JYVÄSKYLÄ STUDIES IN COMPUTING
135

Jukka Mäkelä

Mobility Management in Heterogeneous IP-networks

JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ STUDIES IN COMPUTING 135

Jukka Mäkelä

Mobility Management in Heterogeneous IP-networks

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksen Lea Pulkkisen salissa
lokakuun 28. päivänä 2011 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in the building Agora, Lea Pulkkinen hall, on October 28, 2011 at 12 o'clock noon.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2011

Mobility Management in Heterogeneous IP-networks

JYVÄSKYLÄ STUDIES IN COMPUTING 135

Jukka Mäkelä

Mobility Management in Heterogeneous IP-networks

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2011

Editors
Timo Männikkö
Department of Mathematical Information Technology, University of Jyväskylä
Pekka Olsbo, Ville Korkiakangas
Publishing Unit, University Library of Jyväskylä

URN:ISBN:978-951-39-4454-4
ISBN 978-951-39-4454-4 (PDF)

ISBN 978-951-39-4453-7 (nid.)
ISSN 1456-5390

Jyväskylä University Printing House, Jyväskylä 2011

ABSTRACT

Mäkelä, Jukka
Mobility Management in Heterogeneous IP-networks
Jyväskylä: University of Jyväskylä, 2011, 86 p.(+included articles)
(Jyväskylä Studies in Computing
ISSN 1456-5390; 135)
ISBN 978-951-39-4453-7 (nid.)
ISBN 978-951-39-4454-4 (PDF)
Finnish summary
Diss.

The availability of different wireless access networks offers new possibilities to create and use new services. Mobile users are able to use these services whenever and wherever they choose. Also the current trend of using online social services increases the number of users and mobile devices. The heterogeneity of the access networks in current architectures sets also several technical challenges. One of the key challenges is the support for mobility between different access networks: the mobility should be seamless to the users so that there is no interruption in the services used.

This dissertation discusses research challenges and problems caused by the heterogeneity of access networks. The work includes a study and development of mechanisms which will enable and help to achieve seamless mobility in heterogeneous environment. Included is also an analysis of the current mobility management functions and protocols. The functionalities of several standard state-of-the art IP mobility protocols are introduced and the challenges of the heterogeneous environment are addressed. A special focus is on solving problems caused by mobility protocol handovers, handover decisions and on collecting and distributing mobility related information. This dissertation also shows how mobility management can work as a distributed solution in the current and future network architectures.

The work resulted in several new functionalities for mobility management: a new mobility protocol, a new system for controlling vertical handovers and a framework, supporting collection, processing and distribution of mobility related events. The discussion of the problems in the heterogeneous networks and the new functionalities developed are summarized, with the detailed results and evaluation studies.

Keywords: Mobility management, handover, heterogeneous networks, mobility triggers, cross-layer information

Author
M.Sc. Jukka Mäkelä
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Supervisor
Prof. Timo Hämäläinen
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Reviewers
Prof. Yevgeni Koucheryavy
Department of Communication Engineering
Tampere University of Technology
Finland

Dr. Enzo Mingozzi
Department of Information Engineering
University of Pisa
Italy

Opponent
Prof. Jukka Manner
Department of Communications and Networking
Aalto University
Finland

ACKNOWLEDGEMENTS

This dissertation has been a significant challenge for me, and its completion would not have been possible without the help of many people. I would like to use this opportunity to thank everyone who helped me in this enterprise.

I would especially like to express my gratitude to my supervisor, Professor Timo Hämäläinen, for all the help and guidance I received throughout my graduate and post-graduate studies and through the process of dissertation work. I also wish to thank Professor Yevgeni Koucheryavy and Professor Enzo Mingozzi for reviewing this dissertation and Professor Jukka Manner for acting as an opponent.

It is a great pleasure to thank all my co-authors and colleagues for their cooperation and contributions to the combined publications of this dissertation. Those people I refer to are: Gabor Fekete, Kostas Pentikousis, Markus Luoto, Tiia Sutinen, Vesa Kyllönen, Jari Tenhunen, Ramón Aqüero, Johnny Choque, Petteri Weckström, Anna-Maija Virkki and Jorma Narikka. I thank Steve Legrand for the language review of this dissertation and Pertti Raatikainen and Jyrki Huusko for valuable comments and discussions during the dissertation work. I'm quite certain that I have forgotten to mention some colleagues who deserve my thanks. My apologies to them. You all know you have been there.

Also the staff and all colleagues at VTT, especially at the Network technologies center, at the Jyväskylä University of Applied science and at Jyväskylä University are acknowledged for providing me with an excellent working environment. I would also like to acknowledge the various projects funded by the Tekes - the Finnish Funding Agency for Technology and Innovation, European Union and several industrial partners. My special thanks to the DRiWE, VerHo, Ambient Networks, MERCoNe and WISEciti projects. I wish to thank all the partners in these projects for interesting and fruitful discussion and joint activities.

I would like to acknowledge the Research and Education Foundation of TeliaSonera Finland, Telealan edistämissäätiö, Ulla Tuominen Foundation, Ellen ja Artturi Nyyssönen Foundation, Nokia Foundation, Finnish Foundation for Technology Promotion and the Finnish Cultural Foundation for their encouragement grants for this dissertation work.

Thanks to my parents (R.I.P.) as well as the other relatives and friends for being there for me. Finally, I can find no words to express the depth of my gratitude in thanking my loved one, Satu, for her love, understanding and support that I have needed to get this work done.

