_{c=1}^{m} R_c t_c}{V_{\text{alt}}}. \tag{56}$$

It was assumed that there was equal the normalized traffic u offered per single BBU in each of υ direct groups:

$$\forall_{1 \le s \le \upsilon}\ u = \sum_{c=1}^{m} \frac{A_{c,s} t_c}{V_s}. \tag{57}$$

The simulation results are shown in Figures 7 and 8 in the form of appropriately denoted points with 95-percent confidence interval, calculated according to the t-Student distribution for 5 series, with 1000000 calls of each class.

On the basis of the obtained blocking probability results in the considered systems we can state that the proposed calculational method for overflow traffic parameters combined with the modification of Kaufman-Roberts formula (28) provides high accuracy of calculations.

6. Conclusion

An analytical method for determining the occupancy distribution and blocking probability in groups of telecommunication networks servicing overflow multi-service traffic is presented in the article. The presented method is based on a modification of the Kaufman-Roberts formula,

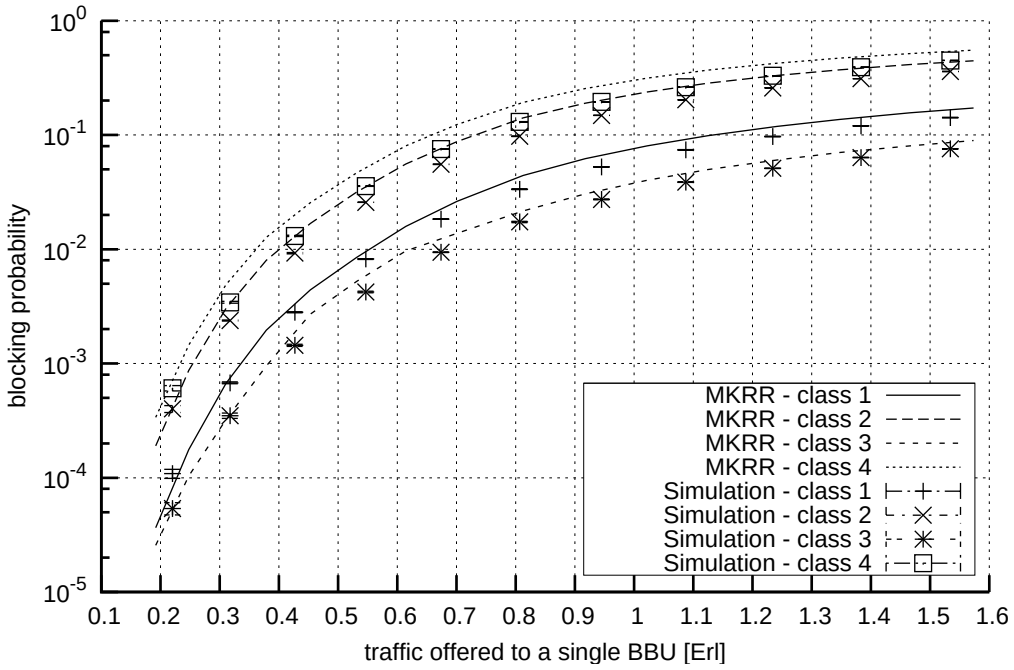


Figure 8. Blocking probability in the alternative group with overflow multi-service traffic with capacity equal to $V = 100$ BBUs; first primary group: $V_1 = 60$ BBUs, $t_2 = 2$ BBUs, $S_2 = 80$, $t_3 = 6$ BBUs, $S_3 = 60$, $A_{2,1}t_2 : A_{3,1}t_3 = 1 : 1$; second primary groups: $V_2 = 80$ BBUs, $t_1 = 1$ BBUs, $S_1 = 100$, $t_4 = 8$ BBUs, $S_4 = 60$, $A_{1,2}t_1 : A_{4,2}t_2 = 1 : 1$; third primary group: $V_3 = 100$ BBUs, $t_1 = 4$ BBUs, $S_1 = 100$, $t_3 = 6$ BBUs, $S_3 = 60$, $t_4 = 8$ BBUs, $S_4 = 60$

which involves an introduction of the peakedness coefficient Z that characterizes the unevenness of the overflow calls stream. Additionally, an analytical method for determining the occupancy distribution and blocking probability in groups of telecommunication networks servicing overflow multi-service traffic with a finite as well as infinite number of traffic sources is presented in the article. The presented method is based on conversion of traffic streams, generated by finite source population, to the traffic streams, generated by infinite source population. The accuracy of the proposed analytical method is verified by the presented simulation data.

References

[1] H. Akimuru and K. Kawashima. *Teletraffic: Theory and Application*. Springer, Berlin–Heidelberg–New York, 1993.

[2] A. Brandt and M. Brandt. Approximation for overflow moments of a multiservice link with trunk reservation. *Journal of Performance Evaluation*, 43(4):259–268, 2001.

[3] A. Brandt and M. Brandt. On the moments of the overflow and freed carried traffic for the GI/M/C/0 system. *Methodology and Computing in Applied Probability*, 2002(4):69–82, 2002.

[4] G. Bretschneider. Die Berechnung von Leitungsgruppen für berflieSSenden Verkehr in Fernsprechwählanlagen. *Nachrichtentechnische Zeitung (NTZ)*, (11):533–540, 1956.

[5] G. Bretschneider. Extension of the equivalent random method to smooth traffics. In *Proceedings of 7th International Teletraffic Congress*, Stockhholm, 1973.

[6] S.-P. Chung and J.-C. Lee. Performance analysis and overflowed traffic characterization in multiservice hierarchical wireless networks. *IEEE Transactions on Wireless Communications*, 4(3):904–918, May 2005.

[7] L. Delbrouck. On the steady-state distribution in a service facility carrying mixtures of traffic with different peakedness factors and capacity requirements. *IEEE Transactions on Communications*, 31(11):1209–1211, 1983.

[8] A. Fredericks. Congestion in blocking systems — a simple approximation technique. *Bell System Technical Journal*, 59(6):805–827, 1980.

[9] M. Głąbowski. Modelling of state-dependent multi-rate systems carrying BPP traffic. *Annales des Télécommunications*, 63(7-8):393–407, Aug. 2008.

[10] M. Głąbowski, K. Kubasik, and M. Stasiak. Modeling of systems with overflow multi-rate traffic. In *Proceedings of Third Advanced International Conference on Telecommunications – AICT 2008*, Morne, may 2007. best paper award.

[11] M. Głąbowski, K. Kubasik, and M. Stasiak. Modeling of systems with overflow multi-rate traffic. *Telecommunication Systems*, 37(1–3):85–96, Mar. 2008.

[12] M. Głąbowski, K. Kubasik, and M. Stasiak. Modelling of systems with overflow multi-rate traffic and finite number of traffic sources. In *Proceedings of 6th International Symposium on Communication Systems, Networks and Digital Signal Processing 2008*, pages 196–199, Graz, July 2008.

[13] M. Głąbowski, D. Mikołajczak, and M. Stasiak. Multi-rate systems with overflow traffic. Technical Report ZSTI 01/2005, Institute of Electronics and Telecommunications, Poznan University of Technology, Poznań, 2005.

[14] U. Herzog. Die exakte berechnung des streuwertes von Überlaufverkehr hinter koppelanordnungen beliebiger stufenzahl mit vollkommener bzw. unvollkommener erreichbarkeit. *AEÜ*, 20(3), 1966.

[15] U. Herzog and A. Lotze. Das RDA-Verfahren, ein streuwertverfahren für unvollkommene bündel. *Nachrichtentechnische Zeitung (NTZ)*, (11), 1966.

[16] J. Holtzmann. The accuracy of the equivalent random method with renewal inputs. In *Proceedings of 7th International Teletraffic Congress*, Stockholm, 1973.

[17] L.-R. Hu and S. S. Rappaport. Personal communication systems using multiple hierarchical cellular overlays. *IEEE Journal on Selected Areas in Communications*, 13(2):406–415, 1995.

[18] V. Iversen, editor. *Teletraffic Engineering Handbook*. ITU-D, Study Group 2, Question 16/2, Geneva, Dec. 2003.

[19] J. Kaufman. Blocking in a shared resource environment. *IEEE Transactions on Communications*, 29(10):1474–1481, 1981.

[20] J. S. Kaufman and K. M. Rege. Blocking in a shared resource environment with batched poisson arrival processes. *Journal of Performance Evaluation*, 24(4):249–263, 1996.

[21] X. Lagrange and P. Godlewski. Performance of a hierarchical cellular network with mobility-dependent handover strategies. In *Proceedings of IEEE Vehicular Technology Conference*, volume 3, pages 1868–1872, 1996.

[22] Y. Rapp. Planning of junction network in a multi-exchange area. In *Proceedings of 4th International Teletraffic Congress*, page 4, London, 1964.

[23] F. I. D. Rios and K. W. Ott. Computation of urban routing by computer. *Journal of the IEE*, 2, 1968.

[24] J. Roberts. A service system with heterogeneous user requirements — application to multi-service telecommunications systems. In G. Pujolle, editor, *Proceedings of Performance of Data Communications Systems and their Applications*, pages 423–431, Amsterdam, 1981. North Holland.

[25] R. Schehrer. On the exact calculation of overflow systems. In *Proceedings of Sixth International Teletraffic Congress*, pages 147/1–147/8, Munich, Sept. 1970.

[26] R. Schehrer. On the calculation of overflow systems with a finite number of sources and full availiable groups. *IEEE Transactions on Communications*, 26(1):75–82, Jan. 1978.

[27] J. F. Shortle. An equivalent random method with hyper-exponential service. *Journal of Performance Evaluation*, 57(3):409–422, 2004.

[28] SIEMENS. Telephone traffic theory tables and charts part 1. Technical report, Siemens, 1970.

[29] M. Šneps. *Sistemy raspredeleniâ informacii*. Metody rasčёta. Radio i Swâz', Moskva, 1979.

[30] M. Stasiak and M. Głąbowski. A simple approximation of the link model with reservation by a one-dimensional Markov chain. *Journal of Performance Evaluation*, 41(2–3):195–208, July 2000.

[31] M. Stasiak, S. Hanczewski, M. Głąbowski, and P. Zwierzykowski. *Fundamentals of Teletraffic Engineering and Networks Dimensioning*. Poznan University, Poznan, Poland, 2009. in Polish.

[32] R. Syski. Introduction to congestion theory in telephone systems. *Studies in Telecommunication, North Holland*, 1986.

[33] M. A. Szneps. *Sistiemy raspriedielienia informacji*. Mietody rasczіota. Radio i Swiaz, Moskwa, 1979.

[34] E. A. van Doorn and F. J. M. Panken. Blocking probabilities in a loss system with arrivals in geometrically distributed batches and heterogeneous service requirements. *IEEE/ACM Trans. Netw.*, 1(6):664–677, 1993.

[35] R. Wilkinson. Theories of toll traffic engineering in the USA. *Bell System Technical Journal*, 40:421–514, 1956.

[36] E. W. M. Wong, A. Zalesky, Z. Rosberg, and M. Zukerman. A new method for approximating blocking probability in overflow loss networks. *Computer Networks*, 51(11):2958–2975, 2007.

Design and Traffic Engineering of VoIP for Enterprise and Carrier Networks

James Yu and Imad Al Ajarmeh
DePaul University, Chicago, Illinois, USA
jyu@cdm.depaul.edu iajarmeh@cdm.depaul.edu

Abstract

The paper presents an extension of the Erlnag-B model for traffic engineering of Voice over IP (VoIP). The Erlang-B model uses traffic intensity and Grade of Service (GoS) to determine the number of trunks in circuit-switched networks. VoIP, however, is carried over packet-switched networks, and network capacity is measured in bits per second instead of the number of trunks. We study different network designs for VoIP, and propose a Call Admission Control (CAC) scheme based on network capacity. We then propose a new measurement scheme to translate network bandwidth into the maximum call load. With this new metric, the Erlang-B model is applicable to VoIP. We conducted experiments to measure the maximum call loads based on various voice codec schemes, including G.711, G.729A, and G.723.1. Our results show that call capacity is most likely constrained by network devices rather than physical connections. Therefore, we recommend considering both packet throughput (pps) and bit throughput (bps) in determining the max call load. If network capacity is constrained by packet throughput, codec schemes would have almost no effect on the maximum call load.

Keywords: VoIP, Erlang B, Call Admission Control, Traffic Engineering, Packet Throughput

1. Introduction

The growing popularity of Voice over IP (VoIP) is evident on the residential, enterprise, and carrier networks. The traditional IP-based networks are designed for data traffic, and there is no engineering consideration for voice traffic which is sensitive to packet delay and loss. To meet the new challenges of network convergence of both voice and data services on the same network, traffic engineering is important to network design as well as to the continual operation of the services. This paper provides an in-depth study of the VoIP traffic engineering and presents an enhanced traffic engineering model for VoIP. Among the various available traffic engineering models, the Erlang-B model has been widely used to engineer the voice traffic of circuit-switched networks for many years [1]. The purpose of the Erlang-B model is to calculate the resources (outgoing trunks) based on the Grade of Service (GoS) and traffic intensity. An example of traditional circuit-switched network is illustrated in Figure 1.

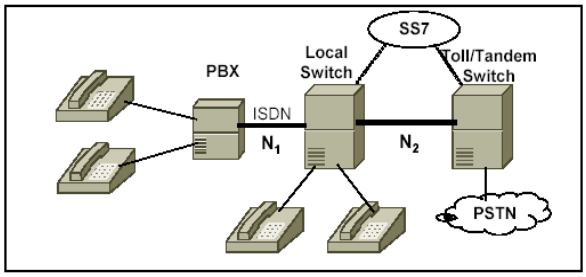


Figure 1. Legacy Telephone Network

The limiting resource in this network is the number of trunks between switches. For enterprise users, this resource is the number of trunks (N1) between their PBX and the local switch. If an enterprise subscribes too few trunks, the end-user would experience a high probability of blocking, for both incoming and outgoing calls. If the enterprise subscribes too many trunks, many of them will not be used, resulting in poor resource utilization and waste of money. On the carrier side of the network, the limiting resource is the number of trunks (N2) between a local switch and a tandem/toll switch. N2 is determined by network engineers to satisfy the traffic demand on the carrier core network. Traffic engineering is to calculate the required network resources (N1 or N2) based on the traffic demand and service requirements.

In packet-switched networks, there are no circuits or trunks. These networks accept any incoming packets. If the arrival rate of incoming packets is higher than the service rate of the network, constrained by network devices or outgoing links, packets will be buffered for later delivery. The effect of packet buffering is longer delay. If the buffer is full, new packets are discarded, which result in packet loss.

When packets are lost, an upper layer protocol between the sender and the receiver (not in the intermediate node) may retransmit the packet, which would result in even longer delay. Of course, some protocols, such as UDP, may ignore the lost packets and take no actions. This operation of packet-switching is not appropriate for voice communication which is sensitive to delay and packet loss.

This paper is an extension of our earlier publication [2] with expanded work on the design of an overlay network for VoIP, more detailed coverage on traffic measurement, and additional VoIP experiments. This paper is organized as follows: Section 2 provides a brief overview of how others are addressing the traffic engineering issue of VoIP. Section 3 explains the traditional Erlang-B model, and Section 4 presents the architecture and design of VoIP networks for the enterprise and carrier environment. A detailed analysis of VoIP traffic and its applicability to the Erlang-B model is given in Section 5. We present a comprehensive experimental design to emulate the VoIP traffic, and the results are given in Section 6. The last section, Section 7, presents the conclusion and some open issues for future work

2. Call Admission Control

The purpose of Call Admission Control (CAC) is to determine if the network has sufficient resource to route an incoming call. In the circuit-switched networks, the Call Admission Control algorithm is simply to check if there are circuits (or trunks) available between the origination switch and the termination switch. VoIP traffic is carried over packet-switched networks, and the concept of circuits (trunks) is not applicable. However, the need for Call Admission Control (CAC) of VoIP calls is the same. Packet switched networks, by nature, accepts any packet, regardless of voice or data packets. When the incoming traffic exceeds the network capacity, congestion occurs. Control mechanism is needed to address the issue of congestion by traffic shaping, queuing, buffering, and packet dropping. As a result of this procedure, packets could be delayed or dropped. Delay is usually not an issue for data-only applications. Packet loss can also be recovered by retransmission, which is supported by many protocols, such as TCP or TFTP. However, retransmission would cause longer delay which is not acceptable to time-sensitive applications. For voice traffic, delay and packet loss would degrade the voice quality, which is not acceptable to end-users. It should be noted that that CAC is different from Quality of Service (QoS) as frequently referenced in the literature. The main difference is that QoS is a priority scheme to differentiate the traffic already on the network, while CAC is to police the traffic from coming to the network when the network is congested [3].

CAC for circuit-switched network is implemented in the Q.931 and SS7 signaling [1]. Q.931 is to determine if there is a free B channel in the ISDN trunk and reserve the B channel for an incoming call. SS7 signaling is to identify a free DS0 channel between central office switches and reserve that DS0 channel for an incoming call. Although VoIP is on a packet-switch network, voice communications still require *circuits* (an end-to-end connection) to guarantee its voice quality.

There are many publications about ensuing voice quality over IP networks, and the general approach of Call Admission Control is to reject a VoIP call request if the network could not ensure the voice quality. CAC mechanisms are classified as measurement-based control and resource-based control.

Measurement-based Control: For measurement-based control, monitoring and probing tools are required to gauge the network conditions and load status in order to determine whether to accept new calls or not [4]. A protocol, such as RSVP, is required to reserve the required bandwidth before a call is admitted into the network.

Resource-based Control: In the case of resource-based control, resources are provisioned and dedicated for VoIP traffic. The resource for VoIP is usually calculated in network bandwidth [5]. The CAC approach in this paper is resource-based control, but our approach to calculating traffic demand is different from others.

Those two mechanisms are also referenced as link-utilization-based CAC and site-utilization-based CAC [6]. Another reference of these two methods is measurement-based CAC and parameter-based CAC [7]. In both CAC methods, the voice quality of a new call and other existing calls shall be assured after a call admission is granted.

3. The Erlang-B Model

The Erlang-B model is the standard to model the network traffic of circuit-switched networks. It is known as the blocked-calls-cleared model [8], where a

[1] SS7 signaling is for North America, and it is known as Common Channel Signaling (CCS) 7 or C7 internationally. Their functions are the same, but implementations are different.

blocked call is removed from the system. In this case, the user will receive an announcement of circuit busy. Note that a busy announcement is not the same as busy signal, which is the case when the callee is already on the phone. From the perspective of the Erlang-B model, not-answered-calls and busy calls are all considered successful calls. This section provides a brief overview of the Erlang-B model and its application to the circuit-switched network. Our goal is to enhance the model and apply it to the IP network.

3.1. Traffic Measurement

In a circuit-switched network, the limiting resource is the number of circuits which is also known as trunks (N). The traffic load on the network is measured by Traffic Intensity which is defined as

Traffic Intensity (A) = Call Rate × Call Holding Time

where call rate is the number of incoming calls during a certain period of time. Call Rate is randomly distributed and follows the Poisson distribution. Call Holding Time is the summation of (a) call duration which is the conversation time, (b) waiting time for agents at call center, and (c) ringing time [9]. The measurement unit of Traffic Intensity is *Erlang* which is the traffic load of one circuit over an hour. For example if a circuit is observed for 45-minute of use in a 60-minute interval, the traffic intensity is *45÷60=0.75 Erlang*.

The third parameter of the Erlang-B model is Grade of Service (GoS) which is *probability* of an incoming call being blocked. For a typical circuit-switched network, the reason for a call being blocked is that all trunks are busy. A GoS of 0.01 shows that there is 1% probability of getting a busy announcement. GoS is a critical factor for calculating the required number of trunks since it represents the trade off between service and cost. For a local telephone switch, if we set the number of trunks (to the tandem office) equal to the number of subscriber lines, the switch would have GoS=0 (100% non-blocking), regardless of the traffic load. Of course, this is a hypothetical example as no carriers would have this engineering practice.

3.2. The Model

The Erlang B model is commonly used to determine the mathematical relationship of the traffic measurements defined in Section 3.1. The assumptions of the Erlang B model are

Infinite number of sources: The model implies that an infinite number of users who could make a call through the network. In practice, if the number of users is much larger than the number of trunks, this assumption is considered valid.

Random call arrival: Since we have a large number of users, each user may initiate or receive a call at any time. The call arrival is random and follows the Poisson distribution, which also implies that the inter arrival time follows the exponential distribution. The randomness also implies that call events are independent of each other, where $Call_{[i]}$ and $call_{[i+1]}$ are two independent calls.

Blocked calls are cleared: When a call is blocked due to insufficient resources (trunks), the user will get a recording or a fast busy tone. The call request is discarded (cleared) by the network and the user must hang up and try again at a later time.

Random holding time: The holding time (call duration and waiting time) also follows the exponential distribution.

It should be noted that the assumptions of the Erlang-B model are transparent to the underlying networks, regardless of whether it is a circuit-switched network carrying traditional phone calls, or a packet-switched network carrying voice calls in the form of VoIP. Another important note is that the Erlang B model has been proved to be fairly *robust* where minor violation of model assumptions would still yield useful and practical results for traffic engineering. For example, one could argue that incoming calls are not totally *independent* of each other, especially during a special occasion. To address this concern, the standard practice is to take a conservative approach in measuring traffic intensity on the Busiest Hour of the Busiest Week/Season (BSBH) in a year. In other words, one should never engineer the network based on the *average* demand; instead, it should be based on *quasi-peak* demand. Based on the above assumptions, we can derive the mathematical formula for the Erlang B model:

$$GoS = (A^N \div N!) \div [\sum_{k=0}^{N} (A^k \div k!),\ k=0,N]$$

where A is Traffic Intensity in Erlangs and N is number or trunks.

Due to the popularity of the Erlang B model among network engineers, an on-line calculator is available to calculate the model parameters [10].

4. Voice over IP (VoIP) Networks

This paper studies three VoIP architectures: (1) enterprise network, (2) access network of Internet service provider, and (3) VoIP carrier network.

4.1. VoIP network for Enterprise

The VoIP network for enterprise is illustrated in Figure 2.