Oulu, October 4, 2011,
Jukka Mäkelä

ACRONYMS

3G	3rd Generation
3GPP	The 3rd Generation Partnership Project
3GPP2	The Third Generation Partnership Project 2
ANDSF	Access Network Discovery and Selection Function
ANI	Ambient Network Interface
ANISI	Ambient Networks Information Service Infrastructure
AR	Access Router
ARI	Ambient Resource Interface
ASI	Ambient Service Interface
CN	Correspondent Node
CoA	Care of Address
CPU	Central Processing Unit
DCMF	Distributed Control and Management Framework
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DRiWE	Dynamic Routing in Wireless Environment
e-UTRAN	evolved UMTS Terrestrial Radio Access Network
EDGE	Enhanced Data rates for GSM Evolution
FA	Foreign Agent
FMIPv6	Fast Mobile IPv6
GPRS	General Packet Radio Service
HA	Home Agent
HDD	Hard Disk Drive
HI	Host Identity
HIP	Host Identity Protocol
HO	Handover
HoA	Home Address
HSDPA	High Speed Downstream Packet Access
HSPA	High Speed Packet Access
HSUPA	High Speed UPstream Packet Access
I-WLAN	Interworking Wireless Local Area Network
IP	Internet Protocol

IPCP	Internet Protocol Control Protocol
IPSec	Internet Protocol Security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IrDA	Infrared Data Association
LIP	Link Information Provider
LMA	Local Mobility Anchor
LTE	Long Term Evolution
MAG	Mobile Access Gateway
MAP	Mobility Anchor Point
MES	Mobile Expert System
MIH	Media Independent Handover
MIP	Mobile IP
MIPv4	Mobile IPv4
MIPv6	Mobile IPv6
MN	Mobile Node
MR	Mobile Router
MTU	Maximum Transmission Unit
NAR	New Access Router
NEMO	Network Mobility
NES	Network Expert System
NIS	Network Information Server
OSI	Open System Interconnection
PAR	Previous Access Router
PC	Personal Computer
PDA	Personal Digital Assistant
PDP	Packet Data Protocol
PPP	Point-to-Point Protocol
QoS	Quality of Service
RFC	Request For Comments
SNMP	Simple Network Management Protocol
SOM	Self Organizing Maps
TCP	Transmission Control Protocol
TMM	Trigger Management Mechanism

TRG	Trigger Management Framework
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
VERHO	Vertical Handover in Heterogeneous Environment
VoIP	Voice over Internet Protocol
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless local Area Network
WPAN	Wireless Personal Area Network
WWAN	Wireles Wide Area Network
XACML	eXtensible Access Control Markup Language

LIST OF FIGURES

FIGURE 1 Examples of different network types with the level of mobility and data rates 20
FIGURE 2 Movement of Mobile Node in a heterogeneous environment 26
FIGURE 3 Vertical handover........ 27
FIGURE 4 Ambient Control Space........ 28
FIGURE 5 Composed Ambient Control Spaces........ 28
FIGURE 6 Mobility Control Space Architecture 29
FIGURE 7 Examples of New Mobility Dimensions 30
FIGURE 8 Mobile IPv4 topology 32
FIGURE 9 Mobile IPv6 topology 32
FIGURE 10 Topology of fast handovers for Mobile IPv6 33
FIGURE 11 Topology of Hierarchical Mobile IPv6 34
FIGURE 12 Proxy Mobile IPv6 topology........ 34
FIGURE 13 Example topology for HIP protocol 35
FIGURE 14 Example topology of Network Mobility 36
FIGURE 15 DRiWE protocol handover mechanism 37
FIGURE 16 DRiWE protocol handover signaling........ 38
FIGURE 17 DRiWE testbed 40
FIGURE 18 Cross-layer architecture 45
FIGURE 19 An example of trigger groups 50
FIGURE 20 TRG-Architecture 52
FIGURE 21 TRG-Policy Manager Message Sequence Diagram........ 55
FIGURE 22 The video is streamed to the laptop (left), followed by a session handover to the PDA (center), and a handover back to the laptop (right)........ 58
FIGURE 23 Topology of the session handover testcases 59
FIGURE 24 Experimental results when triggering a session HO 60
FIGURE 25 Zoom in to experimental results when triggering a session HO . 60
FIGURE 26 Triggers in Test Case 1 61
FIGURE 27 Triggers in Test Case 2 61
FIGURE 28 Processing time in Test Case 1 without filter processing........ 63
FIGURE 29 Processing time in Test Case 1 with filter processing........ 63
FIGURE 30 Average processing time per trigger in Test Case 2 without filter processing 64
FIGURE 31 Average processing time per trigger in Test Case 2 with filter processing........ 65
FIGURE 32 The Distributed Control and Management Framework (DCMF) 66
FIGURE 33 Main components of Distributed Control and Management Framework 67
FIGURE 34 Message flow between the components........ 68
FIGURE 35 Heterogeneous networks testbed and the DCMF evaluation scenario 69

FIGURE 36 One-way packet delay and handovers in our testbed evaluation 71
FIGURE 37 Signal strength measurement ... 71
FIGURE 38 Network congestion in WLAN1 in case of DCMF assistance ... 72
FIGURE 39 Network congestion in WLAN1 without the assistance from DCMF ... 73
FIGURE 40 User experience in handover case ... 74

LIST OF TABLES

TABLE 1 Trigger format 54
TABLE 2 Total number of delivered triggers and average processing time (in ms) per trigger in test case 1 62
TABLE 3 Example triggers in the testcase.. 70
TABLE 4 Packet delay distribution (sec)... 73

CONTENTS

ABSTRACT
ACKNOWLEDGEMENTS
ACRONYMS
LIST OF FIGURES
LIST OF TABLES
CONTENTS
LIST OF INCLUDED ARTICLES
CONTRIBUTIONS IN THE INCLUDED ARTICLES

1 INTRODUCTION 19
  1.1 Mobility 21
  1.2 Mobility triggers 21
  1.3 Problem statement and motivation 22
  1.4 Outline of the dissertation 24