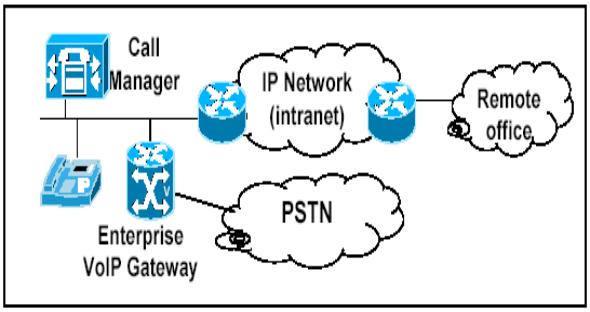


Figure 2. VoIP for Enterprise Networks

In the enterprise network, voice calls are carried over the packet-switched IP network within the enterprise. The VoIP network has an interface to the PSTN network, usually a T1 link. At the perimeter, the VoIP gateway provides the signaling interworking between Session Initiation Protocol (SIP) and Q.931/ISDN. The signaling function is to establish a duplex end-to-end connection between the caller and the callee, and it could be initiated from either direction. After the call setup, the VoIP gateway extracts the voice payload from the IP packets (for outgoing calls) or encapsulates the voice payload onto the IP packets (for incoming calls).

In some implementations, the enterprise phone network consists of IP phones, and a Call Manager. In other cases, the enterprise local phone system has both IP and analog phones. In the latter case, the call control process requires a hybrid PBX supporting both IP and analog calls [11].

Traffic engineering for the enterprise network has two elements. The first one is the engineering of the trunk capacity (number of DS0's) to the PSTN, and the Erlang-B model is applicable for this element. The second element is the network capacity (in bps) on the enterprise network which carries both voice and data traffic as illustrated in Figure 3.

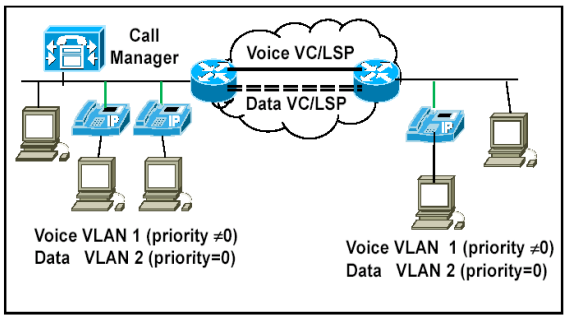


Figure 3. Enterprise Voice and Data Network

In general, Local Area Network (LAN) is either 100BaseTX or Gigabit Ethernet with capacity up to 1000Mbps. Although it is unlikely to see network congestion on LAN, we need to consider the bursty nature of data traffic. Therefore, our recommendation is to enable VLAN-tagging (802.1Q) with priority (802.1p). 802.1p supports a 3-bit priority scheme, with up to eight priority queues. Most Ethernet switches and IP phones support 802.1Q/p, but many support only two priority queues: priority≠0 for priority (voice) traffic and priority=0 for best effort (data) traffic. Frames with priority≠0 have priority over frames with priority=0 and will be processed first. With this priority scheme, we could consider 100% of the LAN bandwidth is reserved for voice traffic. If there is no voice traffic, Ethernet switches will then forward data frames. Because of the high capacity bandwidth of Ethernet and the use of 802.1p, traffic is unlikely to encounter congestion on the LAN.

The Wide Area Network (WAN), however, has relatively low bandwidth, usually from 1.5M (T1) to 45M (DS3). In rare cases of large enterprises, it could go up to 155M (OC-3). Figure 3 illustrates an example of a single connection between two locations, and this connection needs to carry both voice and data traffic. As discussed in Section 2, we propose to use the resource-based control mechanism where we provision a dedicated connection for voice traffic. The dedicated connection could be a physical link, an ATM or Frame-Relay Virtual circuit (VC), or an MPLS-based Label Switch Path (LSP). The dedicated connection has guaranteed bandwidth for voice traffic, and the traffic engineering model will be based on this bandwidth. This network design does not need to consider the bursty nature of data traffic and would never experience network congestion (for voice traffic) if Call Admission Control (CAC) is implemented. The Call Manager decides whether to accept or reject an incoming call request based on provisioned bandwidth and available bandwidth.

4.2. Access Network

The second VoIP architecture is the access network, where an enterprise subscribes to the VoIP service through an Internet Service Provider (ISP). The network architecture is illustrated in Figure 4. Because the VoIP traffic is carried over the public Internet which is a best-effort network and does not support QoS, we cannot apply Call Admission Control in this architecture. The engineering of trunks between the ISP voice gateway and the PSTN follows the Erlang-B model as described in Section 3.2.

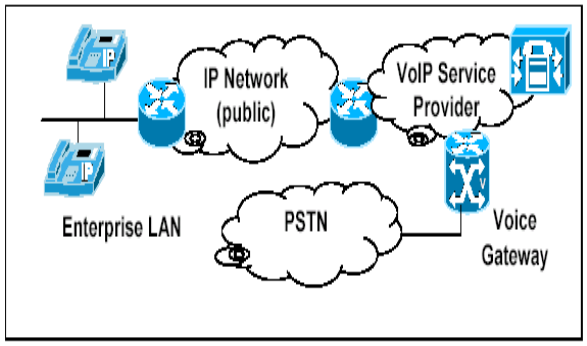


Figure 4. VoIP for Access Networks

4.3. Tandem Service over a Carrier Network

The third VoIP architecture is tandem service over the carrier network as illustrated in Figure 5. The two major network elements are Voice Trunking Gateway and Softswitch. Voice Trunking Gateway receives Voice Time Division Multiplexing (TDM) traffic from legacy voice switches and converts it to IP packets and forwards the packets to the IP backbone for transport. Softswitch uses the Signaling System 7 (SS7) to interface with the legacy voice switches and also to interface with other softswitches. The purpose of the SS7 is to establish an end-to-end connection between the caller and callee. It should be noted that the edge router may also accept VoIP traffic from another VoIP carrier.

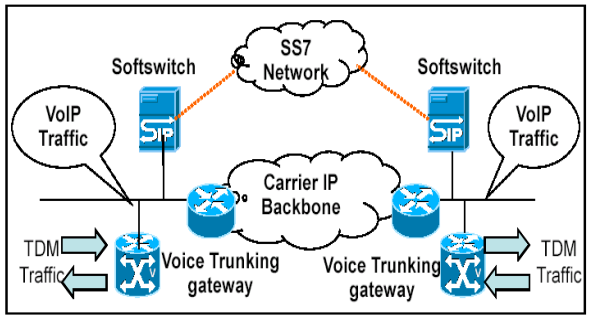


Figure 5. VoIP for IP-based Carrier Networks

Figure 5 shows only the voice traffic, and the IP backbone carries both voice and data traffic as illustrated in Figure 6.

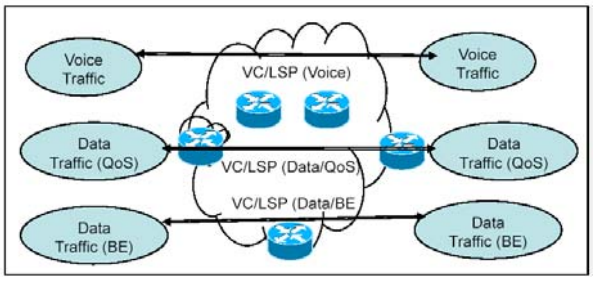


Figure 6. Carrier IP Backbone

In our network design of resource-based control, we propose three over-lay networks on the IP-backbone: voice network, QoS data network, and Best Effort (BE) data network. As discussed earlier, we could use either Virtual Circuit (VC) or Label Switch Path (LSP) to provision virtual connections and create the overlay network among physical nodes. Because voice network is a dedicated network, we could avoid the network congestion issue by implementing Call Admission Control (CAC) on softswitches. If the voice network has capacity to ensure voice quality for a new call, the call is accepted and the softswitch uses the SS7 signaling protocol to establish a connection over the IP backbone. Otherwise, the call request is rejected. Traffic engineering is to calculate the demand and determine the bandwidth required on the voice overlay network to ensure Grade of Service (GoS).

5. VoIP Traffic Analysis

VoIP packets are transported over Real-time Transport Protocol (RTP) which in turn uses UDP. RTP provides sequencing and time-stamp to synchronize the media payload. Real-time Transport Control Protocol (RTCP) is used in conjunction with RTP for media control and traffic reporting. Our experiment shows that RTCP is only about 1% of the VoIP traffic, so RTCP traffic is excluded in our analysis for traffic engineering.

5.1. VoIP Packet Overhead

VoIP encapsulates digitized voice in IP packets. The standard Pulse Code Modulation (PCM) uses 256 quantization level and 8,000 samples per seconds. As a result, we have a digitized voice channel of 64 kbps

(DS0). If we use 20ms sampling interval, each sample will be

64,000 bps × 20 ms = 1,280 bits = 160 bytes

This digitized voice is then encapsulated in an RTP/UDP/IP packet as illustrated in Figure 7 [12].

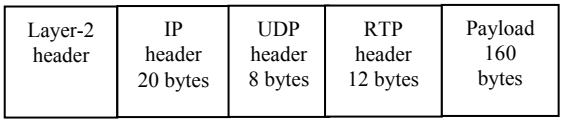


Figure 7. VoIP Frame

If the layer-2 is Ethernet, the 802.3 frame header, Frame Check Sequence (FCS), preamble, and Inter-Frame Gap (IFG) add additional 38 bytes. If the layer-2 is Point-to-Point Protocol (PPP), its header and FCS are 7 bytes.

PCM is the standard codec scheme for G.711, which does not use any voice compression algorithm. If a codec compression algorithm is used, the bandwidth for a voice channel is reduced to 8 kbps for G.729A and 5.3-6.3 kbps for G.723.1. Some codec schemes employ a silence compression mechanism where the bit rate is significantly reduced if no voice activity is detected. Furthermore, look-ahead algorithms are used in order to anticipate the difference between the current frame and the next one. In this paper we do not address those enhancements. A summary of voice codec schemes is shown in Table 1 [13].

Table 1. Vocoding and VoIP Overhead

	G.711 (10 ms sampling interval)	**G.711** (20 ms sampling interval)	**G.729A** (20 ms sampling interval)	**G.723.1** (30 ms sampling interval)
Raw BW in bps [1]	64,000	64,000	8,000	5,300
VoIP Payload (bytes)	80	160	20	20
VoIP overhead (802.3)	78	78	78	78
VoIP overhead (PPP)	47	47	47	47
BW in bps (802.3)[2]	126,400	95,200	39,200	26,133
BW in bps (PPP) [2]	101,600	82,800	26,800	17,867

[2] The bandwidth (BW) is for one voice channel. Required Bandwidth includes the overhead based on the codec packet sampling rate.

5.2. VoIP Traffic Characteristics

VoIP Systems use two types of messages on the IP networks: (a) Control Traffic, and (b) IP Voice Payload Traffic. The control traffic is generated by the call setup and management protocols and is used to initiate, maintain, manage, and terminate connections between users. VoIP Control traffic consumes little bandwidth and does not require to be included in the traffic engineering modeling. It is possible to provision another overlay network for signaling messages which have more stringent requirements than the payload traffic.

IP voice payload traffic consists of the messages that carry the encoded voice conversations in the form of IP packets. This type of traffic is what concerns network engineers as it requires relatively high bandwidth and has strict latency requirements. IP Voice payload Traffic is referred to as VoIP traffic and has some unique characteristics that require special handling and support by the underlying IP networks. The traffic characteristics that should be considered for VoIP networks are:

Real Time Traffic: Voice conversations are real time events. Therefore, transmitting voice data over IP networks should be performed as close to real time as possible, maintaining packet sequence and within a certain latency and latency variation (jitter) limits.

Small Packet Size: In order to minimize the sampling delay and hence maintain the latency constrains, VoIP data is carried in relatively small IP packets.

Symmetric Traffic: VoIP calls always generate symmetric traffic, same bandwidth from caller to calee and from callee to caller. This characteristic of VoIP traffic combined with the small packet size will have impact on the network devices as we will see later in this article.

Any-to-any Traffic: any user might call any other user on the VoIP network which limits the ability of network engineers to predict the path of traffic flow. VoIP traffic might be initiated or terminated at any terminal point of the network, unlike many of the IP data networks where the majority of the traffic flows are known (e.g., clients to servers).

5.3. VoIP Call Requirements

Although human ear can tolerate some degradation in the voice quality and still be able to understand the

conversation; however, there are certain requirements that should be met so that a VoIP call is acceptable. Transporting a Voice Call over the packet switched network has many challenges posed by the nature of the IP-based network which was originally designed for the data traffic. On the VoIP network, the major factors that determine voice quality are given as follows:

Delay: Represents the one-way end-to-end delay which is measured from speaker's mouth to listener's ear (mouth-to-ear). Delay includes coding/decoding, packetization, processing, queuing, and propagation delay. The ITU-T G.114 [14] recommends for the one-way delay to be less than 150 ms in order to maintain a quality conversation and transparent interactivity. If VoIP packets are delayed more than this limit, collisions might happen when the call participants talk at the same time.

Jitter: This is a measure of the variation in time of arrival (TOA) for consecutive packets. The original voice stream has fixed time intervals between frames; however, it is impossible to maintain this fixed interval on the IP network. The variation is caused by the queuing, serialization and contention effect of the IP networks. VoIP endpoints provide jitter buffers to compensate for the variation in TOA and to support the re-sequencing process. Packets enter the jitter buffer at a variable rate (as soon as they are received from the network) and are taken out at a constant rate for proper decoding. Buffering increases the overall latency and the jitter buffer size should be carefully chosen in a way to keep the overall latency (one-way delay) within the acceptable range. Packets arriving outside the jitter buffer boundaries will be discarded. Jitter calculations should also consider voice activity detection, out of order packets, and lost packets.

Packet Loss: Unlike data connections, VoIP has some tolerance to packet loss; however, if packet loss ratio exceeds a certain limit the quality of the call will be negatively affected. Several reasons might lead to packet loss in a network such as network congestion, transmission interference, attenuation, rejection of corrupted frames, and physical link errors. Different voice codec schemes have different tolerance to packet loss; however, it is recommends that packet loss be kept bellow 1%. It should also be noted that some packets might reach the intended destination and yet be dropped because they are late by more than the jitter buffer value. Therefore, measuring packet loss must also include the jitter buffer loss which is a factor of jitter buffer size and packet delay variation.

Vocoding (voice codec): the vocoding scheme is another important factor in determining voice quality. A codec scheme could implement compression algorithm, redundancy and lost packet hiding techniques. Different vocoding schemes also generate different digitally encoded voice frames in terms of frame size, bit rate, and the number of frames per second.

5.4. Measurement of Voice Quality

Based on the above requirements for VoIP calls, the ITU-T standard provides the following guideline for the voice quality measurement [15]:

Table 2. VoIP Quality Measurement

Network Parameter	Good	Acceptable	Poor
Delay (ms)	0-150	150-300	> 300
Jitter (ms)	0-20	20-50	> 50
Packet Loss	0-0.5 %	0.5-1.5%	> 1.5%

A common voice quality measurement scheme is the Mean Opinion Score (MOS) where different voice samples are collected and played back to a group of people who rank the voice quality between 1 and 5 (1 is the worst and 5 is the best). An MOS of 4 or better is considered toll quality. The objective of Call Admission Control is to prevent network congestion so that all calls could achieve toll quality or better.

5.5. Erlang B Model for VoIP

In the previous sections, we studied different VoIP architectures, network design, VoIP call requirements and traffic engineering using Erlang-B model. This section presents how to use the Erlang-B model to engineer the VoIP traffic so that we can provide the optimum solution to balance between service quality and cost. The goal is to provide adequate bandwidth and network devices capable of supporting the call demand. In VoIP networks, the concepts of Grade of Service (GoS), and traffic intensity (call arrival rate and call holding time) are the same as in circuit-switched networks. However, the number of trunks in the Erlang-B model is not applicable to a packet-switched network. Therefore, we propose to use the maximum number of simultaneous calls with toll quality. This parameter is also referenced as *maximum call load* in this paper. We will provide an experimental framework to measure this parameter in Section 6. This parameter is comparable to the number of trunks used in the Erlang-B model. With the

proposed revision, the Erlang-B model has the same three parameters:

A: Traffic Intensity
GoS: Probability of blocking calls
N: Max Call Load

6. Experimental Design and Analysis

We developed an empirical framework to emulate the VoIP traffic in the lab environment. The emulated VoIP traffic is the UDP traffic with the payload size equal to the RTP header and vocoding data.

6.1 VoIP Traffic Emulation

Our experiments were performed using different network links and architectures. The lab configuration is illustrated as follows:

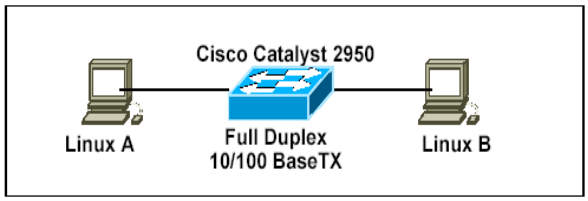


Figure 8a. VoIP Test over Switched Ethernet

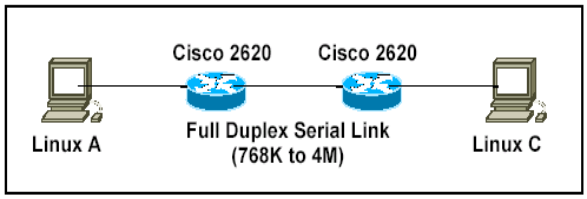


Figure 8b. VoIP Test over Serial Interface

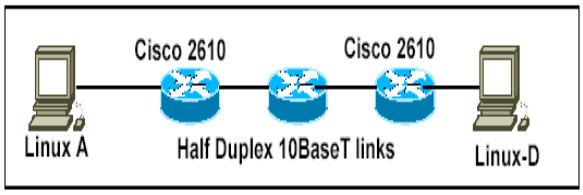


Figure 8c. VoIP Test over Routed Ethernet

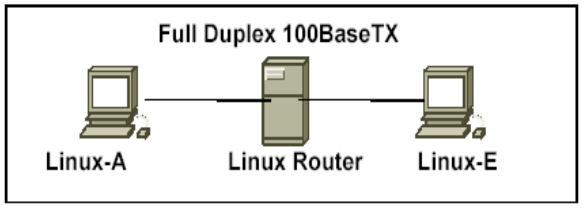


Figure 8d. VoIP Test over Routed Fast Ethernet

The switched Ethernet environment is for the baseline measurement which is to ensure the validity of our measurement tool and the measurement process. The low speed link (serial interface up to 2Mbps) is to emulate the enterprise intranet, and the high speed links (4Mbps and up) are to emulate a potential carrier IP backbone.

In each experiment run, the sender sends a batch of UDP messages (with a sequence number and a time stamp on each message) to the receiver. When the receiver receives messages, it echoes them back immediately. The symmetric traffic is to emulate a voice call. When the sender receives the echoed message, it computes the delay and then sends the message with a new time stamp and a new sequence number. The number of messages in the batch is similar to the TCP window for flow control and congestion control. Our objective is to achieve the maximum link utilization by having the maximum number of messages in the batch without causing any congestion or packet loss. When network congestion or packet loss happens, it implies poor voice quality.

During the experiment, we also monitor the CPU utilization of the sender and receiver machines. If the CPU utilization is above 60%, we consider the experiment invalid as the bottleneck is on the CPU and not on the network. We also conducted a baseline measurement in which we use the message size close to the MTU of 1,500 bytes. The purpose of the baseline measurement is to demonstrate that the experiment is able to achieve the wire speed performance. The expected results (theoretical limit) are calculated based on the overall bandwidth requirements for each codec shown in Table 1. Table 4 shows a summary of the theoretical maximum call load for different codec schemes on different links.

Table 4. Theoretical Call Capacity

Links	G.711 (20ms)	G.711 (10ms)	G.729A (20ms)	G.723.1 (30ms)
FD FT1 (768k)	9.3	7.6	28.7	43
FD E1 (2.0M)	24.2	19.7	74.6	111.9
FD 2×E1 (4.0M)	48.3	39.4	149.3 [1]	223.9[3]
10BaseT (HD)	52.5	39.6	127.6 [1]	191.3[3]
10BaseT (FD)	105		255.1	382.7
100BaseTX (FD)	1,050	791.1	2,551	3,827

[3] Note that a Full Duplex Serial link of 4.0M carries more calls than a half duplex 10BaseT link because PPP has less overhead than Ethernet. (See Table 1

The following section presents the experimental results. We compare the experimental results with the theoretical limits presented in Table 4 as follows:

Utilization = experimental result ÷ theoretical limit

This new metric is to measure the efficiency of a link for voice calls, and it is different from the traditional measure of data throughput and link utilization.

6.2. Experiment Results

The first experiment is a VoIP traffic test over a full duplex 10/100BaseTX link. The key measurement is the maximum number of simultaneous calls with toll quality (max call load). The results of this experiment are presented in Table 5. The column labeled "utilization" is the comparison to the theoretical limit presented in Table 4. Figure 9 shows a graphical comparison between the theoretical and experimental max call limit on a 10BaseT full duplex link.

Table 5. 10BaseT Full Duplex Switched Link

Message Size (bytes)	Codec	Max call Load	Utilization (%)
1450	(baseline)	---	96%
160	G.711	105	100%
20	G.729A	251	98%
20	G.723.1	376	98%

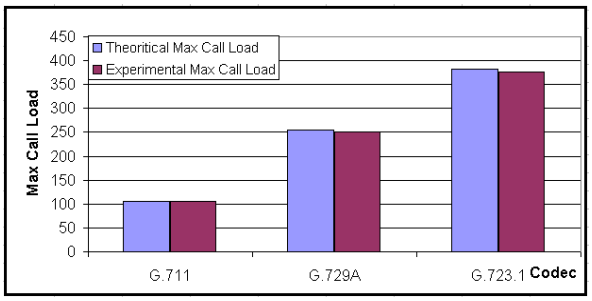


Figure 9. 10BaseT FD Switched Link

When we tried to run this experiment over the 100BaseTX link, the CPU utilization of the Linux machine reached 98%. Therefore, the experiment of 100M is considered not applicable for measuring the max call load.

The second experiment is to test the VoIP traffic over a serial link with two routers; we configured the link speeds to 768Kbps, 2Mbps, and 4Mbps. The results are given in Table 6. Figure 10a, Figure 10b, and Figure 10c show the graphical comparison between the theoretical and experimental max call limit on a 768Kbps, 2Mbps, 4Mbps serial links respectively.