2 MOBILITY MANAGEMENT 25
  2.1 Mobility management in heterogeneous IP Networks 25
  2.2 Handover types 26
  2.3 Ambient Networks mobility management 28
    2.3.1 Mobility Dimensions 30
  2.4 Mobility management protocols 31
  2.5 DRiWE mobility protocol 36
    2.5.1 DRiWE protocol functionality 36
    2.5.2 DRiWE protocol handover mechanism 37
      2.5.2.1 DRiWE location update mechanism 39
    2.5.3 Implementation and evaluation 39
  2.6 Comparison between MIPv6, FMIPv6 and DRiWE protocols 40
  2.7 Handover decisions 43
    2.7.1 VERHO system for mobile nodes 44
    2.7.2 Decision process in VERHO system 44
    2.7.3 Information sources used in VERHO system 46
  2.8 Summary 48

3 MOBILITY EVENT COLLECTION AND DISTRIBUTION 49
  3.1 Mobility triggers 50
  3.2 Trigger management framework 51
    3.2.1 Triggering Events Collection 53
    3.2.2 Trigger Processing 53
    3.2.3 Trigger Repository 54
    3.2.4 Policies and rules in trigger management 54
  3.3 Trigger management framework and 802.21 Media independent handover 55
  3.4 Validation and experimental evaluation 57

3.4.1 Session mobility with trigger management framework...... 57
3.4.2 Performance analysis of trigger management framework .. 60
3.5 Distributed mobility management in heterogeneous networks...... 66
3.5.1 Distributed Control and Management Framework (DCMF) 67
3.5.2 Validation in heterogenous networks testbed 69
3.5.3 Results.. 70
3.6 Summary.. 74

4 CONCLUSIONS AND FUTURE WORK.. 76

YHTEENVETO (FINNISH SUMMARY) 78

REFERENCES.. 79

INCLUDED ARTICLES

LIST OF INCLUDED ARTICLES

PI J. Mäkelä, G. Fekete, J. Narikka, T. Hämäläinen and A-M Virkki. Soft handover and routing mechanisms for mobile devices. *Proceedings of The 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2004)*, 2004.

PII J. Mäkelä, T. Hämäläinen, G. Fekete, and J. Narikka. Intelligent Vertical Handover System for Mobile Clients. *Proceedings of the 3rd International Conference on Emerging Telecommunications Technologies and Apllications (ICETA2004)*, 2004.

PIII J. Puttonen, G. Fekete, J. Mäkelä, T. Hämäläinen and J. Narikka. Using Link Layer Information for Improving Vertical Handovers. *Proceedings of The 16th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2005)*, 2005.

PIV J. Mäkelä, R. Agüero, J. Tenhunen, V. Kyllönen, J. Choque. Paving the Way for Future Mobility Mechanisms: A Testbed for Mobility Triggering & Moving Network Support . *Proceedings of 2nd International IEEE/Create-Net Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (Tridentcom 2006)*, 2006.

PV J. Mäkelä and K. Pentikousis. Trigger Management Mechanisms. *Proceedings of the Second International Symposium on Wireless Pervasive Computing (ISWPC 2007)*, 2007.

PVI J. Mäkelä. Towards Seamless Mobility with Cross-layer Triggering. *Proceeding of 18th Annual IEEE International symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007)*, 2007.

PVII J. Mäkelä, K. Pentikousis and V. Kyllönen. Mobility Trigger Management: Implementation and evaluation. *International Journal of Communications, Network and System Sciences, Vol.2, No.3, (IJCNS 2009)*, 2009.

PVIII J. Mäkelä, M. Luoto, T. Sutinen and K. Pentikousis. Distributed Information Service in overlapping Multiaccess Networks. *International Journal of Multimedia Tools and Applications, special issue on mobile media delivery, volume 55, issue 2, 2011, Springer* , 2011.

CONTRIBUTIONS IN THE INCLUDED ARTICLES

This section gives a short description of the included articles and the author’s contribution on the content of the papers.

Article [PI] introduces mechanisms which offer soft handover support between two different IP subnets for mobile devices. The mobility support protocol that was developed, DRiWE, tries to solve the same problem as Mobile IP, that is, how to manage up-to-date information about the location of a moving node and to keep its connections established during its movement. The author contributed to the idea and definition of the implementation for the protocol developed. The author also participated in the testbed realization and testing of the implementation in the testbed. Also the evaluation of the test results was part of the author’s contribution. This article and the results are summarized in Section 2.5.

Article [PII] introduces a vertical handover system (VERHO). The VERHO system is able to make decisions about which available access technology should be used in a mobile device. The system uses the information deduced from the decision process and controls the Mobile IPv6 handover. The decision when handover should occur depends on the information collected from each available technology and data flows in situations where several different access technologies are available. The author was one of the key contributors for the idea behind the system. The author’s work include the design and specification of the VERHO system and its implementation. The design of VERHO included also the design to support the use of cross-layer information and analysis of the feasibility of the seamless handovers. This article and the results are summarized in 2.7.1.

Article [PIII] introduces a common triggering mechanism (cross-layer framework) and logic to gather link state and quality related information from the link layers of different access technologies. This information may be utilized by upper layer protocols and applications to react to changes of the link layer. As one of the main architects of the VERHO system, the author participated in the work by providing the requirements and guidelines on how the systems, such as link layer information provider, presented also in this article, can be integrated with the VERHO system. This article and the results are summarized in Section 2.7.3.

Article [PIV] presents a feasibility study of some architectural components of a mobility control space, which was designed in the context of the Ambient Networks project. The work in this article followed a fully experimental approach and focused on and successfully realized two concepts: a facility for triggering mobility events and support for moving networks. The author participated in the definition of the testbed scenario and in the realization of the testbed. One major part of the author’s work, described also in this article, was the design, implementation and tests of the session handover mechanism. The author participated also in the evaluation of the test results. This article and the results are summarized in Section 3.4.