Table 6. Full Duplex Serial Links (2 routers)

Codec	Serial Link (768K)		Serial Link (2M)		Serial Link (4M)	
	Max Load	Util.	Max Load	Util.	Max Load	Util.
Baseline	---	98%	---	98%	---	98%
G.711	9.2	99%	24.2	100%	40.0	83%
G.729A	28.0	98%	61.5	82%	70.0	47%
G.723.1	42	98%	92.3	82%	105.0	47%

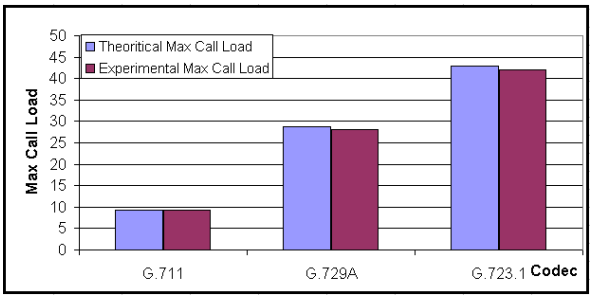


Figure 10a. Serial Link (768Kbps)

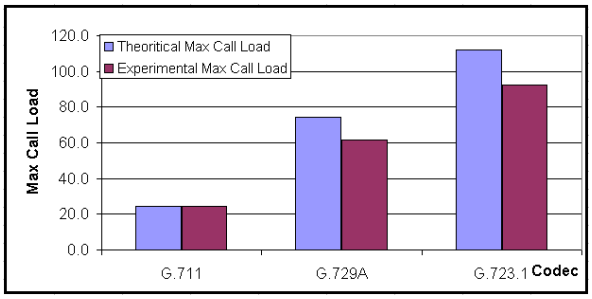


Figure 10b. Serial Link (2Mbps)

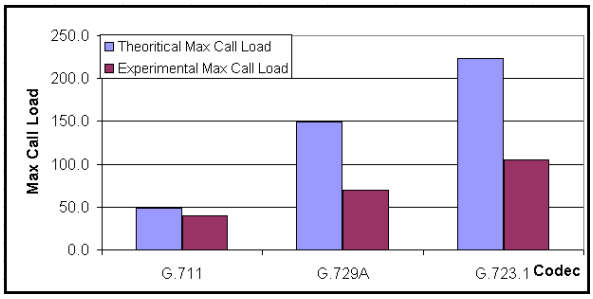


Figure 10c. Serial Link (4Mbps)

The third experiment is to emulate VoIP over three routers with 10BaseT link (half duplex), and the results are presented in Table 7 and Figure 11. During the experiment run, we also monitor the CPU utilization of traffic transmitter and receiver. The CPU utilization on the transmission side is 40% for G.723.1 and G.729A and 20% for G.711. The utilization is much lower on the receiver side, less than 10% in all cases.

Table 7. 10BaseTX Routed Link

Codec	Half Duplex (10BaseT)	
	Max Call Load	Utilization (%)
Baseline	---	97%
G.711	41	78%
G.729A	73	57%
G.723.1	109.5	57%

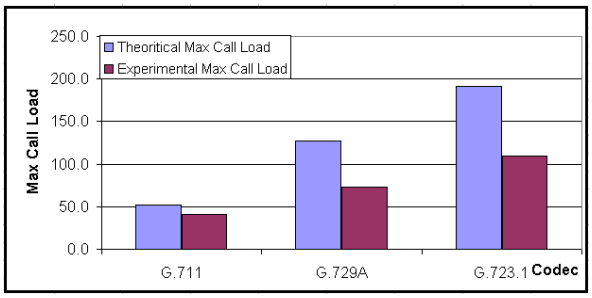


Figure 11. 10BaseTX HD Routed Link

The fourth experiment is to emulate VoIP over a routed full duplex 100BaseTX link. In this experiment, we used a Linux-Based router on a Pentium 4 machine, and the CPU utilization for sender and receiver is less than 40% in all cases. The results of this experiment are shown in Table 8 and Figure 12 bellow.

Table 8. 100BaseTX Routed Links

Codec	Full Duplex (100BaseTX)	
	Max Call Load	Utilization (%)
Baseline	---	97%
G.711	390	37.1%
G.729A	465	18.3%
G.723.1	897	18.2%

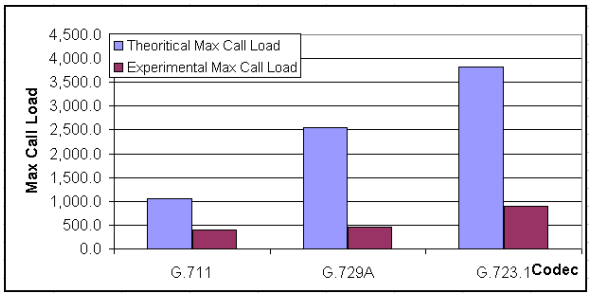


Figure 12. 100BaseTX FD Routed Link

A summary of the observed maximum call loads versus expected (theoretical) maximum call loads is shown in Figure 13.

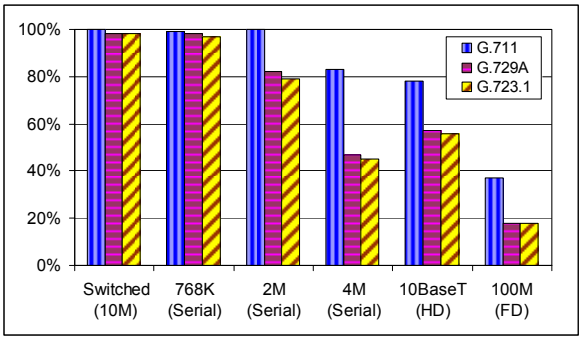


Figure 13. Call Utilization for Various Links

The fifth experiment is to study the effect of the sampling interval on the maximum call load. In this experiment we changed the sampling interval for G.711 to 10ms, and the payload size was also changed to 80 bytes. We ran the experiment over 10BaseTX full duplex switched link and 10BaseT routed link. Table 9 and Figures 14a and 14b show the comparison between Max Call Load and link utilization for different packet sampling rates.

Table 9. Call Load and Packet Sampling Rate

Codec	10BaseT Switched Link		10BaseT Routed Link	
	Max Call Load	Util.	Max Call Load	Util.
G.711 (10ms)	77	98%	26	67%
G.711 (20ms)	105	100%	41	78%

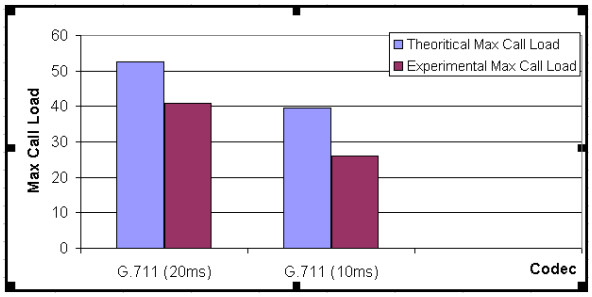


Figure 14a. Packet Sampling Rates and Codec on 10BaseT Half Duplex Link

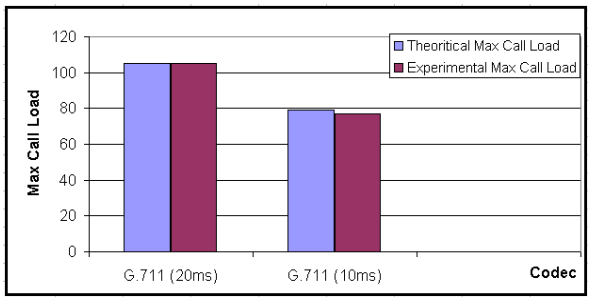


Figure 14b. Packet Sampling Rate and Codec on 10BaseT Full Duplex Link

The observations from these experiments are summarized as follows:

1. We are able to achieve wire speed performance (96% or better) using the max message size in all experiments. This result confirms the validity of the measurement tool and the experiment process.
2. The data shows close to 100% utilization on 10BaseT switched Ethernet (Table 5.) It shows that we could achieve the max call load as calculated from the available bandwidth.
3. In the cases of routed networks, we observed close to 100% utilization only on low speed links, but poor utilization on high speed links. It shows that the max call load cannot be achieved on the high speed links.
4. G.711 always yields better utilization than G.729A which is comparable to G.723.1. It shows that the smaller size for a codec scheme would yield lower utilization on the link. This is an interesting result, and we will investigate further later.
5. Although G.729A and G.723.1 compress the voice payload by a factor of 8-10, their improvement to the max call load is less than 10% on high speed links.
6. When using larger packet sampling rates (from 10ms to 20ms), we notice significant increase in the Max Call Load.

In summary, the experimental results raise a question about how to measure call loads for VoIP. Many other studies calculate the call load based on the bit throughput (bps), and our experiment shows that bps alone could not explain the results observed in the experiment as there is a large discrepancy between observed data and calculated data.

6.3. Packet Throughput and Max Call Load

Our lab experiments show that in the case of low utilization, it always involves routers. This observation leads to the study of packet throughput (number of packets processed per second) of network devices. The routers used in this experiment are Cisco 2610 and Cisco 2620. According to the product specifications [16], these routers are able to carry 1,500 packets per second (pps). If Cisco Express Forwarding (CEF) is enabled and the traffic pattern is applicable, the router could achieve 15,000 pps. Each VoIP call requires two connections (one in each direction) and this is the symmetric characteristic of VoIP traffic we discussed in Section 5.2.

The way pps is calculated for router is that each packet is counted twice as it goes through the incoming port and the outgoing port. If we use 20ms sampling interval and 64-byte frames, the calculated max call load of a router would be

15,000 pps ÷ (1000 sec ÷ 20 ms) ÷ 4 = ***75 calls/sec***

And for 30ms sampling interval (G723.1) we have

15,000 ÷ (1000 ÷ 30) ÷ 4 = ***112 calls/sec***

These numbers are consistent with all the experimental results of the routers. In other words, the max call load is bounded by the router "capacity" rather than the link capacity.

We also noticed that we were able to achieve maximum utilization on the physical links for the baseline tests (using MTU as the packet size). The inconsistency in utilization leads to the question about the root cause of difference between the baseline tests and emulated VoIP tests. To answer this question, we need to study the VoIP traffic characteristics in 5.1 and compare with the processing of packets by network devices. We find that VoIP uses small packet size to transfer calls. In order to achieve higher link utilization using small packet size, we need to send more packets per second. Pushing more small packets into the network would not cause congestion on the link itself; instead, the routers on the network may not be able to process the demand and become the congesting point.

For example if we use G.729A codec on a half duplex 10BaseT link:

Frame Size = 98 bytes (or 784 bits)
20 byte (payload) +
8 byte (UDP) +
12 byte (RTP) +
20 byte (IP) +
38 byte (Ethernet, preamble, and IFG)

If we want to achieve full link utilization (10M bps) using G.729 codec, we need packet throughput of

10,000,000 bps ÷ 2 ÷ 784 bit/packet = 6,377pps

Since VoIP traffic is symmetric in both directions, we need the network to handle twice this amount. According to the product specification, each packet is counted twice as it goes through the router (coming and leaving). Therefore, the required packet throughput for the router is:

$$6{,}377 \times 2 \times 2 = 25{,}508 \text{ pps}$$

As discussed earlier, our router (Cisco-2600) is capable of processing only 15,000 pps. Because of this constraint, we observe a lower link utilization which is

$$15{,}000 \div 25{,}508 = 58.8\%$$

This calculated utilization is almost identical to our experimental results of 57% as presented in Table 7

This example of calculation is applicable to all the results we obtained in this research. It proves our point that the limiting factor (bottleneck) is on the router's capability to process packets rather than the network itself. Therefore, to provide sound traffic engineering for VoIP we need to consider *pps* as well as *bps*.

When we use a Linux machine as a router, we are able to achieve a much higher call load, close to 470 calls/sec (Table 8). However, this number is still far below the link capacity of 100BaseTX. In our experiments, each router has only two interfaces. If the call load is constrained by the router capability, then adding more interfaces to the router would further lower the utilization for each link.

If a carrier has a high-end router, such as Cisco 12000 series with the capability of 4,000,000 pps, this router could handle up to:

4M ÷ (1000 ÷ 20) ÷ 4 = ***20,000 calls/sec***
(Based on the 20ms sampling interval)

This capacity would be sufficient to achieve the theoretical limit of G.711 on a gigabit link, but still fall short for G.729A on the same link. If we choose a more aggressive packet sampling rate, such as 10ms, this capacity would not meet the demand of G.711 for a single gigabit link while most routers have multiple gigabit links and OC-3/OC-12 links.

If the bottleneck is on a network device (as we observed in our experiments), using a compression scheme would not solve the congestion problem. This is because most commonly used codec schemes require the same packet throughput. In other words, compression will not reduce the number of packets generated. The choice of the packet sampling interval, 10ms vs. 20ms, would significantly change the Maximum Call Load as it directly affects the transmitted number of packets per second.

The theoretical Maximum Call Load, if calculated based on bandwidth consumption, increases with the increase of the packet sampling rate. The reason is that higher packet sampling rate is associated with larger packet size and less overhead.

It should also be noted that Robust Header Compression (ROHC defined in RFC 3409) for RTP/UDP/IP does not improve max call load if the limiting factor is on pps instead of bps. ROHC reduces the header overhead but does not reduce the number of packets.

7. Conclusion

The Erlang-B model has been used by the telecom industry to determine the call capacity of circuit-switched networks for many years. We are proposing to use the max call load for VoIP networks as a comparable measure to network trunks. With this modification, the Erlang-B model is applicable to determine the call capacity of VoIP networks.

Packet-switched networks, by nature, do not have the concept of blocking, and all incoming packets are accepted even if the new packets will add more loads on the network which could result in delay and packet loss. In the case of VoIP, this will cause quality degradation to the new calls as well as to the existing ones. The solution to this problem is to use a Call Admission Control (CAC) where call manager or softswitch can apply the Erlang-B model to implement a CAC algorithm to accept or reject an incoming call request.

The traditional approach of calculating the maximum call load is based on network bandwidth, and our experiments show that this approach fails to work on some routed networks with high speed links. Our experiments show that packet throughput (pps) of network devices could be the constraint for VoIP traffic engineering. Based on our findings, network engineers should calculate not only the physical bandwidth of network interfaces but also the capacity of network devices. If the device capacity is the limiting factor, codec schemes would have no effect on the call capacity; instead, packet sampling interval could significantly change the maximum call load. For example, one of our experiments shows that increasing the packet sampling rate from 10ms to 20ms would increase the max call load by 37%. Of course, a higher packet sampling rate introduces longer delay which

will adversely affect voice quality. Therefore, this is a trade-off between call capacity and call quality in traffic engineering.

We also acknowledge one deficiency in applying the Erlang-B for VoIP traffic. Many VoIP implementations support silence suppression. During the silence time, the VoIP end-device (an IP phone or a VoIP gateway) may transfer small number of packets while the Erlang-B model assumes the same packet transmission rate as the talking state. This issue could be addressed by applying a new model for traffic intensity as presented in [17], and such a model is a direction of our future research.

Acknowledgement

This research project is partially supported by the Quality Instruction Council (QIC) grant of DePaul University. The authors would like to thank ISP, Inc. at British Columbia, Canada for its generous donation of a high capability Linux server for the experiment.

REFERENCES

[1] Cisco, "Voice Design and Implementation Guide" http://www.cisco.com/en/US/tech/tk1077/technologies_tech_note09186a0080094a8b.shtml

[2] James Yu and Imad Al Ajarmeh, "Call Admission Control and Traffic Engineering of VoIP," Second International Conference on Digital Communications, ICDT 2007, San Jose, CA, July 2007

[3] Cisco, "VoIP Call Admission Control" http://www.cisco.com/en/US/docs/ios/solutions_docs/voip_solutions/CAC.html

[4] Solange R. Lima, Paulo Carvalho, and Vasco Freitas. "Admission Control in Multiservice IP Networks: Architectural Issues and Trend," *IEEE Communications,* Vol. 45 No. 4, April 2007, 114-121

[5] Erlang and VoIP Bandwidth Calculator, http://www.voip-calculator.com/calculator/eipb/

[6] Shenquan Wang, et. al. "Design and Implementation of QoS Provisioning System for Voice over IP," *IEEE Transactions on Parallel and Distributed Systems*, Vol 17 No. 3, March 2006

[7] Xiuzhong Chen, et. al. "Survey on QoS Management of VoIP," International Conference on Computer Networks and Mobile Computing, IEEE 20-23 October 2003, 68-77.

[8] R. F. Rey (editor) "Engineering and Operations in the Bell System," AT&T Bell Laboratories, 1983. pp. 158-160

[9] Richard Parkinson, "Traffic Engineering Techniques in Telecommunications", Infotel Systems Corporation, April 2002

[10] Erlang on-line Calculator, http://www.erlang.com/calculator/

[11] Karen Van Blarcum, "VoIP Call Recording – Understanding The Technical Challenges of VoIP Recording", AudioCode Inc. White Paper, December 2004

[12] Bruce Thompson and Xiaomei Liu, "Bandwidth Management for the University Edge," Cisco, NCTA 2005

[13] John Downey, "Understanding VoIP Packet Sizing and Traffic Engineering," SCRE Cable-Tec Expo White Paper (June 2005) http://www.recursosvoip.com/docs/english/cdccont_0900aecd802c52e5.pdf

[14] One way Transmission time, ITU-T Recommendation G.114, May 2003

[15] A. Markopoulou, F. Tobagi, and M. Karam, "Assessing the Quality of Voice Communications over Internet Backbones", in IEEE/ACM Transactions on Networking, Vol.11, Issue 5, October 2003, pp.747-760.

[16] Cisco Portable Product Sheet – Router Performance http://www.cisco.com/web/partners/downloads/765/tools/quickreference/routerperformance.pdf

[17] Jorn Seger, "Modeling Approach for VoIP Traffic Aggregations for Transferring Tele-traffic Trunks in a QoS enabled IP-Backbone Environment", International Workshop on Inter-Domain Performance and Simulation, Austria, February 2003.

Mobile TV Research Made Easy: The AMUSE 2.0 Open Platform for Interactive DVB-H/3G Services

Raimund Schatz, Andreas Berger, Norbert Jordan

Telecommunications Research Center Vienna – ftw.
A-1220 Vienna, Austria
{schatz, berger, jordan}@ftw.at

Abstract

With the convergence of telecommunications and media, Mobile TV has become an intensively investigated and hotly debated new service class. While the different Mobile TV bearer technologies such as DVB-H have been extensively tested and standardized, the focus of attention is shifting towards advanced concepts that go beyond pure re-broadcast of television. In order to explore the possibilities of advanced interactive Mobile TV, the research community requires an open environment for prototyping technology on the network and service layer. However, required key components such as open, programmable TV-enabled phones and flexible white-box broadcast tools are still not available to the community. As one solution to this fundamental problem, we present our open source platform for mobile interactive TV for early stage technology and application prototyping, with a focus on mobile client, broadcast network and service aspects. Furthermore, we illustrate the utilization of the system and outline future development directions.

Keywords: Mobile TV, DVB-H, Interactive TV, Mobile Service Platforms, Service Prototyping.

1. Introduction

Mobile TV services are widely considered as major future growth driver in mobile multimedia markets. According to market research analysts such as Datamonitor, the mobile television market is set to grow exponentially – by 2010, 65.6 million people worldwide are expected to subscribe to mobile television services, growing up to 155.6 million subscribers in 2012 (Datamonitor, 2006). Such prospects have triggered a number of technological and commercial Mobile TV trials in Europe. Furthermore, it is expected that interactive content and services will add significant value to mobile broadcast service offers in terms of differentiation opportunities and new revenue streams (UMTS Forum, 2006). Common examples are quizzing, voting, chat as well as personalized ESG and advertisements. Mobile phones are prime candidates for delivering such interactive mobile TV experiences, since they natively provide the required back-channel via the cellular network. Concerning mobile TV technology R&D and standardization, much work has been already accomplished in the fields of media encoding and delivery, transport protocols for content delivery, service/content protection and basic ESG description. Nonetheless, there is a need for intendified research on advanced interactivity support and rich-media integration. However, advanced research on infrastructures for mobile interactive broadcast services remains difficult for two major reasons: a lack of versatile, programmable DVB-enabled mobiles and the lack of open, affordable and modular testbeds.

This article presents our approach to a mobile TV research platform and is structured as follows: In Chapter 2 we present the research project AMUSE 2.0, envisaged demo services and requirements for a hybrid DVB-H research platform. In Chapter 3 we briefly discuss the most relevant issues and standards concerning interactive broadcast services. In Chapter 4 we discuss different approaches towards extending Mobile TV with interactivity as well as related work on hybrid infrastructures. In Chapter 5 we present the AMUSE hybrid test platform with a focus on broadcast network, service framework and DVB-H client issues. We then discuss its application to our research use cases in Chapter 6 as well as our conclusions and planned future work in Chapter 7.

2. Project Background and Requirements

2.1 AMUSE Project Background

AMUSE 2.0 (Advanced Multimedia Services)[1] is an applied research project conducted at ftw., the Austrian competence centre for telecommunications research[2]. Within a consortium including partners such as mobilkom austria, Kapsch CarrierCom and Alcatel-Lucent, the project investigates mobile convergent services which we see as 'Mobile TV 2.0' – mobile TV beyond the currently rolled-out first generation of broadcast services which offer little or no interactivity. Upcoming next-generation TV services are characterized by advanced interactivity, user-to-user interaction, pervasive service access and made-for-mobile content formats (see Figure 1).

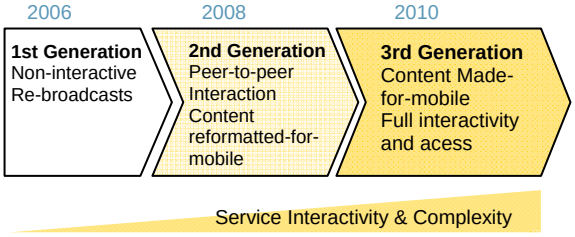


Figure 1. Mobile TV Generations Timeline (Schatz et al. 2007a).

In order to investigate the impact of such upcoming service generations and the enabling technologies required, our activities focus on the following aspects:

Hybrid mobile service platform architectures and clients that integrate broadcast and 3G/UMTS/WLAN connectivity. Key aspects are the tight integration of mobile services and IP-Datacast as well as the generation of interactive broadcast/unicast clients for mobile Symbian and Linux handhelds.

Interactive mobile broadcast services that leverage the potential of hybrid unicast/broadcast architectures. A particularly focus lies on the investigation of advanced interactivity, push-services, and person-to-person interaction.

Extensive user involvement throughout the project. Since real-world deployment is the only way to fully investigate the complex interactions between mobile applications, their users, and the environment, user testing in field settings using real-world telecom clients are an essential part of the project and thus need to be supported by its infrastructure.