Article [PV] presents the main concepts that govern the trigger management framework. These concepts aim at operating in a richer mobility management framework, and enable the deployment of new applications and services. After summarizing the architectural requirements with respect to trigger collection, processing, storage, and dissemination, the article introduces a real implementation of the Trigger Management Mechanism/Framework (TMM or TRG in this dissertation) on commodity mobile devices. The article briefly reviews the testbed environment and presents the experimental results of a test case of a session handover with a lossless streaming video session handover between a laptop and a PDA. The test case described in this article is the continuation of the work described in the previous article [PIV]. The author's contribution to this work is the design of the trigger management framework and analysis of the available mobility triggers. The author also participated in the tests and evaluations of the results presented in this article. This article and the results are summarized in Section 3.2.

Article [PVI] puts together the work described in articles [PIII] VERHO and [PV] TRG. It shows how the trigger management framework (TRG) can be bound to the vertical handover controller system (VERHO), enabling handover decisions based on several input parameters. This article was produced solely by the author and concentrates on the conceptual level of the design of the system where TRG and VERHO may be integrated. This article and the results are summarized in Section 2.7.

Article [PVII] concentrates on the evaluation of the trigger management framework (TRG) implementation. The article gives a brief introduction to the trigger management framework, the definition of which has been one of the key parts of the author's dissertation work, and addresses several implementation issues, such as event collection and processing, storage, and trigger dissemination. A real implementation for commodity mobile devices is also introduced. The article consist of two different test cases defined for validating and testing the TRG implementation performance. The author participated also in the definition of the test cases, the results of the evaluations of which are presented in this paper. The article and the results are summarized in Section 3.4.2.

Article [PVIII] presents a distributed information service, which can enhance media delivery over multiaccess networks. The article describes the information service, which is built upon the new distributed control and management framework (DCMF) and the mobility management triggering functionality (TRG). The article introduces a testbed, which includes WLAN, WiMAX and 3G/HSPA network accesses, evaluates the proposed architecture and presents results that demonstrate its value in enhancing video delivery and minimizing service disruption in a scenario involved. The author was one of the main architects and participated in the design and specification of the distributed control and management framework (DCMF). The author also participated in the planning and performing the tests in the testbed realized and in the evaluation and analysis of the achieved results. The article and the results are summarized in Section 3.5.

1 INTRODUCTION

New wireless techniques offer new possibilities for creating new services. The availability of wireless access networks with more and more bandwidth enables the use, for example, of some streaming software or allows watching live video streams or TV broadcasts with mobile devices. The current trend of using social medias such as Facebook has increased the use of the network services. Users also want to use the services whenever and wherever they choose, with optimized bandwidth and costs. The increasing use of services is not all about entertainment; it also enables services like mobile payments and more personal healthcare as well [65]. The large number of available services has attracted more users, and the number of mobile devices is rising all the time. It is also thought that the use of connected time in networks will increase in the future. The key technology enabler in current networks is the Internet Protocol (IP) with versions IPv4 [62] and IPv6 [23]. These protocols are the key elements to enable different networks and nodes to work together. Effective use of new services also sets several requirements to networks and protocols. Mobile devices have several available access technologies, including WLAN, 3G, LTE, WiMAX or GPRS, and the users should be able to connect to each network with their devices. One of the key issues is to support the mobility of the devices in the infrastructure. Supporting mobility and enabling access in the different networks available will give people more freedom regarding the time and place for using the services. Users are no longer static: they are expecting to be connected anytime and anywhere.

The problem of heterogeneity of networks is caused by the differences in technology. Each technology needs much hardware dependent technology. Also dedicated signaling protocols and implementations are needed. One way to categorize networks is based on their coverage area. Networks with a typically small coverage area (10 meters) belong to Wireless Personal Area Networks (WPAN). Common technologies for WPAN are Bluetooth and Infrared Data Association (IrDA) technology. Wireless Local Area Networks (WLAN) usually cover an area between ten and a few hundred meters. Technologies such as IEEE 802.11 a/b/g/n are common WLAN technologies and used, e.g., in home environment. The largest coverage area is provided by Wireless Wide Area Networks (WWAN).

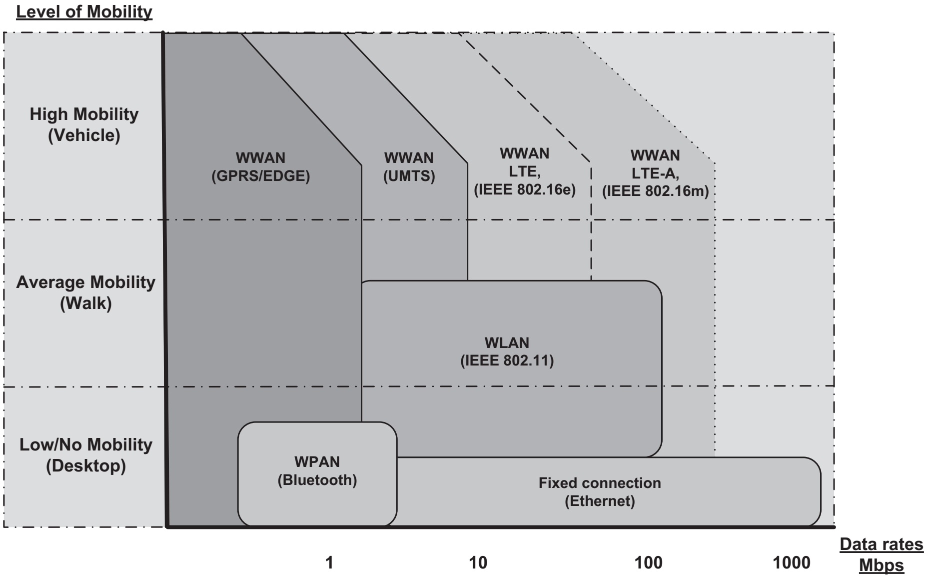


FIGURE 1 Examples of different network types with the level of mobility and data rates

These networks are typically maintained by operators or service providers. Technologies like GPRS, 3G/UMTS/ or upcoming LTE are examples of the WWAN networks. Even if all needed accesses could be covered with WWAN networks, there are clear benefits for using also smaller coverage area networks. Building a WLAN network at home with IEEE 802.11 technology devices is considerably cheaper than using WWAN technologies and easier to manage. WPAN networks are suitable for very short distance point-to-point connections. It is also expected that the wider the networks coverage area is the more mobility with less bandwidth is available within the network. The optimal case is when the users can move across the technology boundaries seamlessly and get the connection, as said before - anytime and anywhere. Figure 1 illustrates the level of mobility and bandwidth in the different types of current and future networks. For example, in WLAN networks, the coverage area and the mobility inside the network is small while the available bandwidth is quite high.

This dissertation discusses the functionality and the problems of the different mobility management protocols and their behavior, especially in the heterogeneous environment. In addition, the information that will trigger the mobility management functions in different states of the devices and networks is discussed.

1.1 Mobility

The problem of mobility management in IP networks is caused mainly by the mobility of end devices. Originally networks were designed for devices like desktop PCs that are static and are rarely moved to different locations. But as the technology evolved and more and more, efficient smaller devices were introduced. People started carrying their devices with them, and the result was a mess in the network topology. The key issue here is the change of the point of attachment, since it often means also a change in the IP-address. So whenever the user is using a connection, the device needs to get an IP address, which enables the communication with other devices, from the attached network. The current location needs to be updated as well so that end-to-end connections with other devices will be possible and the IP-packet can be routed to the right destination. What makes it more difficult is that users started to use IP connections not only in different locations but also while moving, and a new mechanism called handover was needed

The idea for keeping the location updated with the help of mobility protocols is quite similar to the approaches used in our everyday life, e.g., in post offices and mail delivery. Let's imagine that Mr. Isotalo's street address is Mattilanniemi 2. Everyone who wants to send a mail to him sends it with this that address. The mail is usually first delivered to the post office where it is rerouted to his delivery address. After a couple of years Isotalo decides to move to Oulu and uses now Kaitoväylä 1 as his address. Now if Isotalo does not inform anyone about his new address, all his mail will still be delivered to the old address, which is now used by Mr. Rannanjärvi, and the mail may get lost.

So here the post office is similar to the home agent that is used in different mobility protocols. The home agent should always know the current location of the mobile node. Sometimes there is also a local mail service, like internal mail services in big organizations where mail is not delivered via post office, and sometimes people might even drop mail directly to Isotalo's mailbox if they are close enough. In a similar way, as discussed later in this dissertation, mobility management protocols use route optimization and localized mobility management mechanisms to optimize data delivery.

1.2 Mobility triggers

To keep the connections on while the users and devices move, support for mobility management is needed. Mobility management functions can work automatically, without the users' assistance. However the functions need to be triggered to start any needed actions. A trigger is typically information about a change in environment. Probably the most common measured variable is the signal strength. When the user is moving out of the coverage of a wireless access network, the signal will weaken. This initiates the handover mechanism to change the point of attachment to another network with a stronger signal strength. Signal strength is just one example of a trigger, and it is good to remember that a trigger may originate from the different layers of the protocol stack. Triggers can be based, for example, on user preferences such as the cost of the network use or the supported quality of service (QoS). Cost free WLAN networks may be the highest priority for some user when faced with choosing a new network access. Mobility triggers, as well as collecting, distributing and using the triggers, is one important topic in this dissertation and discussed more in the following Chapters 2 and 3.

1.3 Problem statement and motivation

The main ideas of the research of this dissertation were based on the assumptions that mobile devices will have several different access technologies available and that the network environment will be heterogeneous with several access technologies. Now in 2011 these assumptions made in 2004 seem to be correct and valid. For example Nokia N8-00 smartphone supports GPRS/EDGE, 3G (HSDPA, HSUPA) and WLAN 802.11 a/b/g/n and Apple Iphone 4 supports GPRS/EDGE, 3G (HSDPA,HSUPA), WLAN 802.11 b/g/n. So the devices support many different technologies that may be used for data access. And the upcoming technologies like LTE, LTE-A, WiMAX will add even more heterogeneity to the network environment used and will provide more challenges also for mobility management functions.

The 3rd Generation Partnership Project (3GPP) [73] has defined mechanisms to support vertical handovers between 3GPP (e.g. LTE based E-UTRAN) and non-3GPP (e.g. WLAN, WiMAX) accesses. This is a good sign, since traditionally different standardization forums have been taking care only of compatibility issues between the technologies in their own agenda. Standardization of the different access network technologies in different forums is one of the main reasons for the current heterogeneity of access networks. 3GPP has already (from release 6) introduced mechanisms to provide a network service also via non-3GPP accesses. One of these, called interworking WLAN (I-WLAN) [1], provides the network services via an encrypted tunnel into the core network, however it does not allow the handover between I-WLAN and 3GPP [66].

From 3GPP release 8, the mobility management supports change between different access technologies. 3GPP release 8 introduces also a new entity access network discovery selection function (ANDSF), which is used to discover non-3GPP networks, such as WiMAX or WLAN. ANDSF is used also for discovery and selection of the 3GPP access, but only when the mobile node is connected to via a non-3GPP access. ANDSF aims to provide information for selecting the most suitable target network and to further minimize the delay caused by the handover. IEEE 802.11 Media Independent Handover [26]] is a similar approach to ANDSF, both aiming to improve the handover behavior, to get a step closer towards seamless handovers [15]. IEEE 802.21 is discussed in Section 3.3. Nevertheless, support for IP mobility management protocols, such as Mobile IP is still needed. This means that the same drawbacks and challenges of IP mobility still remain, which gives also more motivation to the work discussed in this dissertation. The current state-of-art mobility management and IP mobility protocol functionalities are discussed in Chapter 2.