[1] Project Homepage: http://amuse.ftw.at
[2] http://www.ftw.at

2.2 Mobile TV Platform Requirements

The given project profile and consortium necessitated the development of a custom research test environment for hybrid Mobile iTV services with the following requirements (Schatz et al., 2007a).

General requirements: a mobile interactive broadcast research platform must be highly *modular*, easily *extensible* and *flexible* enough to cover new mobile convergent service scenarios. Flexibility also demands for *programmable components* with open, well-documented APIs. As research projects tend to face major budget constraints (particularly concerning high-end equipment) system components should be *low-cost*, i.e. off-the-shelf hardware or open-source software at best. Nonetheless, components as well as architectures used must be *compliant* to common broadcast and telecommunications standards (such as TCP/IP, HTTP, MPEG, DVB-H, 3G/UMTS). Nonetheless, while accepting below carrier-grade equipment quality, the overall platform must be robust enough for performing user trials in the wild in a pre-commercial context.

Flexibility on bearer-level is another general key requirement for three reasons: on a pragmatic level, service prototyping and evaluation are facilitated by the option to bypass the DVB-H transmission by means of directly feeding multimedia packet-streams to the client by e.g. WLAN. Secondly, in the long run the user should be shielded from the prevailing diversity of standards (DVB-H, DMB, MBMS) and access networks (broadcast, unicast). The overarching goal here is providing seamless user experience across services and networks. Thirdly, the provision and seamless hand-over between different transmission paths is a hot research topic: broadcast-technologies such as DVB-H will exhibit coverage gaps, particularly in early roll-out stages. Handover to alternative bearers such as 3G/UMTS aids in maintaining quality of service, particularly in deep-indoor scenarios.

Specific requirements: project context (Europe) and consortium (telecommunications companies) demand for a focus on DVB-H (which in Europe at the moment is the mobile broadcast standard the strongest industry) as well as on using mobile 2.5G/3G smart-phones or comparable devices for the client side. The necessity to use standard platforms for mobile phones (and not PDAs or other larger, bulky mobile clients) such as J2ME or Symbian results from the project requirement to deliver results to telecommunications stakeholders, which favor a MNO-centric (Mobile Network Operator) operational model for Mobile TV.

This requirement is still non-trivial: at the time of writing, the only available DVB-H phones were closed feature phones with proprietary operating systems or Symbian-phones (such as Nokia N92, N77 and N96) which lack an open, documented DVB-H API[3].

2.3 Demo Service Scenarios

In order to guide the R&D process within our project, we developed a number of scenarios that feature the possibilities of advanced interactive Mobile TV services. Based on these scenarios we focused on the development of the following three demo services that we considered as most attractive from a commercial and technological perspective: Mobile Social TV, Live Sports, and CRM/Advertising.

Service 1 – Mobile Social TV. The remarkable success of the mobile phone as communication device suggests a fusion of entertainment features with person-to-person interaction. Similar to triple-play Social TV applications such as AmigoTV (Coppens et al., 2004), broadcast content (i.e., the currently aired TV show) can serve as context for social user-to-user interaction. The social interaction is enabled by IM (instant messaging via text and emoticons) and advanced presence (answering questions such as: Who is watching TV? Who watches the same programme?). To this end, the Mobile TV functionality is extended with public and private chat-rooms for mobile viewers (see Figure 2). Further features include *ShareMarks* which are "See-what-I-see" TV-content bookmarks and invitations exchanged among users via MMS and *JointZapping*, the synchronization of channel switching among peers (Schatz et al., 2007b).

Figure 2: Mobile Social TV with Chat.

Service 2 – Live Ski Race. This service enhances live TV sports coverage with interactivity features in the context of a ski race. It enriches streamed AV content with additional information such as current athlete and results. Parallel to watching the ski race, the user can browse the starting list, current rankings, and the list of not qualified runners. This information is regularly updated as the race goes on. In addition, personalization features allow for marking favorite athletes. In turn, notifications are sent to the user when one of them is about to start so that the runs of favorite athletes are not missed.

Service 3 – CRM/Advertising. Our third scenario addresses customer relationship management and click-through advertising. It utilizes the different enablers of the AMUSE platform to push advertisements to clients. When an advertisement is displayed the user can react to it (see Figure 3 below). For example, the service allows users to register with one-click for an SMS info channel, which the operator then uses to address users directly with relevant information about special offers, coupons, etc.

Figure 3: Advertising Banner with One-click Registration.

3. Interactive Broadcast: Standards and Related Work

This section discusses the most relevant standards and technologies that provide the foundation for interactive Mobile Broadcast TV.

3.1 DVB-H: Digital Broadcast for Handhelds

DVB-H (Digital Video Broadcasting – Handheld) is the digital broadcast standard for the transmission of broadcast content to handheld terminal devices, which was developed by the international DVB-Project[4] and

[3] See for example the Aug 2008 discussion on http://discussion.forum.nokia.com/forum/showthread.php?p=449105, [last access 12th Jan 2009]

[4] http://www.dvb.org

published in November 2005 by ETSI (European Telecommunications Standards Institute). DVB-H is based on the DVB-T standard for digital terrestrial television but is tailored to the special requirements of the pocket-size class of receivers (ETSI, 2004). Furthermore, the DVB-H data stream is fully compatible with DVB transport streams carrying legacy DVB-T streams. These properties guarantee that the DVB-H data stream can be broadcast in both, dedicated DVB-H and DVB-T networks.

As a transmission standard, DVB-H specifies the layer from physical up to network layer level. It uses a power-saving algorithm based on temporally multiplexed transmission of different services. The technique, called time-slicing, enables considerable battery power-saving. Additionally, time-slicing allows soft handover if the receiver moves from network cell to network cell.

Figure 4 shows the DVB-H protocol stack and characteristic extensions such as time-slicing. For reliable transmission under poor signal reception conditions, DVB-H introduces an enhanced error-protection scheme on the link layer. This scheme is called MPE-FEC (Multi-Protocol Encapsulation – Forward Error Correction). MPE-FEC performs additional coding on top of the channel coding included in the DVB-T specification in order to increase reception robustness for indoor and mobile contexts.

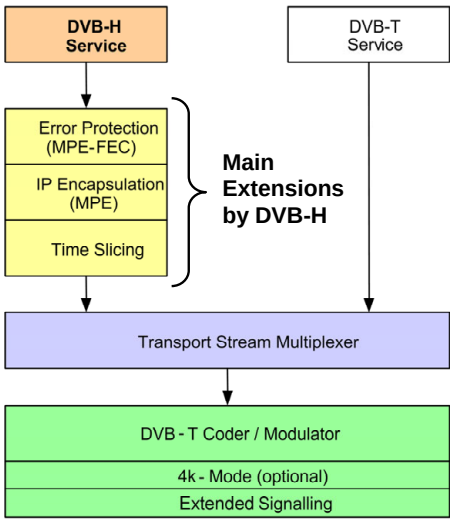


Figure 4: Protocol Stack Overview highlighting the main Extensions by the DVB-H Standard (based on ETSI 2004).

3.2 IP Datacast (IPDC)

In order to use DVB-H for delivering complete services to the end-user, protocols of the higher OSI levels on top of DVB-H are required. In addition to supporting the standard DVB applications like TV, radio and MHP, support for all kinds of services including the use of complementary cellular communications systems is required. To this end the DVB Project has introduced *IP Datacast (IPDC)* for an end-to-end system approach around DVB-H. The IPDC specification (ETSI TR 102 468; ETSI, 2006) defines the electronic service guide (ESG), service access management, delivery protocols, bearer signaling, QoS, mobility, roaming, and will further provide information on the terminal capabilities to make them suitable for IP Datacast.

IP Datacast specifies two transport protocols based on IP (RTP and FLUTE/ALC), since the IP protocol on its own does not serve all required use cases of service delivery. Services may be sent via RTP (Real Time Protocol) for *real-time* streaming content (for example a live TV channel). *Non-real-time* data (e.g., file downloads) is delivered by a FLUTE/ALC[5] (Paila, 2004) data carousel. For selecting the services, IPDC foresees an XML-based ESG[6] that contains metadata and access information about the available services (i.e., mostly TV-programmes), transmitted via FLUTE/ALC. Note, that a return-channel is not mandatory for IPDC which therefore specifies the UDP[7] protocol for connectionless transport.

3.3 Return-channel Interactivity for Mobile TV: Hybrid Architectures

Since IP Datacast via DVB-H constitutes a unidirectional transmission path, it enables only ocal interactivity. This means that viewers can only interact with e.g., ESG information or content previously downloaded to the terminal such as teletext, also known as *enhanced TV* (Jensen, 2005). However, more complex services such as chat and presence require a *two-way return-channel* to carry the viewer's commands and responses back to the service provider. This step actually allows for the evolution of Mobile TV towards complete interactive Mobile TV services in the sense of Jensen (2005).

In the context of Mobile TV the most suitable option for realizing the return-channel is the use of a packet-switched wireless 3G network. The advantages of this approach are threefold: bandwidth is sufficiently high (starting from 384kbit/s for base UMTS packed data service level), packet delay is low (<250 ms) and 3G is the standard connectivity offered by smartphones, the

[5] FLUTE/ALC = File Delivery over Unidirectional Transport / Asynchronous Layered Coding

[6] ESG = Electronic Serivce Guide, which in general also includes the EPG (Electronic Programme Guide)

[7] USP = User Datagram Protocol

main platform for Mobile iTV (UMTS Forum, 2006). The combination of both DVB-H and 3G for service delivery is described by the *hybrid network reference model* depicted in Figure 5 below. A hybrid network consists of a broadcast and a unicast path being jointly used in order to exploit their complementary advantages (cf. Hartl el al., 2005; ETSI, 2006).

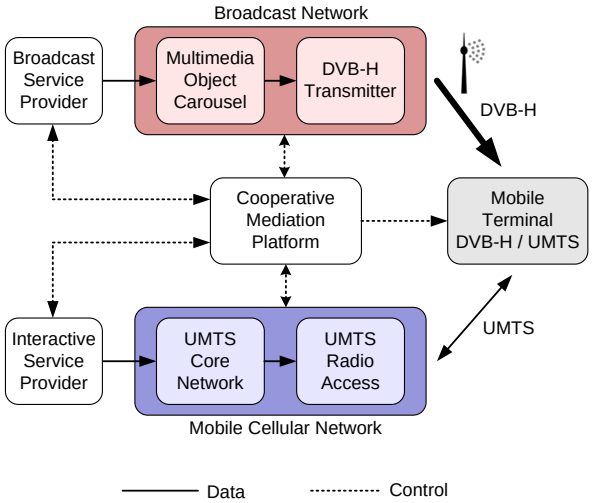


Figure 5: Hybrid Network Architecture applied to Mobile TV combining DVB-H Broadcast and Cellular Unicast.

From a purely theoretical perspective, the broadcast network as well as the mobile network can be used for multimedia content and interactivity data transmission to the mobile. Nonetheless, the most common scenario is the delivery of mass TV-content and EPG-information by DVB-H broadcast while interactivity and personal content are handled by the cellular network, thus taking the strengths of each bearer type into account. Furthermore, the hybrid reference architecture foresees system control and coordination between the two paths. This task is performed by the Cooperative Management Platform (CMP) and its interfaces (Hartl, 2005; Baldzer, 2005). The CMP handles interaction between the interactive service and the broadcast playout, for example when user voting determines the next TV clip to be played. In addition, a CMP may optimize the load distribution in the network and including proposals for load balancing between the DVB-H and UMTS transmission path to the mobile end-terminals. One example are videostreams, that the CMP routes over broadcast and unicast depending on overall popularity and number of users requesting it.

4. Extending Mobile TV with Interactive Multimedia: General Approaches

Mobile TV technology is still at an early stage of diffusion with standardization activities having mostly focused on the detailed specification of broadcast delivery of non-interactive TV content. In contrast, the mechanisms for interactivity and value-added services have only been addressed on a highly abstract level. For example, standardization within IPDC and OMA BCAST[8] does not specify triggers for elementary actions on the terminal (such as invoking an URL or menu choices) like for example the open source standard HisTV (Skrodzki, 2006). These circumstances have led to a highly fragmented landscape of proprietary approaches towards interactive Mobile TV (Petrovic, 2006; Baier & Richartz, 2006; Setlur et al., 2006) and a fragmented standardization landscape (Martinez, 2006). This heterogeneity is contrasted by a lack of standardized Mobile iTV approaches and open reference implementations, an issue that constitutes a major challenge for Mobile TV research and development (Kumar, 2007; Högg et al., 2007).

This section compares existing approaches and introduces a generic approach that takes advantage of established mobile service technologies and allows for rapid prototyping of Mobile iTV services on video-enabled smartphones. In general, the spectrum of currently investigated approaches to mobile interactive TV and Video can be divided into three categories: generic browser, rich media and download applications (Baier & Richartz, 2006).

4.1 Approach 1: Generic Browser

Generic browser approaches are characterized by the usage of a lean, universal software runtime as thin-client for rendering the statically declared content. This approach is ideal for application prototyping as well as large-scale deployments, since services do not rely on local code that needs to be adapted for each terminal platform. If only broadcast is used for content delivery, pre-defined scenes or pages are rendered locally, for example when DVB-H ESG service guide information is transmitted via FLUTE and locally browsed using the Mobile TV player. If the runtime is an HTML- or XHTML-browser, standard web-applications accessed via the unicast back-channel serve as the basis for Mobile iTV services. A representative usage example is a hypertext side-information browser to complement

[8] See also http://www.openmobilealliance.org/Technical/release_program

the current news coverage currently presented on the screen. This approach has considerable advantages, particularly for prototyping and development: since the majority of existing mobile multimedia services are web-applications (Ofcom, 2006), this approach takes advantage of mature web technologies and application server platforms (e.g., J2EE) that are widely supported in the telecommunications and web domain. Furthermore, service deployments and updates only need to be made on the server side. Nonetheless, a major drawback of this approach is its dependency on connectivity to the server. Applications that depend on frequent page-changes or a high number of images per page are vulnerable to wireless network latency or drops in bandwidth, since web-browsers issue separate HTTP-requests for each media element (Sullivan & Connors, 2008).

4.2 Approach 2: Rich Media

Rich-Media services are based on dynamic, interactive collections of multimedia data such as audio, video, graphics, and text. Applications range from movies enriched with vector graphics, overlays and interactivity to complex services with real-time interaction and different media types per screen, delivering a user experience as known from e.g., Adobe Flash[9] applications on the Internet. For the mobile domain Rich-Media yields the following advantages according to (Dufourd et al., 2005):

1. Graphics, animations, audio, video and scripts are packaged and streamed altogether. Rich-Media technologies are based on well defined and deterministic scene- and container-components that integrate the media content. This integration improves the fluidity and quality of the end user experience.
2. Full screen interactivity with multiple media streams. With the use of vector graphics, content can easily be made to fit the screen size, allowing the design of user interfaces similar to native mobile applications.
3. Real-time content delivery. Rich-Media allows for efficient delivery over constrained networks. Content can be delivered as streamed packages, allowing display of content as soon as the first packet is received. As such, services can be designed with reduced perceived delays and waiting times.
Unfortunately, because of these qualities, Rich-Media runtimes put very high demands on the capabilities of the device, particularly in terms of CPU-load and graphics processing. Therefore, the usage of Rich-Media in the context of Mobile iTV on current smartphone platforms is problematic, since the device also has to cope with the reception and display of the broadcasted video (Cazoulat & Lebris, 2008). Furthermore, the lack of open mobile player runtime components and server engines constitutes a major roadblock for the application of Rich-Media to Mobile iTV prototyping.

4.3 Approach 3: Download Applications

Download applications represent a thick-client approach that allows for the execution of complex, performance-intensive logic (e.g., games) locally on the mobile terminal. Typically, such applications are based on the J2ME[10] platform or they are native, i.e., specifically developed for an operating system such as Symbian S60. Unfortunately, these dependencies also narrow compatibility to very specific platforms and handsets which makes portability of native applications difficult. However, cross-platform portability and compatibility are also problematic for applications based on J2ME which is supposed to realize the vision of "write-once-run-anywhere" for mobile software development (Blom et al., 2008). J2ME suffers from several complications, including a large set of options as well as buggy and inconsistent virtual machines and package implementations (cf. Coulton et al., 2005). Nonetheless, the recently standardized JSR 272 Mobile Broadcast Service API for Handheld Terminals[11] for J2ME which uses an approach similar to MHP[12] Xlets has the potential to drive a broad uniform support of Java download applications by future Mobile TV handsets that implement this API. Table 1 overleaf compares the three approaches along with relevant projects and standards.

[9] http://www.adobe.com/flash

[10] J2ME = Java Micro Edition, see http://java.sun.com/javame

[11] The JSR 272 Mobile Broadcast Service API for Handheld Terminals is a standard effort lead by Nokia and Motorola to define a middleware-level API enabling the development of Mobile TV applications on J2ME. See also http://jcp.org/en/jsr/detail?id=272

12 MHP = Multimedia Home Platform, see www.mhp.org

Approach	Capabilities and complexity	Properties	Advantages (+) / Disadvantages (-)	Projects and Standards relevant to Mobile iTV
1. Generic Browser	Low	Generic browser client pre-installed on terminal Interactivity using static declarations of scene-description or pages	+ Portability + Limited demands concerning device capabilities + Simple deployment + Leverages established mobile service technologies and platforms - Limited multimedia features - Access to mobile device not standardized - Network dependencies	HisTV (www.histv.org), misc. 3G-Videostreaming Portals (e.g. Vodafone Live Mobile TV) XHTML, HTTP, IP-Datacast ESG
2. Rich-Media	Medium	Generic Rich-Media Player pre-installed Multimedia streaming or discrete content access Dynamically generated Scene-descriptions	+ Versatile multimedia and GUI capabilities + Enables asynchronous client/server communication and modular services - Complex technology - High demands on mobile device hardware/software - Fragmentation of standards and specifications - Only closed, proprietary runtimes available	MORE (Setlur et al. 2007), Ikivo, Streamezzo Flash, SVG-T, MPEG-LASeR (www.mpeg-laser.org), OMA-RME
3. Download Applications	High	Complete applications executed on the client Broadcast channel mainly used for application download, return-channel mainly for user feedback	+ High performance + Enables complex applications that leverage device capabilities - Portability and cross-platform service provisioning problematic - High development effort - On proprietary solutions available	HSP (Steckel,2006), Vodafone DVB-H Trial Client (Baier et al. 2006a) Java JSR 272 Mobile Broadcast API

Table 1: Comparison of the Three Main Approaches for Mobile Broadcast Interactivity (based on Baier & Richartz, 2006).

4.4 The AMUSE Approach to Interactivity: An Enhanced Generic Browser Client

Given the requirements stated in section 2.2, using a *generic browser client* emerges as the most suitable approach for a Mobile iTV R&D system. In line with the guidelines and recommendations for mobile and ubiquitous computing systems by (Greenhalgh et al., 2007), this approach is optimal for rapid prototyping of interactive services by shifting business- and presentation logic to the server. This is an important design feature, since despite recent advances in mobile operating systems, mobile phone application development is still an arduous and slow endeavor. Main reasons are limited debugging support, inconsistencies and errors in virtual machines and libraries, as well as long development cycles (Coulton et al., 2005; Huebscher et al., 2006). This general approach of using the generic browser as *the mobile application's main UI component* yields the following three key **advantages** (cf. Zucker et al., 2005):

1. **Flexibility and easy authoring.** Using markup allows for rapid development and customization, facilitating prototyping and the integration of new features, even if they are orthogonal to other services. The latter is important for features such as chat that have to be accessible also when the user interacts with other iTV services like side-information browsing.
2. **Dynamic updates.** Our approach also leverages the browser's inherent capabilities to dynamically update content from local and remote sources. This capability is particularly required for deployments in field evaluation settings where the application need to be managed remotely.
3. **Platform independence.** The player can be extended with additional GUI features without having to rely on OS-specific GUI programming, which increases portability e.g., between Symbian and Mobile Linux.

Concerning end-user experience, the chosen web-based approach allows for visual and navigational qualities similar to iTV set-top boxes, since mobile smartphone platforms such as Symbian S60 and Nokia Maemo[13] offer XHTML microbrowser components that can be integrated with video-based applications. This way, video and interactive content can be simultaneously presented in a split-screen fashion (see Figure 6). In addition, the presentation capabilities of

[13] Maemo is a Linux-based Mobile OS, used for Nokia N770, 800, 810 devices, see http://www.maemo.org

contemporary CSS-enabled XHTML microbrowsers have sufficiently matured to enable industry-grade Mobile TV information and entertainment services (Kumar, 2007; Liebermann, 2007). The key reason is that – compared to standard mobile application interfaces and web pages – mobile iTV services impose additional constraints in application design due to the fact that multiple content elements have to share the same screen. Mobile iTV application interfaces therefore are considerably simpler, since interaction designers have to severely limit visual and navigational complexity of services in order to avoid attention and size conflicts with the simultaneously displayed main video (Roibás, 2004; Trefzger, 2005; Knoche & McCarthy, 2005).

Figure 6: Split-screen Concept for the Mobile TV Client featuring a Live Sports Information service with Push and Pull Content.

Furthermore, the increasing capabilities (scripting, DOM[14] tree access) of mobile microbrowser runtimes such as the Nokia S60 WebKit[15] allow for the utilization of AJAX[16] (Asynchronous JavaScript and XML). AJAX constitutes a set of techniques that mitigate the shortcomings of purely HTML-based applications (such as page reloads, limited responsiveness) with mechanisms such as asynchronous event-processing and local partial updates which considerably improve the mobile user experience (Garrett, 2005).

[14] DOM = Document object model

[15] The S60 WebKit is a component used by Noka to equip its current range of Smartphones and has been pu into open source. See also the S60 WebKit Project Homepage, http://trac.webkit.org/projects/webkit/wiki/S60Webkit

[16] AJAX refers to the usage of a bundle of web standards (XML/XHTML, DOM, CSS, JavaScript, XMLHTTP-Request Object) in order implemenet web-applications with richer GUIs and improved user interaction similar to Rich-Media applications. Since updates and communication between client and server can happen asynchrously in the background, AJAX considerably lowers demand for page reloadsing and return-channel bandwidth.

4.5 Related Work

In the field of hybrid architectures for interactive broadcast services, the following projects are related to our work:

Hartl et al. (2005) have reported upon a system setup for a German DVB-H trial and implemented prototype application and some initial results of coverage measurements and service application feedback by friendly test users. The trial focused on the general technical feasibility of a hybrid DVB-H and GSM mobile multimedia broadcast system. Ollikainen and Peng (2006) go another step further and switch between the DVB-H and the UMTS networks without doing any frequency scans. They tested service handover approaches with almost no packet loss during the handover process. This, however, was only achieved by keeping the UMTS IP connection open all the time, which in practice means very high battery consumption. In the 'Podracing' project (2005-2007), which tested Mobile TV in 3G, DVB-H and WiFi networks, an interactive Mobile TV J2ME client and servier infrastructure was developed by Ollikainen et al. (2008). Similar to our approach, they used a browser run-time for interactivity. However, the project suffered common limitations of the J2ME runtime environment (i.e., no DVB-H API access, limited video integration) and thus focused less on parallel interactive service but rather on the evaluation of different content delivery mechanisms such as on-demand, download, and broadcast. Finally, Klinkenberg and Steckel (2006) introduced an approach to modularize interactive services in broadcast-only and respectively hybrid networks. Their work focuses on Java-applications complemented by XML-media descriptions, delivered to PDAs via DVB-H and the IP Datacast framework on top.

In addition to the demonstration of the general technical feasibility of hybrid network architectures by above projects, the AMUSE project features a browser-based thin-client approach on existing smartphones, true DVB-H reception, and a modular open source implementation.

5. The AMUSE Platform

The following discussion of the system architecture focuses on three key pars: broadcast chain, unicast path and mobile iTV client. The AMUSE platform includes an open extensible Mobile TV broadcast subsystem well as a J2EE framework for developing advanced interactive Mobile TV services. It is based on the requirements, technology choices and standards

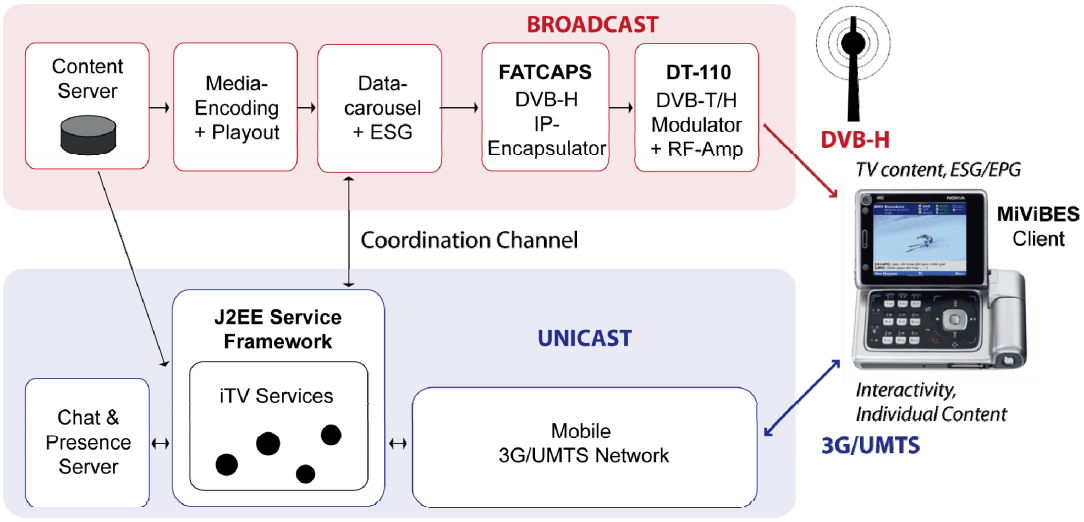


Figure 7: Architectural Overview of the AMUSE Mobile TV Platform.

discussed in the previous sections of this chapter. Figure 7 overleaf presents an architectural overview of the AMUSE system and the communication networks used. Modeled according to the hybrid reference architecture discussed in section 3.3, the platform consists of a broadcast path and a unicast path operating in a coordinated fashion. Furthermore, the environment includes an open mobile terminal client for interactive Mobile TV based on microbrowser runtimes. In this sense, the AMUSE Mobile iTV environment serves four major Mobile TV R&D purposes:

1. AMUSE enables the **operation of plain Mobile TV services**, since its components comply with the DVB-H and OMA BCAST standards. This means that DVB-H enabled phones such as the Nokia N92 can display the platform's broadcast output, while the MiVIBES player is also able to receive (unencrypted) DVB-H from other broadcast sources.
2. The platform allows for extensive Mobile TV **technology prototyping** since all components are open, white-boxed and easy to modify. For example, the broadcast path can be reconfigured down to the transport stream protocol level in order to evaluate new service types and content transmission schemes (Berger et al., 2008).
3. The AMUSE approach enables **rapid service prototyping** for Mobile iTV since the unicast subsystem is based on a service framework for developing applications based on J2EE technology. This means that new iTV services are developed as web-applications fully on the server-side based on SOA components and enablers, thus avoiding the difficulties and steep learning curve of mobile client development.

Finally, the AMUSE environment supports reliable user testing of Mobile iTV services in laboratory and in field settings, since users can interact with the applications on the same networks and terminals that are also used for real-world deployments in mobile communications contexts. Furthermore, platform and client support flexible switching between live and local simulation configurations and provide a mobile user experience that is not affected by unwanted, system-related usability issues (e.g., high latency, GUI rendering errors) caused by technical limitations of the platform itself.

The following paragraphs present a brief overview of the main subsystems of the AMUSE environment.

5.1 Broadcast Path

In essence, the platform's broadcast subsystem is a chain of open components that transform a bouquet of TV-channels and additional content (such as ESG, EPG and interactivity data) into a DVB-H compliant MPEG transport stream which is then transmitted to Mobile TV clients.

5.1.1 Broadcast Subsystem Overview

The broadcast subsystem constitutes an open end-to-end DVB prototyping platform that consists of the following main functional components:

1. The **content server** is a standard fileserver that hosts looped TV channels and optional unicast multimedia content as MPEG files. Furthermore, it provides access to a set of TV tuners used for feeding live TV into the system.

2. **Media-encoding and playout.** This component manages the conversion of streamed of file-based A/V material into MPEG-4 streams, controlled by an active playlist for each TV channel. It is based on the Apple Darwin[17] streaming server and the VideoLAN client[18].

3. **The data-carousel and ESG** generates additional IP data-streams (carrying e.g., ESG/EPG information, public chat-messages, alerts and other events) by means of an object carousel based on the DVB-H FLUTE standard (ETSI 2004). This added content can be of static nature or is dynamically added by the application server of the interactive platform.

4. The **DVB-H IP-Encapsulator** receives the different A/V- and other data IP-streams and transforms them into a DVB-H broadcast stream via multi-protocol encapsulation (ETSI 2004). This process involves the computation of Forward Error Correction (FEC), additional headers and time-slicing to create a DVB-H compliant MPEG-2 transport-stream optimized for mobile reception (details regarding the encapsulation are discussed in the following subsection).

Finally, the DVB-H MPEG-2 transport-stream is processed by a Dektec DVB-T/H Modulator card and broadcasted via a custom DVB UHF radio frontend.

5.1.2 Software IP-Encapsulation – FATCAPS

As already mentioned, one of the key components of the DVB-H broadcast chain is the IP-Encapsulator which transforms the incoming content and metadata streams into an MPEG-2 transport stream ready for transmission.

In general, the priorities behind our design of the broadcast chain were compliance to DVB standards and maintaining full openness of all components. Consequently, also for the IP-Encapsulator we opted against carrier-grade black box solutions and implemented the required functionality in software. The resulting contribution to the research community is the DVB-H encapsulator FATCAPS (Freakin' Advanced Tremendously Useful enCAPSulator). The software performs all the encapsulation and multiplexing necessary to generate a DVB-H compliant MPEG transport stream (Berger et al., 2008). Its execution platform is a standard PC running the Linux operating system. FATCAPS was developed within AMUSE and put into open source[19].

FATCAPS accepts arbitrary IP/UDP data as input. Hence, it can also handle the FLUTE[20] protocol for reliable file transmission over unidirectional channels, a mandatory requirement in the DVB-H standard (ETSI, 2006). As the latter enforces stringent timing characteristics of the transmitted signal due to its timeslicing feature, the implementation of FATCAPS faced significant challenges concerning ensuring real-time behavior. Fortunately, the flexibility of the Linux operating system and its mature real-time capabilities allowed us to meet the DVB-H standards' requirements and keep burst jitter below 10ms.
Figure 8 shows the basic functionality of FATCAPS at a glance. The Content Server streams arbitrary data to the DVB-H transmitter. For each DVB-H channel, an instance of FATCAPS's *data-aggregator* tool runs on the transmitter box. It collects the data stream received from the Content Server and hands it further on to the *Encapsulator*. This component organizes the data in DVB-H compliant MPE frames, adds several header fields, and optional additional error correction information (MPE-FEC). Subsequently, the *sec2ts* tool splits the generated frames in a stream of MPEG-2 TS packets of 188 Bytes size each. The *Timeslicer* tool takes care of maintaining the time-multiplexing features, a mandatory requirement in DVB-H. It outputs channel bursts at distinct moments in times, adhering the maximum burst jitter discussed above. A multiplexer component finally periodically adds PSI/SI service information before the stream is written to the Dektec modulator for transmission.

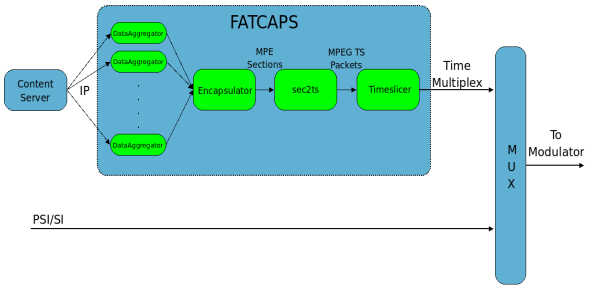


Figure 8: FATCAPS DVB-H Tools Schematic Overview.

5.1.3 Validation – Measurements in the Field

We verified the robustness and versatility of our broadcast tools during a measurement campaign on in-building transmission of DVB-H. During a large trade fair in Vienna, we evaluated critical transmission parameters like burst jitter, forward error correction

[17] Apple Darwin Streaming Server, http://developer.apple.com/opensource/server/streaming/index.html
[18] VideoLAN - VLC media player. http://www.videolan.org
[19] FATCAPs can be freely downloaded under http://amuse.ftw.at/downloads/encapsulator
[20] We used the MAD Project FLUTE implementation from Tampere University of Technology. For documentation and download see http://mad.cs.tut.fi

computation, and the correctness of mandatory DVB-H service information (PSI/SI). Our end-to-end setup proved to function in real-world settings with adverse propagation characteristics (obstructions, reflections) with many sources of transmission errors such as a great variety of obstacles (exhibition stands and visitors moving around). As a key result, we found that a low-cost amplifier with an output power of 1W was sufficient to cover an entire square-shaped 17.000 m² exhibition hall. Our results, which we will report on in a future publication, do not only demonstrate the high robustness of DVB-H transmission, but also the ability of our platform to cover small- to medium-scale testing sites and interoperate with mobile DVB-H standard and measurement receivers.

5.2 Unicast Path

The main purpose of the unicast subsystem is to extend the broadcast Mobile TV path with return-channel interactivity and communication features. The central component is a service framework for prototyping interactive web-applications that provides the 3rd party interfaces and enablers that are necessary for turning standard web-applications into iTV services. These services communicate with the Mobile iTV clients via the wireless UMTS/3G back-channel using different push and pull channels (HTTP, TCP/IP sockets, MMS/SMS). The unicast path consists of the following main elements:

1. The **cellular 3G/UMTS network** provides packet-switched wireless connectivity to the clients as well as mobile network specific communication gateways for messaging and voice telephony. The network also exposes certain interfaces (e.g., for sending/receiving SMS) to the AMUSE service-platform and is usually provided by a Mobile Network Operator[21].

2. The **J2EE Service Framework** has a dual role: as a *framework*, it provides a skeleton of an iTV service to be customized by application developers as well as a set of reusable business objects for standard tasks such as persistence, event-based messaging, server-side push, user-identity and subscription management as well as device-specific UI adaptation and formatting (Johnson, 1997). As a *service platform* running on a Jboss[22] J2EE application server, it manages and provisions these services and components. The platform also hosts the CMP components that coordinate the unicast and the broadcast subsystem, e.g., to let the FLUTE object carousel inject additional data such as EPG-XML fragments or public chat message objects into the DVB-H stream.

3. The **Chat and Presence Server** extends the platform with instant-messaging (IM) and presence which are elementary functions for Mobile Social TV services. It is a representative example of how the platform integrates 3rd Party components and offers them as resource to interactive TV services. The decision for using an external component for messaging, chat and presence functions was made since *interoperability* with existing infrastructures is a strong requirement for Social TV: instant-messaging protocols such as Jabber/XMPP[23] and SIP/SIMPLE[24] have become widely adopted standards and a number of mature, open IM clients and servers exist that are available for integration (Chatterjee, 2005). In the case of the AMUSE platform, an open-source Jabber server[25] is interfaced by the application server so that iTV services can provide chatrooms, buddy-lists and presence-awareness to their users. Moreover, the Jabber server can be accessed by any XMPP-compatible client and federated with other Jabber servers in order to join a larger instant-messaging network. Therefore this approach presents a sensible way to ameliorates one of the currently most problematic issues of Social TV: interoperability with existing communication infrastructures and other Social TV platforms.

5.3 Mobile iTV Terminal Client

The requirement for the AMUSE testbed to support flexible service prototyping and reliable evaluation in field on current smartphones in conjunction with the problem of the availability of only closed, proprietary Mobile TV solutions such as Streamezzo[26] or HisTV[27] has necessitated the development of an custom software client: MiVIBES. MiVIBES[28] is an open Mobile iTV player for Symbian and Linux-based mobiles that integrates streaming video/TV with additional services displayed via multiple side-frames in a split-screen view (Figure 9). For back-channel interactivity, the client uses generic microbrowser components to access and render web-based services

[21] In the context of the AMUSE 2.0 project the consortium partner 'mobilkom austria' provided the cellular network infrastructure, see www.mobilkom.at

[22] The JBoss Foundation, http://www.jboss.org

[23] http://www.jabber.org

[24] http://www.softarmor.com/simple/

[25] Ignite Realtime: Openfire Server, see http://www.igniterealtime.org/projects/openfire

[26] http://www.streamezzo.fr

[27] http://www.histv.org

[28] MiVIBES = Mobile interactive Video Browser Extended Software, see also http://amuse.ftw.at/downloads/aitv-client/MIVIBES

according to the thin-client approach described in Section 4.

Figure 9: iTV Split-screen View on a Nokia E61 Symbian Device based on two Browser- and one Video Panel.

5.3.1 Key Features

Like the AMUSE platform, MiVIBES is designed for prototyping and evaluation of Mobile iTV offering choice different network and service configurations.

Key features of the client are:

1. Tight integration between video, broadcast data and interactive browser components (e.g., clicked links referencing video content are automatically caught and with the media presented in the main window).

2. Flexible switching between different screen layouts (e.g., full-screen, one- and two-panel)

3. Support of the different bearer types (UMTS/WLAN/DVB-H) relevant for video-streaming and interactivity

4. Live DVB-H channels (relayable via WLAN) and simulated offline channels

5. Support for push via TCP/IP sockets, SMS and MMS (as necessary for messaging/chat, event notifications, updates of additional content)

6. High performance and low-level system access enabled by native C++ implementation on Symbian S60 (Nokia E61i, N92, N95) and Mobile Linux (N800, N810 Maemo) devices.

Finally, a general handicap in prototyping new ideas in the context of broadcast Mobile TV is the current lack of open handsets with an API that enables access to the received DVB-H stream. To address this issue, we have developed a basic receiver platform using the Linux-based Nokia N800 Internet Tablet. A standard DVB-T adapter[29] connected via USB enables the N800 to receive and display both, AV content and FLUTE data (Figure 10). Together with the AMUSE 2.0 platform, this setup constitutes the world's first fully open, end-to-end testing platform for DVB-H.

Figure 10: N800 device extended with DVB-H reception.

5.3.2 Client Architecture

The MiVIBES system architecture features the abstraction of the different mobile networking and communication channels available on the device (WLAN, UMTS, SMS/MMS, etc.) as well as a tight integration between media rendering and the user interface (UI), including the microbrowser-based service presentation component (see Figure 11). This approach enables a seamless presentation of iTV services on mobiles as well as flexible adaption to different runtime environments.

Mainly two components in the overview above constitute the player's thin-client player approach, which differentiates MiVIBES from related work: the Interaction Manager and the Service Presentation Manager. The *Interaction Manager* aggregates and routes the different user- or system-events and network connections to components for Service- and A/V-Content Presentation. This tasks also includes processing and dispatching push-events and -content (received e.g. via TCP/IP, SMS or DVB-H/FLUTE) so that e.g., public chat messages are displayed in the correct window.
The *Service Presentation Manager* (SPM) hosts the generic runtime components such as an XHTML microbrowser through which the user accesses the iTV

[29] Note that a standard DVB-T adapter (like the AVerMedia AVerTV DVB-T Volar) is not able to pick up a DVB-H signal using the 4k OFDM mode. However, for most use cases the supported 2k and 8k modes are fully sufficient.

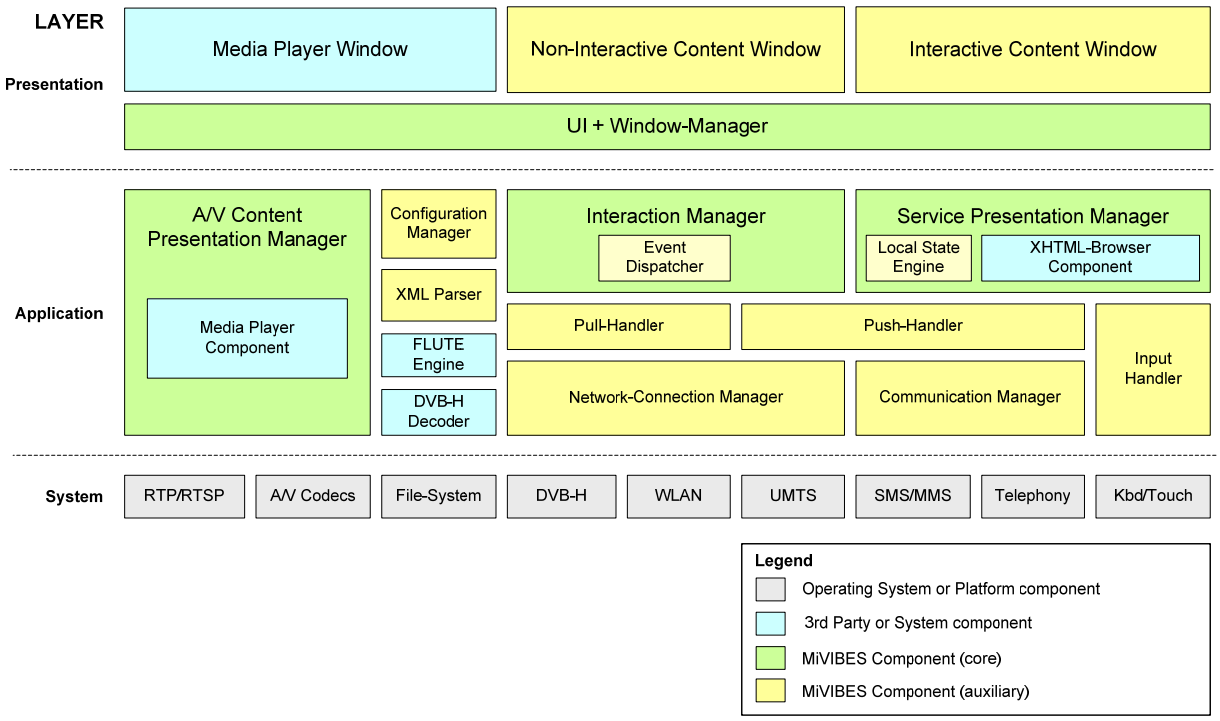


Figure 11: MiVIBES Mobile iTV Client Architecture.

web-applications provisioned on the service-platform. The SPM manages the integration of such components with the player's overall UI so that for example, the click on an URL that references an RTP-stream gets intercepted and then rerouted to the A/V-player. The SPM also abstracts away the hosted components' interfaces, so that additional runtimes (e.g., a SMIL[30] player) can be used to extend the range of supported interactive service types and standards. Furthermore, the SPM uses the microbrowser not only for interfacing and displaying iTV services, but also for local GUI functions such as rendering contextual menus and application settings dialogues.

5.3.3 Bearer Agnostic Network Access

We address the requirement for bearer-agnostic access by using an intermediary All-IP access layer, which allows for flexible acquisition of the video stream via 3G/UMTS, WLAN and DVB-H. This feature makes the streamed media source transparent to application modules located at higher levels. Bearer independence also allows Mobile TV reception even on smartphones without TV reception capability. In order to receive DVB-H streamed content on such clients, we use the following workaround: a small DVB-T enabled notebook operates as gateway and relay between the broadcast air interface and mobile phone. Multiplexed video channels and the additional services including multimedia data of the incoming DVB-H stream are de-multiplexed on the notebook and redistributed to the phone client as separate streams via WLAN, as illustrated in Figure 12 below.

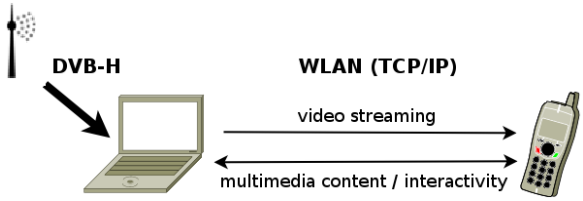


Figure 12: DVB-H via a Notebook as Relay to WLAN.

Utilizing a small laptop still guarantees sufficient mobility of the reception setup. We use a suite of Linux open source tools for analyzing, demultiplexing and relaying the DVB streams to the mobile phone. To decode the DVB video content, we use the open source Video LAN Client (VideoLAN, 2008) because of its versatility and its native support for re-streaming. In addition, we have developed the multicast reflector

[30] Synchronized Multimedia Integration Language, see also http://www.w3.org/AudioVideo/

MCREFLECT[31] that relays a multicast connection (e.g., the video of a specific channel in the DVB-H Stream) to multiple unicast connections. This conversion is necessary, since many clients (such as Symbian phones) only support TCP/IP unicast but not multicast connections. IP Datacast content other than video (such as ESG fragments) is directly passed on to the mobile phone via an additional TCP/IP connection.