The target of this dissertation work was to study and develop mechanisms which will enable and help to achieve seamless mobility in a heterogeneous environment. The work addresses several different research areas. First mobility management protocols' functionality was studied resulting in a definition of DRiWE, a new mobility protocol. After that the research concentrated more on the management of vertical handovers and handover decisions in a heterogeneous network environment (VERHO system); this research is discussed in Chapter 2. One important part of the research concentrated on the problem of collecting and distributing mobility related events that may trigger the mobility management function as discussed in Chapter 3. The main research problems addressed in this dissertation are summarized in the list below:

- Optimizing vertical mobility management handovers and protocols. Mobile devices have several different interfaces and support for multihoming. This means that devices will need support for maintaining connectivity while changing their access technology. In other words, a mechanism for handovers is mandatory. Handovers should happen also as seamlessly as possible. The current standards' mobility protocols are not able to maintain connectivity seamlessly without any extensions. The handover decisions based on different inputs plays an important role in the mobility management and should be efficient enough to enable a good quality of service during handovers. These research problems are discussed in Chapter 2

- Collecting and distributing mobility related events and triggering handovers. Mobility events, and collecting and distributing them, plays a very important role in the mobility management. Guiding handover decisions with current and important information available in different networks, as the mobile device enables the handover processes to react faster to the changing environment. The most important information is related to the detection of movement and to possible upcoming lost of connection. Movement detection is usually based on events caused by weakened signal strength or can rely purely on network layer events such as advertisement messages of the routers. Mobile IP protocol [56] uses the latter approach and is aware only of IP connection changes. In addition to these, there is a significant amount of other information available such as power consumption of a certain technology or interface, the cost of using a certain access technology or the supported level of quality of service. To support the collection and distribution of these various mobility related events in an efficient way will help the mobility management functions work faster and more intelligently. The collection and distribution of these events is discussed in Chapter 3

1.4 Outline of the dissertation

The rest of the dissertation is organized as follows. Chapter 2 presents mobility management in heterogeneous environment. It introduces several protocols suitable for the IP-layer mobility management, including also the description of the DRiWE protocol as an example of the research results of this dissertation work. The chapter continues with the description of a concept, VERHO, for vertical handovers supporting intelligent decision algorithms. Chapter 2 also describes how VERHO can work together with the Triggering Management Framework as a part of mobility management. Chapter 3 concentrates more on mobility related events collection and distribution. It describes the functionality of the designed Triggering Management Framework (TRG) and outlines several implemented testbeds, used for validating the work. This chapter describes also how mobility management works in a distributed way and introduces the concept of distributed management and control framework (DCMF). Finally Chapter 4 concludes the mobility management work discussed in this dissertation and gives directions for future research.

2 MOBILITY MANAGEMENT

This chapter describes the main functionalities and protocols for mobility management in IP networks and summarizes the background and the main results of the publications PI-PIII and PVI.

2.1 Mobility management in heterogeneous IP Networks

In a heterogeneous network environment several different access technologies are available, including WLAN, 3G, GPRS, LTE. To be able to use these different access technologies, devices need a vertical handover [40] support, a mechanism for handovers between the technologies.

Figure 2 illustrates a situation in a heterogeneous environment when a Mobile Node is moving through an area where GPRS, WiMAX and IEEE 802.11 hotspots are available. There are already many protocols, such as those described in Section 2.4 designed to support vertical handovers. The optimal solution would be that the data flow could run all the time during the handover with no interruptions in the services. This mechanism is called seamless handover. It would be easier to achieve seamless handover if a connection to the new access point could be built before the old connection breaks. Unfortunately, many layer 2 technologies require that the connection to the old access point must be cut before the new one can be established. However, the first step before the handover and updating the location of the Mobile Node is the decision about the handover. It may also lead to a decision about using a different access technology. The decision is usually based on the availability of the network, which is, of course, a basic requirement for keeping the connection established. The decision may depend also on the services and applications the user wants to use (required bandwidth, etc). There is also a significant amount of information, like traffic load, available in the network. Facts such as the actual cost of using a technique, may be quite important for users as well. This information can originate from the device itself or from the network.

Information can originate from different OSI Reference Model [83] layers. This multilayer information is known as cross-layer information. It is shown by many studies that the use of cross-layer information provides significant benefits for making decisions about adaptations for multimedia streams [25, 41]. There are different types of events related to the mobility management - see, for example, [63, 79, 27, 7, 6] - and it can be argued that the use of cross-layer information and enhancements for mobility mechanisms are in the key role towards seamless mobility. The collection and use of that information is discussed more in Chapter 3.

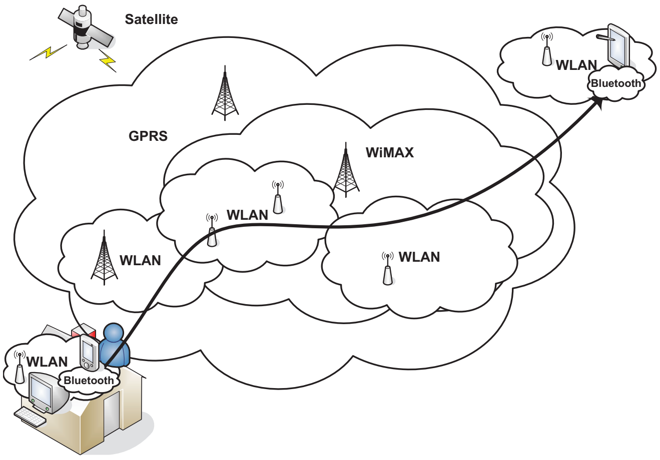


FIGURE 2 Movement of Mobile Node in a heterogeneous environment

2.2 Handover types

There are many types of mobility cases where the point of attachment can change in IP networks. This is why also handovers have been categorized based on the type of the handover. The most important types of handovers, following the terminology defined in RFC 3753 [40], related to this dissertation work are described below:

- Horizontal handover: When a mobile node moves between the access points using the same technology is typically called horizontal handover.
- Vertical handover: When a mobile node is changing the used point of attachment technology e.g. from WLAN to 3G it is usually called vertical handover. Figure 3 illustrates the vertical handover between two different networks.