6. Research Use Cases

In addition to prototyping interactive Mobile TV and DVB-H measurement campaigns, the AMUSE platform has been successfully applied to the following research use cases:

6.1 Mobile TV User Studies

For the purpose of user studies, the AMUSE test environment allowed us to evaluate Mobile iTV service concepts on behalf of high-fidelity prototypes. On top of the AMUSE platform, we developed the three demo services (Sports, Mobile Social TV, Advertising) described in Section 2.3 as J2EE web-applications. We then set out to evaluate our prototypes with end-users under realistic conditions, a highly common and essential step within R&D projects that investigate new services. This means that applications have to be tested in situ on mobile handsets. To achieve this goal, we used the framework of combined lab and field user studies designed as controlled experiments, allowing in-depth observation of participants' interactions and behaviors (Schatz & Egger, 2008). Test users were asked to engage with the iTV services and features in different contexts such as living room, café or at the bus stop. Thanks to the presence of our mobile DVB-H/WLAN-relay (described in the previous section) that was carried by one observer, participants could test mobile iTV with a variety of Symbian devices (Nokia N92, N95, E61) without any constraints.

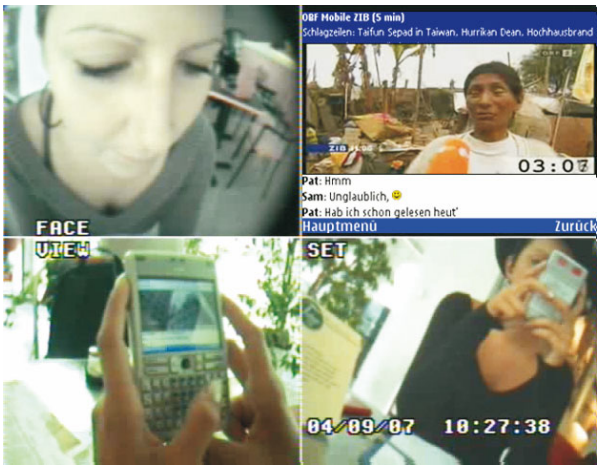


Figure 13: Study Participant in the Café filmed by Cameras mounted on a Hat of our LiLiPUT system.

In addition, we used our custom-designed LiLiPUT (Reichl et al. 2007) system to capture user behavior (see Figure 13). LiLiPUT is a mobile observation system based on a wireless camera-equipped hat that captures the end-user's reactions and immediate context. This is achieved by recording the subject's facial expressions, the mobile device screen status, as well as any sounds involved. As such, LiLiPUT enabled us to gather behavioral data from mobile conditions with the same fidelity as in indoor settings. For further details on our Mobile TV user studies please refer to (Schatz & Egger, 2008).

6.2 EPG Recommender

The second research use case features the extension of the broadcast path and EPG subsystem of the AMUSE platform in order to prototype accelerated access to the 'right' Mobile TV channel by means of a content-based recommender system. The recommender utilizes text mining and NLU (Natural Language Understanding) to extract key features from the EPG metadata of TV shows[32] and match them with user preferences (Bär et al., 2008). Figure 14 shows the interface of the recommender service. The four sliders specify the user's mood preference (action, thrill, fun, erotic) while the text field allows for alternative verbal input of keywords. The result is a ranked list of best matching programs links ordered by a match score.

[31] Further information and download under http://amuse.ftw.at/downloads/dvbh-relay/

[32] For accessing EPG metadata we used the XMLTV system. See also http://www.xmltv.org

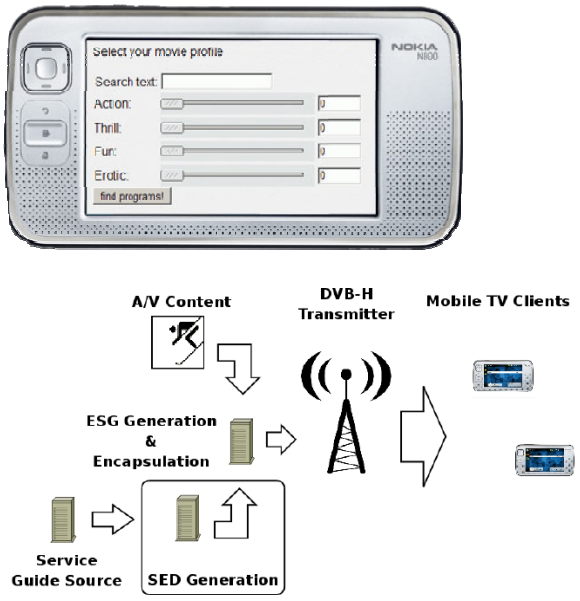


Figure 14: Recommender interface (top) and integration of SEDs in broadcast stream (bottom).

The recommender system is deeply integrated in to the DVB-H broadcast path and features a two-stage processing cycle: SED-generation/transmission and on-device recommendation generation. Firstly, when new content metadata is imported into the system, each EPG metadata entry is preprocessed in terms of extracting recommendation-relevant features (i.e., topics and moods). Each entry is then stored in the AMUSE platform's EPG database which we extended store the feature data as so-called *Semantically Enriched EPG Dataset* (SED). The enriched EPG information itself is delivered as OMA BCAST compliant XML fragments: the programme entries are encapsulated by the complex element 'Content'. We use the child element 'TargetUserProfile' to carry the additional semantic information of the SEDs. The enhanced EPG data is then multiplexed with the video streams by the platform's DVB-H IP-Encapsulator (FATCAPS) and transmitted to the clients. Secondly, upon user request, the MiVIBES player generates a recommendation based on local processing of the SEDs previously received. This approach has several advantages: It only requires a unidirectional broadcast link, avoiding the roundtrip latency of a wireless backchannel. Secondly, the user does not have to perform any registration and profiling steps before using the system. Thirdly, the user's privacy is protected since no personal information leaves the client device. This example demonstrates how the openness and flexibility of the AMUSE platform enables the rapid integration of new features and protocol extensions. For further details on our Mobile TV recommender please refer to (Bär et al., 2008).

7. Conclusion and Outlook

We consider a flexible open-source testbed as a vital enabling platform for the research community for experimentation with hybrid broadcast/unicast architectures and related multimedia services. In this paper we presented our open-source platform that allows for flexible prototyping and evaluation of interactive hybrid multimedia services and technologies.

After an evaluation of existing approaches, we chose an All-IP SOA using standard J2EE components and frameworks for advanced interactivity via unicast as well for mediation platform functions. For the broadcast path we used a suite of open source tools for encoding and streaming in conjunction with standard DVB-H/T equipment, with an optional bypass of the DVB-H air interface. This approach has proven economic, bearer-agnostic and flexible enough for research purposes. On the client side, a major result is the integration of video display with browser functionality to realize a unified rich media player for the S60 mobile platform as well as Mobile Linux. This architecture allows for rapid prototyping of mobile iTV concepts by shifting business logic and presentation authoring from the client to the server.

The AMUSE platform and service framework enables any research group to become a digital TV broadcast operator *en miniature* without having to strain their project budgets with the procurement of carrier-grade equipment. Even more important, due to the openness of all components, the platform serves as a perfect tool for research purposes. It lays the foundation for prototyping new broadcast/unicast services such as interactive advertisement and program recommender systems. Furthermore, the AMUSE platform and service framework can be used to support experimentation on the lower layers of the OSI stack, for instance for the development of new error correction schemes, improvements on channel coding algorithms, statistical multiplexing (cf. Jacobs et al., 2008) as well as the investigation of synchronization of AV content with supplementary data (cf. Leroux et al., 2007). Plenty of ideas for future improvements of our open source components exist and we cordially invite the community to work with us on them together.

8. Acknowledgement

This research has been performed within the projects M2 AMUSE 2.0, U0 and N0 at the Telecommunications Research Center Vienna (ftw.) supported by the Austrian Government and by the City of Vienna within the competence center program COMET. We would like to thank our AMUSE 2.0 team members Sebastian Egger, Arian Bär, Thomas Ebner, Erwin Wittowetz and Siegfried Wagner for their hard and fruitful work.

9. References

Baldzer, J., Thieme, S., Boll, S., Appelrath, H.-J. and Rosenhager, N. (2005) Night Scene Live: A Multimedia Application for Mobile Revellers on the Basis of a Hybrid Network, Using DVB-H and IP Datacast. *IEEE International Conference on Multimedia and Expo ICME'05*, 6-6 July 2005, pp.1567-1570.

Baier, A. and Richartz, M. (2006) Mobile TV - From pure Broadcast to Interactivity. *Workshop on Multiradio Multimedia Communications MMC'06*, Berlin, 19-20 Oct, 2006.

Bär, A., Berger, A., Egger, S. and Schatz, R. (2008) A Lightweight Mobile TV Recommender: Towards a One-Click-to-Watch Experience. In Proc. of the *6th EuroITV Conference on Interactive Television*, Salzburg, Austria, July 3-4, 2008.

Berger, A., Pallara, L. and Schatz, R. (2008) An Open Source Software Framework for DVB-* Transmission. *ACM International Conference on Multimedia*, Vancouver, BC, Canada, 27-31Oct, 2008.

Blom, S., Book, M., Gruhn, V., Hrushchak, R. and Kohler, A. (2008) Write Once, Run Anywhere A Survey of Mobile Runtime Environments. The *3rd International Conference on Grid and Pervasive Computing, GPC Workshops'08*. pp.132-137, 25-28 May 2008.

Cazoulat, R. and Lebris, T. (2008) An efficient multimedia system for J2ME mobile device. *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (ISBMSB)*, pp.1-5, March 31 - April 2, 2008.

Chatterjee, S., Abhichandani, T., Haiqing Li, TuIu, B. and Jongbok, B. (2005) Instant messaging and presence technologies for college campuses. *IEEE Network*, vol.19, no.3, pp. 4-13, May-June 2005.

Coppens, T., Trappeniers, L. and Godon, M. (2004) Amigo TV: towards a social TV experience. *2nd European EuroITV Conference on Interactive Television*: Enhancing the Experience, March 31-April 2, 2004, Brighton, UK.

Coulton, P., Rashid, O., Edwards, R., Thompson, R. (2005) Creating entertainment applications for cellular phones. *Computer and Entertainment*, 3(3).

Datamonitor (2006) Technology developments in the European digital TV Sector, London, 7 July 2006. http://www.datamonitor.com/~1ec12699c5a6421c85765060fbaeccd1~/industries/research/?pid=BFTC1363&type=Brief

Dufourd, J.C., Avaro, O. and Concolato, C. (2005) An MPEG Standard for Rich Media Services, *IEEE MultiMedia*, 12(4), 60-68.

ETSI (2004) Digital Video Broadcasting; Transmission System for Handheld Terminals (DVB-H), ETSI standard, EN 302 304 V 1. I. 1, 2004.

ETSI (2006) Digital Video Broadcasting; IP Datacast over DVB-H: Electronic Service Guide (ESG), ETSI standard, ETSI TS 102 471 V1.1.1, 2006.

Ftw (2007) AMUSE 2.0 project site, http://amuse.ftw.at

Garret, J.J. (2005) Ajax: A New Approach to Web Applications. February 18, 2005, http://www.adaptivepath.com/publications/essays/archives/000385.php

Greenhalgh, C., Benford, S., Drozd, A., Flintham, M., Hampshire, A., Oppermann, L., Smith, K., and von Tycowicz, C. (2007) Addressing mobile phone diversity in ubicomp experience development. In L.J. Krumm, G. D. Abowd, A. Seneviratne, & T. Strang (Eds.) *Ubicomp*, LNCS vol. 4717, pp. 447-464. Springer, Berlin-Heidelberg.

Hartl, M., Rauch, C., Sattler, C., Baier, A. (2005) Trial of a Hybrid DVB-H / GSM Mobile Broadcast System. *14th IST Mobile and Wireless Communications Summit*, Dresden, June 2005.

Högg, R., Martignoni, R. and Stanoevska-Slabeva, K. (2007) The impact of interactivity on mobile broadcasting value chains, *16th IST Mobile & Wireless Communications Summit*, 1-5 July 2007, Budapest, Hungary.

Huebscher, M., Pryce, N., Dulay, N., Thompson, P. (2006) Issues in developing ubicomp applications on Symbian phones. In: Proc. *Future Mobile Computing Applications, International Workshop on System Support*, 17 Sept 2006, pp. 51–56

Jacobs, M., Barbarien, J., Tondeur, S., Van de Walle, R., Paridaens, T. and Schelkens, P. (2008) Statistical multiplexing using SVC. *IEEtE International Symposium on Broadband Multimedia Systems and Broadcasting (ISBMSB)*, April 6-7 2006, Las Vegas, USA.

Jensen, J.F. (2005) Interactive Television: New Genres, New Format, New Content. In Proc. of the *Second Australasian Conference on Interactive Entertainment 2005*, ACM International Conference Proceeding Series, 23-25 Nov 2005, Volume 123, Sydney, Australia, pp. 89-96.

Johnson, R.E. (1997) Frameworks = (components + patterns). *Communications of the ACM*, 40(10), Oct 1997, 39-42.

Klinkenberg, F. and Steckel, P. (2006) A Modular Approach for a Java-based Service Framework in Hybrid Networks,

IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (ISBMSB), April 6-7 2006, Las Vegas, USA.

Knoche, H. and McCarthy, J. D. (2005) Design requirements for mobile TV. *7th international Conference on Human Computer interaction with Mobile Devices & Services (MobileHCI'05)*, Salzburg, Austria, 19-22 Sept 2005, vol. 111. ACM, New York

Kumar, A. (2007) *Mobile TV: DVB-H, DMB, 3G Systems and Rich Media Applications*. Focal Press Media Technology Professional, ISBN: 9780240809465.

Leroux, P., Verstraete, V., De Turck, F. and Demeester, P. (2007) Synchronized Interactive Services for Mobile Devices over IPDC/DVB-H and UMTS. *2nd IEEE/IFIP International Workshop on Broadband Convergence Networks (BcN'07)*, 21-21 May, 2007.

Liebermann, L. (2007) Mobile Ajax and Application Adaptation. Paper at the *W3C/Open Ajax Alliance Workshop on Mobile Ajax*, September 2007, Silicon Valley, California, USA.

Martinetz, G. (2006) DVB/CBMS and OMA/Bcast CBMS *Standardisation. Workshop on Multiradio Multimedia Communications MMC'06*, Berlin, Oct 19-20, 2006.

Ofcom (2006) The Communications Market 2006. http://www.ofcom.org.uk/research/cm/cm06/main.pdf

Ollikainen, V. and Peng, C. (2006) A Handover Approach to DVB-H Services. IEEE International Conference on Multimedia & Expo (ICME), July 9-12 2006, Toronto, Canada.

Ollikainen, V. (2008) Mobile TV Should Be More Than a Television, The final report of the Podracing project. VTT Research Notes 2439, Espoo, Finland.

Paila, T. Luby, M. Lehtonen, R. Roca, V and Walsh, R. (2004) FLUTE - File Delivery over Unidirectional Transports, IETF RFC 3926.

Petrovic, O., Fallenböck, M., Kittl, Ch. and Langl, A. (2006) Mobile TV in Österreich. Schriftenreihe der Rundfunk und Telekom Regulierungs-GmbH (RTR-GmbH) Österreich, Band 2/2006.

Reichl, P., Froehlich, P., Baillie, L., Schatz, R. and Dantcheva, A. (2007) The LiLiPUT Prototype: A Wearable Lab Environment for User Tests of Mobile Telecommunication Applications. Extended Abstracts of CHI2007, Conference on Human Factors in Computing Systems, San José, CA, USA.

Roibas, A.C. (2004) Ubiquitous media at the intersection: iTV meets mobile communications. In Proceedings of HCI 2003, Bath. BCS conference series, 2. Springer-Verlag, London, UK. ISBN 1852337664.

Schatz, R., Jordan, N. and Wagner, S. (2007a) Beyond Broadcast – A Hybrid Testbed for Mobile TV 2.0 Services. The 6th International Conference on Networking (ICN'07), Martinique, April, 2007. IEEE Computer Society, Washington, DC.

Schatz, R., Wagner, S., Egger, S. and Jordan, N. (2007b) Mobile TV becomes Social - Integrating Content with Communications. Proc. of The International Conference on Information Technology Interfaces (ITI'07), Dubrovnik, Croatia, June, 2007. IEEE, Washington, DC.

Schatz, R. and Egger, S. (2008) Social Interaction Features for Mobile TV Services. In Proc. of *IEEE Broadband Multimedia Systems and Broadcast Symposium*, April 2008, Las Vegas, USA.

Setlur, V. Capin, T. Chitturi, S. Vedantham, R. and Ingrassia, M. (2006) MORE: Mobile Open Rich-Media Environment. *IEEE International Conference on Multimedia and Expo (ICME'06)*, Toronto, Ontario, Canada.

Skrodzki, S. (2006) HisTV Air Interface Draft Version 1.3. GMIT GmbH, 07. Nov 2006, http://www.histv.org

Sullivan, B. and Connors A. (2008) Mobile Web Application Best Practices. W3C Working Draft 29 July 2008, http://www.w3.org/TR/2008/WD-mwabp-20080729

Trefzger, J. (2005) Mobile TV-Launch in Germany – Challenges and Implications. *Working paper of the Institut für Runfunkökonomie*, Universität Köln, Nov 2005, ISBN 3-938933-07-0.

UMTS Forum (2006) Joint Mobile TV Group Whitepaper – Mobile TV: The Groundbreaking Dimension. The UMTS-Forum, Nov 2006, http://www.umts-forum.org

VideoLAN (2008) VideoLAN project homepage, http://www.videolan.org

Zucker, D.F., Uematsu, M. and Kamado, T. (2005) Content and Web services converge: a unified user interface. *IEEE Pervasive Computing*, 4(4), pp. 8-11, Oct.-Dec. 2005

HTTP over Bluetooth: a J2ME experience *

Vincenzo Auletta, Carlo Blundo
Dipartimento di Informatica ed Applicazioni
Università degli Studi di Salerno
I-84084 Fisciano (SA) - Italy
{auletta,carblu}@dia.unisa.it

Emiliano De Cristofaro
Information and Computer Science
University of California Irvine
Irvine, CA, 92617 - USA
edecrist@uci.edu

Abstract— **Over the last years, computation and networking have been increasingly embedded into the environment. This tendency has been often referred to as pervasive or ubiquitous computing, to remark the aim to a dense and widespread interaction among computing devices. User intervention and awareness are discarded, in opposition to an automatic adaptation of applications to location and context. To this aim, much attention is drawn to technologies supporting dynamicity and mobility over small devices which can follow the user anytime, anywhere.**

The Bluetooth standard particularly fits this idea, by providing a versatile and flexible wireless network technology with low power consumption. Operating in a license-free frequency, users are neither charged for accessing the network nor they need an account with any company. Bluetooth dynamically sets up and manages evolving networks, by providing the possibility of automatically discovering devices and services within its transmission range.

Research studies have forecasted that within a few years, most of the devices accessing the Web will be mobile, and presumably most of them will be Bluetooth-enabled. Therefore, we need solutions that encompass networking, systems, and application issues involved in realizing mobile and ubiquitous access to services.

In this paper, we present a lightweight solution to extend the possibility of accessing Web resources also from Bluetooth-enabled mobile phones. All the implementation details will be hidden both to users and to application developers, allowing an easy and complete portability of applications working on traditional TCP/IP communication protocols towards the Bluetooth technology.

Index Terms— **Ubiquitous Computing, HTTP, Bluetooth, Mobile Phones, J2ME.**

I. Introduction

The evolution of technology has led to a deep transformation of users habits, with an increasing requirement of support for mobility and connectivity. Furthermore, nowadays a brand new set of applications fit mobile environments and allow device interactions over wireless channels. Today's mobile phones are small, powerful, and usable enough to be fundamental working instruments, and to be considered for the deployment of complex applications.

Modern applications, however, require connectivity and thus a critical issue for the diffusion of mobile devices is the capacity to run network applications, especially Web applications. In the last years, several new protocols have been presented for wireless communications, such as IRDA, WLAN, and GPRS/UMTS. However, IRDA connections are limited to two devices with a direct line of sight, and thus IRDA is not practically useful for a real intercommunication scheme. WLAN instead has been designed as a powerful technology to support multipoint connections, but diffusion of WLAN on mobile devices and particularly on mobile phones is still low. GPRS/UMTS are widely supported but they provide connectivity at modest speed and requires a personal account with a phone company. At the same time, we witnessed the growth of Bluetooth, that is a low-cost, robust, powerful, and flexible short-range wireless network technology with low power consumption [4]. It operates in a license-free frequency range, so that user is not charged for accessing the network nor needs an account with any company, thus allowing a relevant decrease of communication costs. Nowadays, the evolution of Bluetooth technology is driven by the Bluetooth SIG, that consists of over 7000 member companies that guarantee a large support to this technology. In fact, Bluetooth technology is used in many widespread different devices, such as handhelds, mobile phones, smartphones, laptops, PDAs. A thorough overview on Bluetooth is given in [6] and [24].

A recent study has pointed out that the number of users accessing the web from a mobile device has overtaken the one using a "standard" terminal [9]. Therefore, we need solutions that encompass networking, systems, and application issues involved in realizing mobile and ubiquitous access to services.

In this paper, we analyze how to extend the possibility of accessing Web resources from Bluetooth-enabled mobile phones.

Our goal is to provide a transparent middleware which allows user to access Web resources by using a Bluetooth connection. In particular, we provide application developers with a lightweight solution to let their mobile applications establish HTTP connections over a Bluetooth channel. In this way, the cost of communication is brought to zero, and the power-consumption is kept low. Furthermore, we have in mind a transparent middleware allowing programmers to ignore the implementation details related to the underlying Bluetooth channel.

Common applications massively using HTTP connections are Web browsers, e.g. Opera Mini [18] – the Web browsed released by Opera Software [19]. Being developed in Java, the Opera Mini browser is platform independent and can be easily deployed on every J2ME-powered mobile phone. Whenever a user wants to surf the Internet, the application instaurates a HTTP connection over WAP, GPRS, or UMTS, which are the available protocols supporting TCP/IP. Provided

*A preliminary version of this work has been published in ICSNC 2007 [23]

that a Bluetooth connection is available to a device acting as a gateway to the Internet, our transparent middleware would allow Opera Software to release a version of Opera Mini which works on the free Bluetooth communication channel, without refactoring the source code.

Paper Organization. The rest of the paper is organized as follows. In Section II, we present the endorsed technologies, i.e. Bluetooth, J2ME, and the JSR-82 APIs. In Section III, we give an overview of our solution to allow J2ME application developers to establish HTTP connections using Bluetooth as the communication channel. Then, Sections IV and V present the details of our implementation respectively for the client-side and the server-side. Subsequently, Section VI discusses the transparency of our solution and presents some application scenarios. Finally, Section VII briefly evaluates the performance overhead.

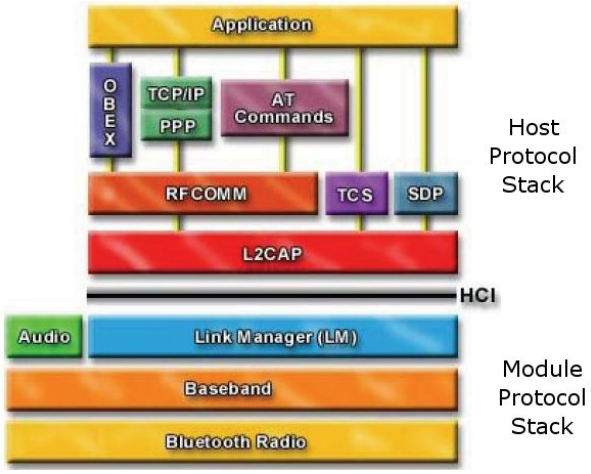


Fig. 1. The Bluetooth Stack [26].

II. Endorsed Technologies

In this section, we present all the technologies on which our work relies: the Bluetooth Standard [4] and the J2ME [11]. The former defines details for the communication between devices, while the latter describes how to write Java applications on mobile devices. Then, we give an overview of the networking management within J2ME. Finally, we present the API needed to use Bluetooth within J2ME, defined by the JSR-82 standard [15].

A. The Bluetooth Wireless technology

Bluetooth specification was introduced in 1994 by Ericsson to provide radio communications between mobile phones, headsets and keyboards. The specifications were then formalized by the Bluetooth Special Interest Group (SIG) [4] in 1998. Within this technology, radio communications can take place by means of integrated and cheap devices with small energy consumption. This is achieved by embedding tiny, inexpensive, short-range transceivers into electronic devices that are available nowadays. The Bluetooth standard defines the following requirements:

- The system must operate globally, and the required frequency band must be license-free and open to any radio system.
- The system must provide peer connections.
- The connection must support both voice and data.
- The radio transceiver must be small and operate at low power.

Bluetooth devices operate in a license-free frequency range (starting from 2,4 GHz). The available bandwidth is divided into 79 channels. In version 1.2 one can establish a 1 Mbps link (a 2 Mbps link is supported by Version 2.0) [6]. Moreover, security and error support allow to assure efficient and reliable connections even in environments with a strong presence of interferences and electromagnetic fields.

Bluetooth-enabled devices can dynamically *discover* other devices in their range and their supported services, through an inquiry process. A *Piconet*, consisting of one *master* device and up to seven *slave* devices, will be settled once the peer connections have been established. Piconets can be interconnected to form a *Scatternet*.

An overview of the Bluetooth stack is presented in Figure 1. The *radio* level is the lowest one and defines the technical details of the communication. Bluetooth adopts the FHSS (Frequency Hopping Spread Spectrum), making 1600 hops per second; thus each physical channel is occupied for 625μs. These intervals are referred to as slots and they are numbered sequentially. Frequency hopping occurs by jumping from one physical channel to another in a pseudorandom sequence. The *baseband* layer handles channels and physical links, providing services such as error correction and security. It supports multipoint communications through FH/TDMA (Frequency Hopping/Time Division Multiple Access). The master device is in charge of defining the hopping sequence to all the slave devices. A physical channel is shared between the master and a slave using a time division scheme in which data are transmitted in one direction at time, with transmissions alternating between the two directions: the master transmits in even slots; slaves transmit in odd slots and may hold the transmission for 1, 3, or 5 consecutive slots. Links between master and slaves can be either *Synchronous Connection Oriented* (SCO) or *Asynchronous Connection-Less* (ACL). The first type of link is used in real-time applications and it allows a symmetric point-to-point communication achieving 64 Kbps transmission rate. The second is used for asymmetric transmission between master and slaves with 723 Kbps for downlink and 57 Kbps for uplink. Bluetooth standard defines two different types of packets for ACL links: Data Medium Rate (DM) which provides a 2/3 FEC Hamming code (i.e., error correcting capabilities), and Data High Rate (DH) which provides no FEC coding (i.e., no error correction at all). Therefore, we have six different data packets according to the slots assignment and data encoding: DH1, DH3, DH5, DM1, DM3, and DM5 (digits denote the number of occupied slots). Up in the stack we find: the *Link Management Protocol* (LMP) handling link setup, authentication, and link configuration; the *Host Controller Interface* (HCI) which provides a uniform

method of accessing the Bluetooth baseband capabilities; the *Logical Link Control and Adaptation Protocol* (L2CAP) which deals with data multiplexing and segmentation. Finally, on top of L2CAP, we find several data communication protocols. The main protocols are:

- SDP (Service Discovery Protocol), which handles the discovery of devices and services within the device's transmission range.
- RFCOMM, which implements emulation of serial connections, setting up point-to-point connections. It supports framing and multiplexing and achieves all the required functions for serial data exchange.
- OBEX (Object Exchange), which is built on the top of RFCOMM to implement exchange of objects, such as files and vCards. Originally, it was developed by IrDA (Infrared Data Association) for IR-enabled devices.
- TCS (Telephony Control protocol Specification), which defines ways to send audio calls between Bluetooth devices.

The Bluetooth technology is also composed by a set of profiles. Bluetooth profiles describe several scenarios where Bluetooth technology is responsible of transmission. Each scenario is described by a user model and the corresponding profile gives a standard interface that applications can use to interact with the Bluetooth protocols. The profile concept is used to decrease the risk of interoperability problems between different manufacturers' products, for instance, some profiles are:

- FTP (File Transfer Profile), which defines how folders and files on a server device can be browsed by a client device.
- HPF (Hands-Free Profile), which describes how a gateway device can be used to place and receive calls for a hand-free device.
- VDP (Video Distribution Profile), which defines how a Bluetooth enabled device streams video over Bluetooth wireless technology.

Other details on the Bluetooth specification can be found in [4] or in [27].

In order to interface applications to the physical layer a Bluetooth Stack implementation is necessary. The stack provides a standard interface between the application layer and the Bluetooth specification. This interface is used to overcome the compatibility problems between application and different Bluetooth devices. Indeed, Bluetooth stacks are responsible of implementing the Bluetooth wireless standards specifications. There are several different stacks targeted to different devices, applications, and operating systems. To our knowledge, currently available Bluetooth stack implementations are:

- Mobile devices vendors' embedded stacks. Vendors providing Bluetooth-enabled devices have to build their own Bluetooth stack; for mobile phones stack implementations depend on the OS (e.g., Symbian).
- Broadcom BTW (not free) [8]. It is addressed to PC OEMs and accessory manufactures to quickly and easily add Bluetooth technology to desktop PC and notebooks running Windows. It includes the object code for the Protocol Stack (L2CAP, SDP, RFCOMM, OBEX, PPP, BTM-Bluetooth Manager), an application programming interfaces (APIs), and test tools.
- Microsoft BT Stack [20]. It is the Microsoft version of the Bluetooth stack and is embedded in Windows XP SP 2. It provides the support for most of Bluetooth profiles, essentially the ones based on the RFCOMM protocol.
- BlueZ (free and open-source) [7]. It is the official Linux Bluetooth Stack. The code is licensed under the GNU General Public License and is included in the Linux 2.4 and Linux 2.6 kernel series. It provides a direct access to the transmission layer and allows developers to set several parameters of the communication (i.e., choosing an ACL or SCO connection, choosing different time shifting, etc.).

B. J2ME

The J2ME (Java Platform Micro Edition) is a collection of Java APIs supporting the development of applications targeted to resource-constrained devices such as PDAs and mobile phones. Formally, J2ME is an abstract specification, however the term is frequently used also to refer to the runtime implementations. The advantages of using Java as programming language are the code portability and an increase of mobile devices' flexibility. In particular, J2ME provides support for deploying dedicated applications, named MIDlets. Since the range of micro devices is so diversified and wide, J2ME was designed as a collection of configurations, where each configuration is tailored to a class of devices. Each configuration consists of a Java Virtual Machine and a collection of classes that provide a programming environment for the applications. Configurations are then completed by profiles, which add classes to provide additional features suitable to a particular set of devices. J2ME defines two configurations: the *Connected Device Configuration* (CDC) [12] and the *Connected Limited Device Configuration* (CLDC) [13].

CDC is addressed to small, resource-constrained devices such as TV set-top boxes, auto telematics. It can add a graphical user interface and other functionalities; CLDC, instead, is addressed to devices with limited memory capacity. In this paper, we restrict our attention to the CLDC configuration, which is the most diffused one. CLDC is a low level specification that includes a set of APIs providing basic features for resource-constrained devices, such as mobile phones and PDAs. Producers should add features to CLDC by providing new libraries and thus creating a Profile. The first profile proposed for CLDC was the MIDP (Mobile Information Device Profile) [14]. MIDP is a set of Java libraries that allows to create an application environment for mobile devices with limited resources. Here, limitations include: amount of available memory, computational power, network communications with strong latency, and low bandwidth. MIDP 1.0 specification was produced by MIDPEG (MIDP Expert Group), as part of the JSR-37 [17] standardization effort; while, the MIDP 2.0 specification was released with the JSR-118 [16] standardization effort. MIDP 2.0 devices have to meet the following requirements:

- *Memory*, 250 KB of non volatile memory for MIDP components, 8 KB for user data.

- *Display*, 96x54 resolution, 1-bit color depth, 1:1 aspect ratio.
- *Networking*, bidirectional and wireless communication, limited bandwidth.

C. Networking in J2ME

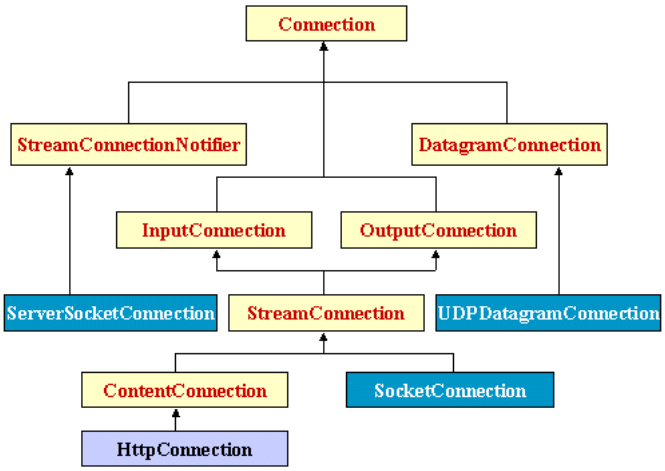


Fig. 2. The J2ME Connection hierarchy diagram.

The J2ME has to support a large variety of mobile devices with different sizes and shapes, different networking capabilities, and I/O requirements. Therefore, networking management in J2ME should be both flexible and device specific. To this aim, the CLDC defines the Generic Connection Framework. Such a framework delineates the abstractions of the networking and file I/O, in order to support the largest variety of devices, while leaving device manufactures to provide real implementations. Abstractions are defined as Java interfaces and the device manufacturers choose which one to implement.

Networking features within J2ME are defined by the MIDP in the `javax.microedition.io` package. It supports the following communication protocols:

- HTTP and HTTPS connections
- datagram
- socket
- secure socket
- serial port communication

Figure 2 shows the interface diagram of the `javax.microedition.io.Connection` hierarchy. Those interfaces are part of the Generic Connection Framework of J2ME's CLDC, together with the `Connector`) class. We remark that no implementation is given at the CLDC level. The actual implementation is left to MIDP. The `Connector` class is the core of the Generic Connection Framework. All connections are created by its static method `Connection open(String connect)`. Different connections are instantiated according to different connect strings. The connect string has a URL-like format:

PROTOCOL:[//TARGET][PARAMS],

where PROTOCOL defines the type of connection (e.g., file, socket, http); TARGET defines a hostname, a port number, or a file name; PARAMS defines optional parameters. Polymorphically, different parameters in the connection string make the `Connector.open` method return a different `Connection` object. For example, a connection string starting with *http://* will drive the `open` method to return a `HttpConnection` object.

We remark that MIDP-powered devices are required to support at least HTTP connections. HTTP is the most used protocol and it is easily implemented over different wireless networks. The use of HTTP allows user to exploit server-side infrastructure which are available for cabled networks. The `HttpConnection` interface defines the MIDP API for HTTP. This interface extends another interface, `ContentConnection`, to add fields and methods required for: URL parsing, request management, response parsing.

HTTP connection parameters can be set up by the following methods:

- `setRequestMethod(String method)`: chooses GET, POST, OR HEAD operations.
- `setRequestProperty(String key, String value)`: sets up a generic request.

In Figure IV, we show an example of how to execute from a MIDP-powered device a simple HTTP post operation to a Java Servlet on a remote Web Server . We remark that the operating system is in charge of establishing a physical connection. If more network interfaces are available, it selects a default one or asks user to choose one.

The Java code performs the following operations:

(1) Open a HTTP connection with the Web Server for both send and receive operations.
(2) Set the request method to POST
(3-6) Send the string entered by user byte by byte.
(7-8) Close the output stream.
(9) Open an InputStream on the connection.
(10-18) Retrieve the response back from the servlet.
(19) Close the input stream.

D. JSR-82

Although the synergy between MIDP and J2ME technologies supplies a large number of communication schemes, it does not provide support for the Bluetooth technology. Therefore, the Java Expert Group JSR-82 [15] introduced the *Java API for Bluetooth Wireless Technology* (JABWT) that provides a standard and high-level support for handling Bluetooth communications in Java applications. This API operates on top of CLDC to extend MIDP functionalities. Its development is still in progress, but about twenty mobile vendors have adopted it in their devices. The last released version (Version 1.1) provides support for:

- Data transmission on the Bluetooth channel (audio and video are not supported).
- Protocols: L2CAP, RFCOMM, SDP, OBEX.
- Profiles: GAP, SDAP, SPP, GOEP.

The Generic Access Profile (GAP) defines the generic procedures related to discovery of Bluetooth devices and link management aspects of connecting to Bluetooth devices. The

```
(1) HttpConnection hc = (HttpConnection) Connector.open(defaultURL, Connector.READ_WRITE);

(2) hc.setRequestMethod(HttpConnection.POST);

(3) DataOutputStream dos = hc.openDataOutputStream();
(4) byte[] request_body = requeststring.getBytes();

(5) for (int i = 0; i < request_body.length; i++)
(6)            dos.writeByte(request_body[i]);

(7) dos.flush();
(8) dos.close();

(9) DataInputStream dis = new DataInputStream(hc.openInputStream());

(10) int ch;
(11) long len = hc.getLength();

(12) if(len!=-1) {
(13)     for(int i = 0;i<len;i++)
(14)         if((ch = dis.read())!=-1)
(15)             messagebuffer.append((char)ch);
(16) } else { // if the content-length is not available
(17)     while ((ch = dis.read()) != -1)
(18)         messagebuffer.append((char) ch);
}

(19) dis.close();
```

Fig. 3. A simple HTTP post operation from a MIDlet

Service Discovery Application Profile (SDAP) defines the features and procedures for an application in a Bluetooth device to discover services registered in other Bluetooth devices and retrieve any desired available information pertinent to these services. The Serial Port Profile (SPP) defines the requirements for Bluetooth devices necessary for setting up emulated serial cable connections using RFCOMM between two peer devices. The Generic Object Exchange Profile (GOEP) defines the requirements for Bluetooth devices necessary for the support of the object exchange usage models.

The interaction between the J2ME environment and the Bluetooth API is shown in Figure 4.

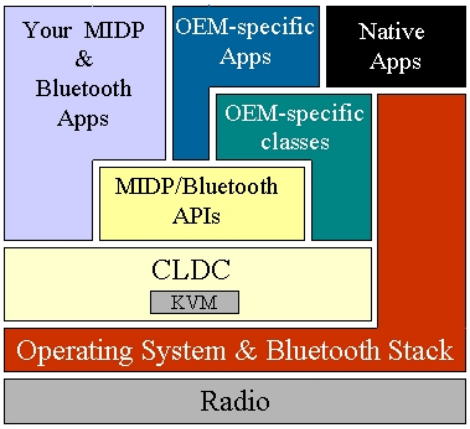


Fig. 4. J2ME - Bluetooth API interaction architecture [25].

Using JABWT, it is possible to interact with the Bluetooth stack in a Java application. In particular, it is possible to call services such as device and service discovery, establishment of RFCOMM, L2CAP, and OBEX connections.

In order to use the Java APIs for Bluetooth, a real implementation of the JSR-82 specification is necessary on the device. To our knowledge, the current JSR-82 implementations are:

- Mobile devices vendors' embedded JSR-82 implementations.
- Atinav aveLink suite (not free) [5]. It offers both a standard implementation of the Bluetooth stack and the implementation of all the standard profiles for ANSI C, JSR-82 for J2SE Java, JSR-82 for J2ME, Windows and Windows CE.
- Impronto Rococo (not free) [10]. It is a complete product that provides the Bluetooth Stack and the integration layer, the JVM and the JSR-82 implementation layer both for J2SE and J2ME.
- Avetana (not free) [2]. It enables writing J2SE applications to access the Bluetooth layer; it is available for Windows, MacOS X, and Linux platforms.
- BlueCove (free) [3]. It provides the Java JSR-82 support for J2SE applications over the Windows XP SP2 Bluetooth stack.

III. HTTP Connections over Bluetooth

In this section, we will present our solution to allow a J2ME application developer to establish HTTP connections using Bluetooth as the communication channel.

Since a few years, Bluetooth has been exploited to connect a PC to the Internet through Internet connections available on a paired mobile phone, such as cell network service (e.g. GPRS, EDGE, UMTS) or WLAN. In this scenario, the mobile phone acts as a gateway, while the communications between the phone and the PC are carried over Bluetooth.

However, the cell network internet service is often expensive or not always available. Also, mobile phones supporting WLAN are still in a minority of the market and are available only on high-end cost phones. In this work, we want to provide HTTP connections to mobile phone users without requiring them to use the cell network Internet connection, i.e. using a Bluetooth connection.

Therefore, we refer to the scenario shown in Figure 5, where a client establishes a HTTP connection with a server.

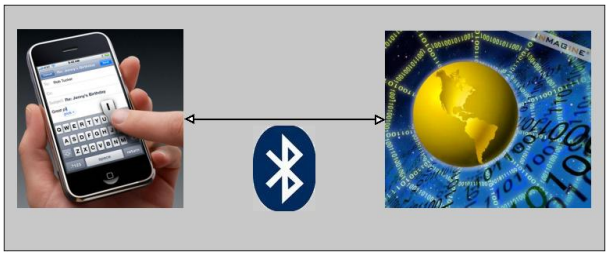


Fig. 5. Ideal scenario: a mobile phone accesses the Web using Bluetooth.

We observe that wireless communication by means of HTTP over GPRS/EDGE/UMTS and WLAN is widely supported within J2ME. On these channels it is easy to create a HTTP connection and deploy MIDlets that network over this channel by means of `HttpConnection` objects. However, to the best of our knowledge, there is no implemented support for creating a `HttpConnection` object which uses an underlying Bluetooth channel. To overcome this limitation, we created ourselves the BtHttpConnection class and introduced a new entity, the `BHSP` (Bluetooth Http Server Proxy), that takes care of interfacing clients to the Web Server. We argue that we can maintain the same server-side architecture and guarantee the interoperability of applications running on the mobile device with any Web Server. Moreover, no modification to the MIDlet is required to use Bluetooth as communication channel, instead of the previously supported channels, i.e. WLAN or GPRS/EDGE/UMTS. Figure 6 illustrates the operational scenario to which we refer.

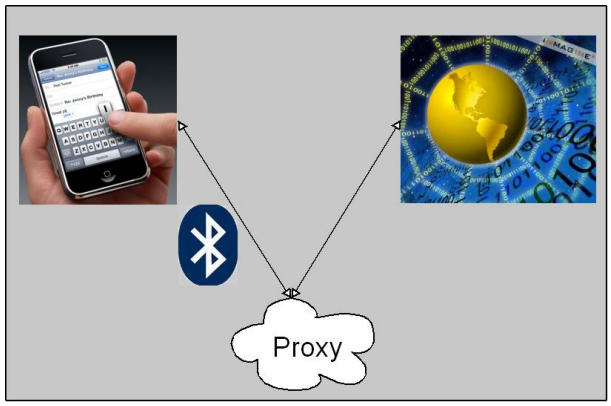


Fig. 6. Operational scenario of our solution.

The resulting framework is then composed of four different entities:

- Client Application. It runs on a Bluetooth-enabled mobile device and it establishes HTTP connections on a Bluetooth channel using the J2ME standard way, i.e. using the `HttpConnection` class.
- `BtHttpConnection` interface performs all the work required to communicate on the Bluetooth channel and to achieved the transparency for the client application's developer.
- `BHSP` - Bluetooth Http Server Proxy. It interfaces clients with Web Servers, by listening to requests on the Bluetooth channel, forwarding them to the Web Server, and giving back results to the client application.
- Web Server. It is a standard Web Server (e.g. Apache) that replies to clients' requests communicating through the `BHSP`.

We remark that the `BHSP` has been designed as a supplementary module of the Web Server (i.e., it is a daemon starting together with the Web Server), conceptually allowing the Web Server to accept requests from a Bluetooth channel, too. Within this scenario, a Web Server administrator can decide to provide its resources not only through the Internet, but also to Bluetooth-enabled mobile devices which are within its transmission range. This Web server will be listening upon port 80 (the standard HTTP port) and upon Bluetooth RFCOMM port, see Figure 7(a). Moreover, our solution fits another scenario as well. Indeed, the `BHSP` can act as a real proxy and be implemented over any device equipped with two interfaces: (i) Bluetooth, used to interact with the client application, and, (ii) an interface supporting TCP/IP, that can interact with the Web Server. In this way, the client application and the Web Server are not required to be within transmission range, see Figure 7(b).

In the scenario depicted by Figure 7(a), the `BHSP` will post the received request to *localhost*, while in the scenario in Figure 7(b) it will post the request to the *domain* request by the client application. We remark that in the first case requests are bounded to the Web Server in the transmission range, while in the second one they can address any Web Server reachable from the `BHSP`.