The definitions for horizontal and vertical handovers are valid in most of the cases, but there are also some exception. For example the change between different WLAN technologies e.g. from IEEE 802.11b to IEEE 802.11g may be called ether horizontal or vertical. One clear difference between these handover types is the change of the IP address. Usually when changing access point with the same technology (e.g. 802.11b to other 802.11.b access point) there is no need to change the IP address used but in vertical handovers (e.g. from WLAN to 3G) typically IP address is changed. When IP address is changed, the handover is also known as layer 3 handover, and when only link layer is involved, the handover is known as layer 2 handover, based on the different protocol layers involved in the handover process.

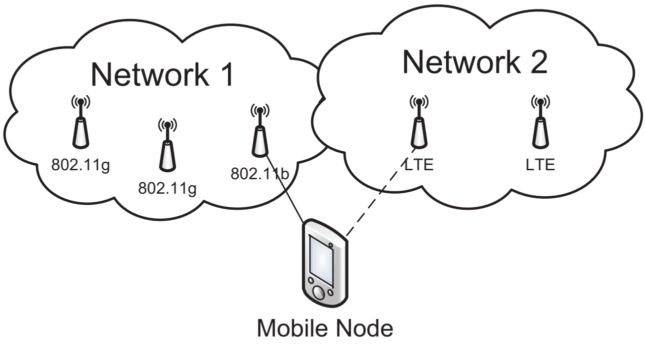


FIGURE 3 Vertical handover

- Proactive and reactive handover: To minimize the effects, like packet losses and/or delays caused by handovers, there is usually signaling in proactive handover before the handover process is initiated. Reactive handover, on the other hand, happens usually suddenly, and no signaling can be done before the handover has actually taken place.
- Seamless handover: This is a type of handover for which the mechanism introduced in this dissertation is meant. In seamless handovers, applications and the user cannot see any changes in the normal operation, and the level of the service is maintained during the handovers.

The terms hard- and soft handover are also commonly used, soft meaning that the connection to the network is not lost during the handover, e.g. when using multiple interfaces, and hard meaning the connection is lost temporarily during the handover. It is important to remember that without any additional notifications the applications see only the change of the IP address during handovers. The change of the point of attachment is not the only reason why mobility mechanisms are needed, there are also other dimensions to mobility. For example, Section 2.3.1 lists 7 examples of these dimensions. It needs to be noted also that not only terrestrial systems need handover support but a similar mechanism is needed, e.g., for satellite systems [16] .

2.3 Ambient Networks mobility management

Ambient Networks project [46, 47] has proposed a scalable, flexible architecture, whose main goal is to provide an appropriate abstraction level so that its functionalities can be used independently of any particular networking operations. Its most important component is the so-called *Ambient Control Space*, which gathers all the control functions that have been designed in order to comply with the identified networking requirements; all these components are structured into *Functional Areas*, which may or may not interact with other peer entities. The Ambient Control Space and some example functions are illustrated in Figure 4. The functions may include support for mobility management, network management, QoS management or multi-access support management.

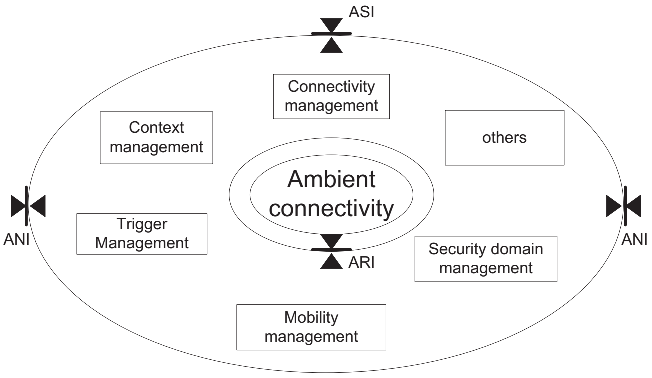


FIGURE 4 Ambient Control Space

Ambient Networks and its control spaces may belong to separate administrative entities and different networks which can collaborate in a co-operative or competitive way. Co-operation between the Ambient Control Spaces and their functional areas is enabled with a composition mechanism. In the composition, two Ambient Control Spaces are forming a new co-operative ambient network which shares the responsibilities of each individual functional area, as shown in Figure 5.

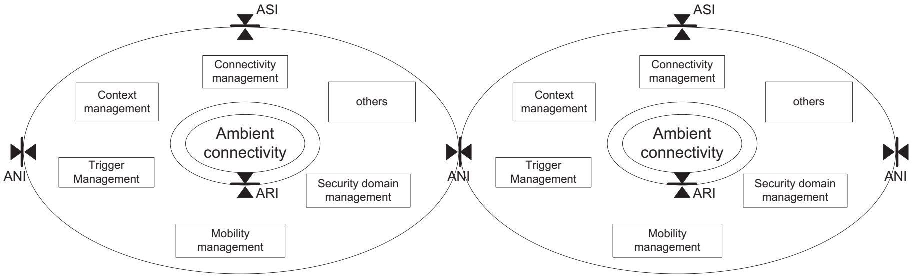


FIGURE 5 Composed Ambient Control Spaces

Ambient networks control space has three different interfaces *Ambient Network Interface (ANI), Ambient Service Interface (ASI)* and *Ambient Resource Interface (ARI)*. ANI allows the cooperation between different Ambient Networks by offering to connect the functions of different ambient control spaces located in different domains (see Figure 5). ASI provides the access to the services in ambient networks. The mechanisms implemented in the ambient control space and the ANI will ensure that external entities see only a single homogeneous control space even though there might be several combined ambient control spaces providing the functions to the application and services. Finally the control space functions interact with resources through ARI e.g. when configuring network resources or setting up end-to-en connections. As a summary, these three interfaces provide simple plug & play connections between ambient networks,enable network reconfigurability, and provide a single interface for the services. [47]

Mobility aspects in the Ambient Networks' interworking architecture were carefully considered: it is suggested that mobility can no longer be restricted to its current scope and that related mechanisms are likely to go beyond the traditional view on mobility based on the idea that not only the devices will be mobile but also many other entities, such as applications, flows, etc.