IV. The BtHttpConnection Class

As discussed in Section II-C, the connection string given as parameter in the `Connector.open` polymorphically drives the type of the object that will be returned, according to the cast made by the developer

Our goal is to provide a new interface in the Connection hierarchy which provides HTTP support over Bluetooth. We named this interface `BtHttpConnection` and we implemented by extending the `HttpConnection` interface. As a result, we can invoke the `Connector.open` method with a HTTP-based connection string and decide to cast the generic `Connection` object returned either as a `HttpConnection` or a `BtHttpConnection` object. In this case, operations will be carried out over a Bluetooth channel. However, in order to do that we should modify the source code of the `Connector.open` method. Although the whole J2ME environment has recently gone open source, this modification should then be reflected in all the J2ME implementations by phones' vendors. Therefore, although the `BtHttpConnection` class extends the `HttpConnection` class, we cannot polymorphically cast the `HttpConnection` object to `BtHttpConnection`. Hence, we let the `Connector.open` method still return a `HttpConnection` object, but then we instantiate a `BtHttpConnection` method which takes in input all the information about the `HttpConnection` object:

Whenever a `BtHttpConnection` is instantiated, the following operations are performed:

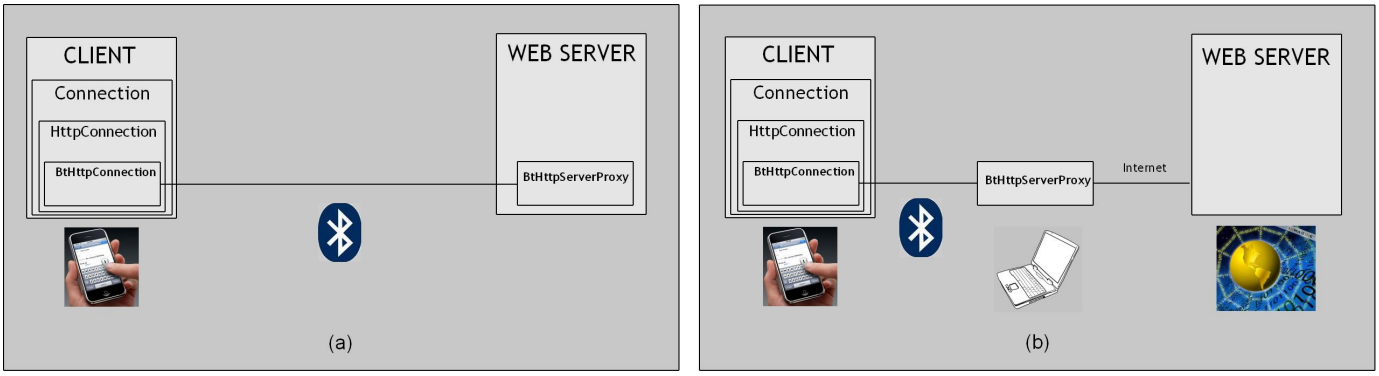


Fig. 7. Two possible settings for the BHSP.

```
(1) HttpConnection hc =(BtHttpConnection)
       Connector.open(defaultURL, Connector.READ_WRITE);

(2) BtHttpConnection bhc =new BtHttpConnection(hc);
```

Fig. 8. From a HttpConnection to BtHttpConnection

- Establish an association between the "*http*"-starting URL (given in the connection string) and a Bluetooth remote device (the `BHSP`).
- If the association has not been previously established, perform an inquiry operation to discover a Bluetooth device which exposes a "Web Server" service and which corresponds to the URL in the connection string. Afterwards, store the association in a local database.
- If the association has been previously established, recover the correspondent Bluetooth address from the local database.
- Once the Bluetooth Address has been found, establish a RFCOMM connection with the `BHSP`, which will be used to send/receive HTTP requests/responses.

Code reuse. We remark that the extra work required to implement HTTP connections over Bluetooth is totally transparent to application developers. In fact, `BtHttpConnection` provides programmers with the same interface as `HttpConnection`, masking all the implementation details of the Bluetooth interactions. Suppose that a programmer has implemented the MIDlet showed in Figure IV to performs a HTTP post to a servlet using WLAN from his mobile phone. If he wants to deploy its application on cheaper mobile phones, not supporting WLAN but supporting Bluetooth, then he has only to use the `BtHttpConnection` instead of `HttpConnection`. We stress that all the methods are unaltered, so no modification to the code is required. The operation is showed in Figure 9, which differs from only for the line code (2).

Figure 10 shows the resulting new class diagram with the new `BtHttpConnection` class to extend the `HttpConnection` class.

V. BHSP: The Bluetooth Http Server Proxy

The `BHSP` has been implemented as a Bluetooth listener daemon. It will run as a J2SE application on a desktop

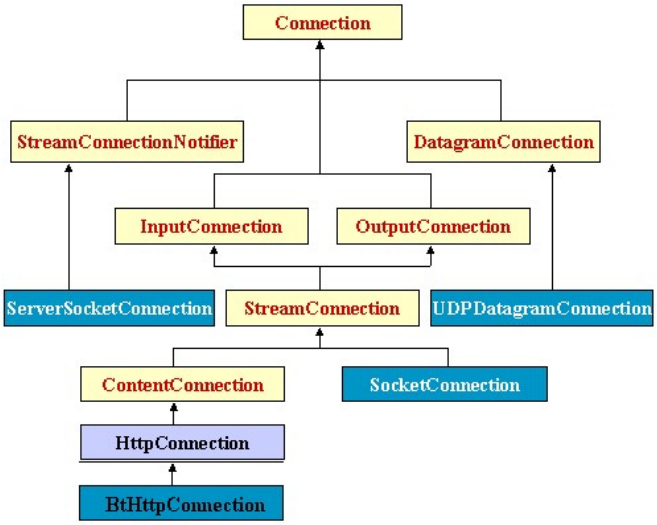


Fig. 10. The J2ME Connection hierarchy diagram

computer but it has been designed to operate at a low-level and to be lightweight (it has a small footprint) in order not to affect performance. It does not interpret processed data but it simply forwards it.

Whenever an incoming request is received from the Bluetooth channel, it is inserted in a queue and processed as soon as possible. The `BHSP` uses the Apache *Commons HttpClient* package [1] to execute the required method on the Web Server. Once that it has got the response, it forwards it back on the Bluetooth channel to the client application.

The work is performed by a `BTServer` class, which takes care of:

- setting the device in *discoverable* mode
- activating a listening connection
- accepting incoming connections
- performing I/O on the Bluetooth channel
- instantiate a `Poster` object.

The `Poster` class is in charge of:

- performing an HTTP post operation on the Web Server, using the Apache *Commons HttpClient* package
- giving back the request to the `BTServer` object

In order to have a licence-free and JSR-82 compliant implementation of our `BHSP`, we have considered two alternatives:

```
(1) HttpConnection hc =(BtHttpConnection) Connector.open(defaultURL, Connector.READ_WRITE);

(2) BtHttpConnection bhc = new BtHttpConnection(hc);

(3) bhc.setRequestMethod(HttpConnection.POST);

(4) DataOutputStream dos = bhc.openDataOutputStream();
(5) byte[] request body = requeststring.getBytes();
(6) for (int i = 0; i <; request body.length; i++)
(7)    dos.writeByte(request body[i]);

(8) dos.flush();
(9) dos.close();

(10) DataInputStream dis =  new DataInputStream(bhc.openInputStream());

(11) int ch;
(12) long len = bhc.getLength();
(13) if(len!=-1) {
(14)     for(int i = 0;i<len;i++)
(15)         if((ch = dis.read())!=-1)
(16)             messagebuffer.append((char)ch);
(17) } else { // if the content-length is not available
(18)     while ((ch = dis.read()) != -1)
(19)        messagebuffer.append(10) ;
}

(20) dis.close();
```

Fig. 9. A simple HTTP post operation performed over a Bluetooth channel

1) To use BlueCove [3], the free implementation of the JSR-82 API within the Microsoft Windows XP SP2.
2) To use BlueZ [7], the Linux Bluetooth stack, and provide a JSR-82 implementation for BlueZ.

We remark that for this part we have modified the implementations of the works in [21] and [22] that performed a similar operations for Web Services invocation over Bluetooth.

Figure 11 shows the main steps of the `BHSP`. The Java code in Figure 11 executes the following operations:

(1) Set the device in discoverable mode.
(2-4) Activate a listening connection on localhost, on the channel 1, named "rfcomm test".
(5) Accept incoming connections.
(6) Open an InputStream on the connection.
(7-8) Read data on the stream.
(9-10) Post the HTTP request at the specified address, using the `Poster` class, to get the response.
(11) Open an OutputStream on the connection.
(12) Write data, i.e. the HTTP response.

VI. A TRANSPARENT AND EFFICIENT SOLUTION

In Figure 12, we summarize how our solution works. The top section of the diagram depicts the association of the `BHSP`'s Bluetooth address to the *http://* URL, performed by the `BtHttpConnection` interface through an inquiry. The bottom section shows a generic operation over the established connection: data is exchanged between the client and the `BHSP` over the Bluetooth channel; requests to the Web Server are posted by the `BHSP` through the use of Apache *HttpClient* package [1].

We stress that the work needed to use a Bluetooth connection is totally transparent to both the application developers and the user. In fact, `BtHttpConnection` provides programmers with the same interface as `HttpConnection`, the only change required w.r.p. a normal HttpConnection-based MIDlet is to use the `BtHttpConnection` instead of `HttpConnection`.

We envision several applications for our solution. For instance, there could be waiting rooms, such as stations or airports, that provide a free Internet access to users, for timetable information, emails, weather forecasts. The same scenario could take place in trains, buses, coffee shops, restaurants. No massive money investment is required for this goal, other than exposing a BHSP (or more according to the expected number of users), to which users can connect with their simple J2ME- and Bluetooth-enabled mobile phone.

Also, we envision a scenario where Sun embeds our solution in the official J2ME specification, so that all the implementations will provide the support for HTTP connections over Bluetooth. The only requirement would be to have a BHSP proxy available in order to support the communication. We remark that our solution does not work on the TCP/IP protocol, but only allows simple POST/GET operations on Web Servers. As a result, there would be no support for congestion control, sessions, and all the other nice features provided by the protocol stack. However, we argue that Bluetooth communications, though as not reliable as TCP/IP ones, provides completely free communications on tiny and inexpensive devices. Moreover, we argue that the new Bluetooth technology (2.0) implemented by the new generation's chips is efficient enough to support simple HTTP operations, such as chats or Internet browsing on most of the web sites normally accessed by mobile phones' users..

Furthermore, we claim that our work fills an important gap of the J2ME environment. In spite of limitations such as communication speed or the necessary presence of a Bluetooth proxy, our solution finally gives the appropriate tool to application developers to deploy complex applications which work on Bluetooth. Indeed, we argue that this free technology could be exploited for many solutions, and not only for simple files exchange anymore. Thanks to its transparency, our solution is ready-to-use for real scenarios on many low-end price class of devices massively available and spread out today.

```
(1)  (LocalDevice.getLocalDevice).setDiscoverable(DiscoveryAgent.GIAC);

(2)  (StreamConnection) notifier = (StreamConnection)
(3)           Connector.open("btspp://localhost:1;name=rfcommtest;master=true;encrypt=false;authorize=false;
(4)           authentication=false;receiveMTU=512;transmitMTU=512");

(5)  notifier.acceptAndOpen();
(6)  InputStream input = notifier.openInputStream();

(7)  /* Perform buffered readings to get the request */
(8)  String request = input.read();
(9)  Poster poster = new Poster(address);
(10) String response = poster.doPost(request);
(11) OutputStream output = notifier.openOutputStream();
(12) output.write(response.getBytes());
```

Fig. 11. Java code for the BHSP.

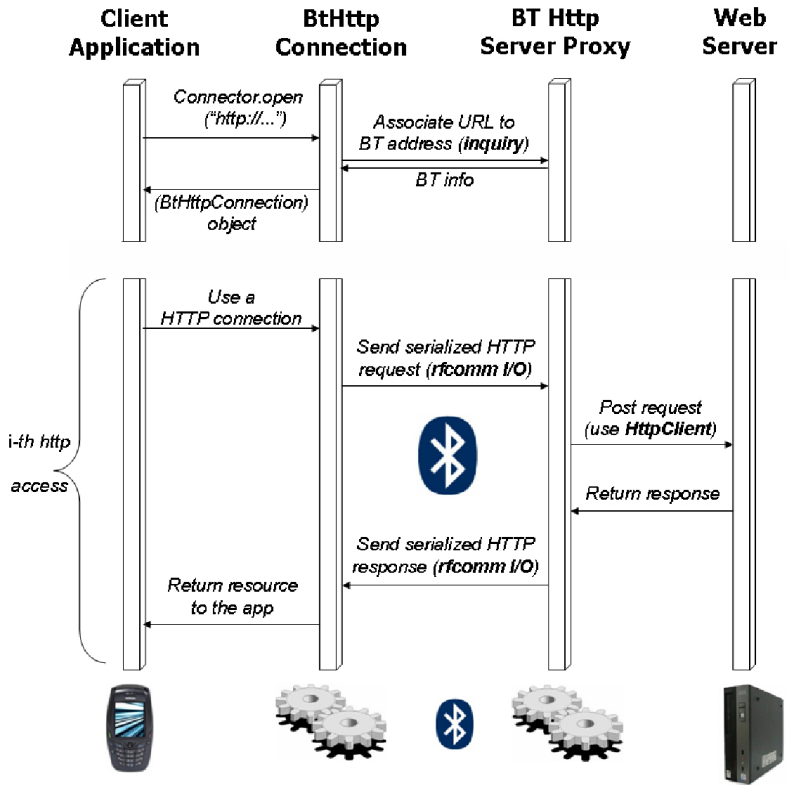


Fig. 12. Time Diagram of a client application accessing a Web resource.

VII. Performance Evaluation

In this section, we analyze the performance of our solution in order to evaluate its lightness and its usability in real world scenarios. To this aim we set up the following test bed:

- The WS and the BHSP lie on a PC IBM ThinkCentre 50 Personal Computer, with Pentium 4 2,6 GHz and 760 MB RAM, running Windows XP Professional SP 2, with a Bluetooth TrendNet TBW-102UB USB dongle, and BlueCove [3] implementation of the JSR-82 Bluetooth API for Java.
- The client application runs on a Nokia N73 mobile phone (Symbian OS 9.1), compliant with MIDP 2.0 and JSR-82 standards.

We evaluated the overhead taken by the `BtHttpConnection` class to let a mobile client interact with a Web Server using HTTP connections over a Bluetooth channel. To this aim, we have compared times to post growing size strings for the following applications:

1) A MIDlet which posts strings to a Bluetooth-enabled Web Service using the `BtHttpConnection`.
2) A simple MIDlet which sends strings to a remote device over Bluetooth.

Figure 13 shows times for strings ranging from 1 KB to 50 KB with an increasing size of 0,5 KB. Each test was repeated 50 times to get significant average times. As we can see in the diagram, the use of `BtHttpConnection` slightly slows down the computation. The other application can directly access the Bluetooth channel and send data over it, while the `BtHttpConnection` implies a small computation overhead to handle the connection and to give a higher profile to the application. However, the overhead is almost constant and small enough to consider our solution efficient enough to be used in real world scenario with complex applications.

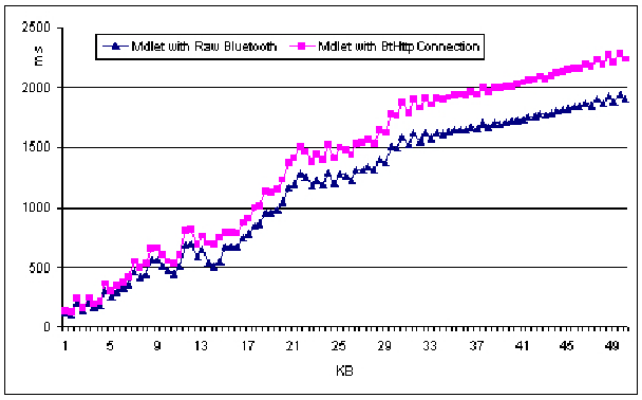


Fig. 13. Performance evaluation of `BtHttpConnection`.

VIII. Conclusion

We have presented a solution to establish HTTP connections over Bluetooth channels from low-end price class J2ME-enabled mobile phones. The resulting solution is lightweight and works at no extra cost for users and application developers, but it only requires the presence of a Bluetooth connection to a device connected to the Internet. The provided implementation requires no code modification and allows programmers to enhance the features of HTTP-based MIDlets with basically no effort, extending their range of action from mobile phones provided with GPRS/UMTS connections (sometimes expensive or not available) or WLAN access (available only on high-end price cost devices), but also to inexpensive mobile phones provided with Bluetooth interface, which is free to use.

Our performance evaluation confirms the real applicability and lightness of our solution, showing that it is efficient enough to be used in a real world scenario for a wide set of applications.

IX. Acknowledgements

This work has been partially supported by the European Commission through the IST program under contracts FP6-1596 (AEOLUS) and by the *Foundations of Adaptive Networked Societies of Tiny Artifacts* project, funded by the European Commission as project number 215270.

References

[1] Apache Commons HttpClient. `http://jakarta.apache.org/commons/httpclient/`.
[2] Avetana jsr-82 implementation. `http://www.avetana-gmbh.de/avetana-gmbh/produkte/jsr82.eng.xml`.
[3] BlueCove. `http://sourceforge.net/projects/bluecove/`.
[4] The Bluetooth SIG Standard. `http://www.bluetooth.com`.
[5] Bluetooth solutions by Atinav AveLink. `http://www.avelink.com/bluetooth/index.htm`.
[6] Bluetooth Wireless Technology. `http://www.ericsson.com/technology/techarticles/Bluetooth.shtml`.
[7] BlueZ: Official Linux Bluetooth protocol stack. `http://www.bluez.org/`.
[8] Broadcom Bluetooth Solutions. `http://www.broadcom.com/`.
[9] Critical Mass – The Worldwide State of the Mobile Web . `http://www.nielsenmobile.com/documents/CriticalMass.pdf`.
[10] Impronto Rococo Software. `http://www.rococosoft.com/`.
[11] J2ME: Java 2 Micro Edition. `http://java.sun.com/j2me/`.
[12] JSR 218, Connected Device Configuration(CDC).
[13] JSR 30, JSR 139: Connected Limited Device Configuration (CLDC). `http://java.sun.com/products/cldc/`.
[14] JSR 37, JSR 118: Mobile Information Device Profile (MIDP). `http://java.sun.com/products/midp/`.
[15] JSR 82: Java APIs for Bluetooth. `http://www.jcp.org/en/jsr/detail?id=82`.
[16] Mobile Information Device Profile 2.0 (MIDP 2.0): JSR 118. `http://jcp.org/aboutJava/communityprocess/final/jsr118/index.html`.
[17] Mobile Information Device Profile (MIDP): JSR 37. `http://jcp.org/aboutJava/communityprocess/final/jsr037/index.html`.
[18] Opera Mini Web Browser. `http://www.operamini.com/`.
[19] Opera Software Company. `http://www.opera.com/company/`.
[20] Windows support for Bluetooth. `http://msdn.microsoft.com/library/default.asp?url=/library/en-us/bluetooth/bluetooth/about_bluetooth.asp`.
[21] V. Auletta, C. Blundo, E. D. Cristofaro, and G. Raimato. A Lightweight Framework for Web Services Invocation over Bluetooth. pages 331–338, 2006.
[22] V. Auletta, C. Blundo, E. D. Cristofaro, and G. Raimato. Performance evaluation of web services invocation over Bluetooth. pages 1–8, 2006.
[23] V. Auletta, C. Blundo, and E. De Cristofaro. A J2ME transparent middleware to support HTTP connections over Bluetooth. In *Proceedings of the Second International Conference on Systems and Networks Communications (ICSNC 2007)*, 2007.
[24] B. Chatschik. An overview of the Bluetooth wireless tecnology. *IEEE Communication Magazine*, 39:86–94, 2001.
[25] Q. H. Mahmoud. The Java APIs for Bluetooth Wireless Technology - Part II. `http://developers.sun.com/mobility/midp/articles/bluetooth2`, 2003.
[26] G. Sarswat and J. Noida. Bluetooth Hacking. `http://cnss.wordpress.com/2007/09/11/bluetooth-hacking1`, 2007.
[27] W. Stallings. Wireless communications and networks, 2005.

Preliminary 2009 Conference Schedule

http://www.iaria.org/conferences.html

NetWare 2009: June 14-19, 2009 - Athens, Greece

- SENSORCOMM 2009, The Third International Conference on Sensor Technologies and Applications
- SECURWARE 2009, The Third International Conference on Emerging Security Information, Systems and Technologies
- MESH 2009, The Second International Conference on Advances in Mesh Networks
- AFIN 2009, The First International Conference on Advances in Future Internet
- DEPEND 2009, The Second International Conference on Dependability

NexComm 2009: July 19-24, 2009 - Colmar, France

- CTRQ 2009, The Second International Conference on Communication Theory, Reliability, and Quality of Service
- ICDT 2009, The Fourth International Conference on Digital Telecommunications
- SPACOMM 2009, The First International Conference on Advances in Satellite and Space Communications
- MMEDIA 2009, The First International Conferences on Advances in Multimedia

InfoWare 2009: August 25-31, 2009 – Cannes, French Riviera, France

- ICCGI 2009, The Fourth International Multi-Conference on Computing in the Global Information Technology
- ICWMC 2009, The Fifth International Conference on Wireless and Mobile Communications
- INTERNET 2009, The First International Conference on Evolving Internet

SoftNet 2009: September 20-25, 2009 - Porto, Portugal

- ICSEA 2009, The Fourth International Conference on Software Engineering Advances
  - SEDES 2009: Simpósio para Estudantes de Doutoramento em Engenharia de Software
- ICSNC 2009, The Fourth International Conference on Systems and Networks Communications
- CENTRIC 2009, The Second International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services
- VALID 2009, The First International Conference on Advances in System Testing and Validation Lifecycle
- SIMUL 2009, The First International Conference on Advances in System Simulation

NexTech 2009: October 11-16, 2009 - Sliema, Malta

- UBICOMM 2009, The Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies
- ADVCOMP 2009, The Third International Conference on Advanced Engineering Computing and Applications in Sciences
- CENICS 2009, The Second International Conference on Advances in Circuits, Electronics and Micro-electronics
- AP2PS 2009, The First International Conference on Advances in P2P Systems
- EMERGING 2009, The First International Conference on Emerging Network Intelligence
- SEMAPRO 2009, The Third International Conference on Advances in Semantic Processing