Functional areas which are related to mobility operations form the *Mobility Control Space*. The Mobility Control Space high level architecture is shown in Figure 6; as can be seen, it comprises the following components:

- **Triggering.** It collects, processes and filters the events that can impact on mobility procedures, and deliver them to the appropriate customers.

- **Handover & Locator Management.** It enables the movement of mobile entities between different locations.

- **Rendezvous & Resolution Service.** It ensures that a correspondent node is always able to locate an AN node, updating its reachability state.

- **Moving Network Support.** It allows managing a set of nodes that move together as a single entity, taking advantage of this.

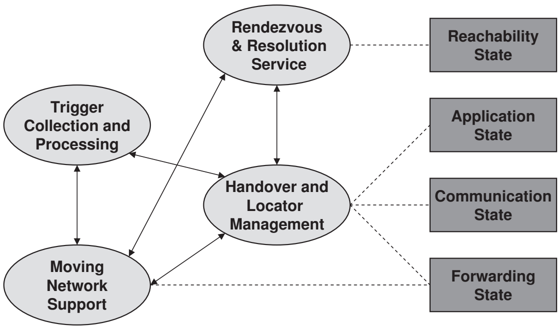


FIGURE 6 Mobility Control Space Architecture

The actual state of the mobile object, in terms of how it can be reached, which identifier it is using, which is the next hop towards it, etc, is represented by the boxes in Figure 6. It is also important to mention that this approach is not restricted to traditional mobility, but it is extended also to mobile applications, flows, etc.

2.3.1 Mobility Dimensions

This dissertation is concentrating on the problems of choosing the best available access during the movement. Mobility may have also other other dimensions than just a change of the point of attachment to the network. In the Ambient Networks project seven different dimensions for mobility were identified [75]. Figure 7 shows examples of these dimensions.

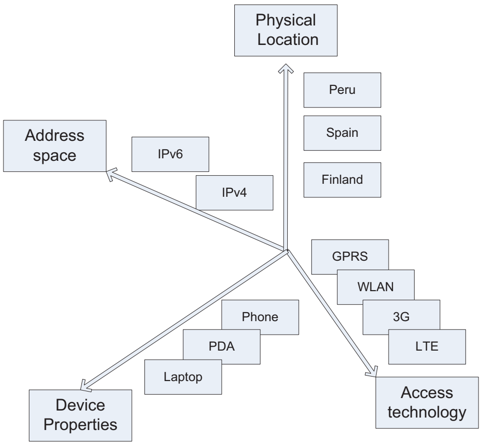


FIGURE 7 Examples of New Mobility Dimensions

- Physical location: A mobile entity moves between access points within the same radio access technology.
- Access technology: A mobile entity moves from one radio access technology to another.
- Address space: A mobile entity moves between networks/devices with a different address space (e.g. IPv4-IPv6)
- Security domain: A mobile entity moves between environments with different level of security.
- Provider domain: A mobile entity moves between networks operated by different provider.
- Device properties: A mobile entity moves from one device to another.

- Time: A mobile entity doesn't move spatially: on-going communication is suspended for a while and resumed afterwards.

2.4 Mobility management protocols

Mobility management protocols aim to enable IP connectivity and keep the location of the mobile node updated. There are several mechanisms and protocols for changing the access between IP networks while moving. This section lists some of the available protocols with a brief functionality description of each.

Probably the best known mobility protocol is the Mobile IP [56]. With Mobile IP, the nodes are able to change the point of attachment to the network and their location can be tracked. There are two different versions of Mobile IP, supporting either the IPv4 or IPv6 protocol. In both of those, the key entity is the home agent (HA). HA is the node which should always know the current location of the mobile node and is located in the mobile node's home network. Home network defines the primary IP address that the mobile node is using for communication. This address is called Home Address (HoA).

When a mobile node moves and gets connected to a new network it needs a way to communicate in that network. In Mobile IPv4 [57], the mobile node uses a new IP address called care of adress (CoA). In Mobile IPv4, the mobile node can get the CoA by using e.g. Dynamic Host Configuration protocol (DHCP) [12] or PPP Internet Protocol Control Protocol (IPCP) [71]. With this CoA, the mobile node is able to communicate in the new visited network. For updating the current location, the mobile node registers the used CoA with HA by using a binding update message. Since the HoA topological location is the mobile node's home network where also HA is located, packets destined to the mobile node HoA are routed to the home network and to HA, which will forward these packets to the mobile node (towards the CoA used) with the help of tunneling technology (IP Encapsulation within IP) [55].

The visited network may also have dedicated entities called Foreign Agents (FA), which can act as a gateway for the mobile node. If FAs are used, they will get all the IP packets destined to the mobile node from HA, and FA forwards the IP packets to the mobile node. When FAs are used, there is no need to allocate CoA for every mobile node; thus the visited network is prevented from running out of available IP addresses. However, the use of FA may lead to a situation called triangular routing. This is because MN is able to send the outgoing packets to other nodes (correspondent node CN) directly via FA while getting the incoming packets via HA and FA. These different routing paths for outgoing and incoming traffic can be avoided by using the reverse tunneling mechanism [44] where all the packet go through HA. Mobile IPv4 topology is illustrated in Figure 8.

Mobile IPv6 protocol (MIPv6) [28] handles mobility in IPv6 networks. Mobile IPv6 benefits from the functionality of IPv6 and from the experience learned from the Mobile IPv4 protocol. The MIPv6 topology is illustrated in Figure 9.

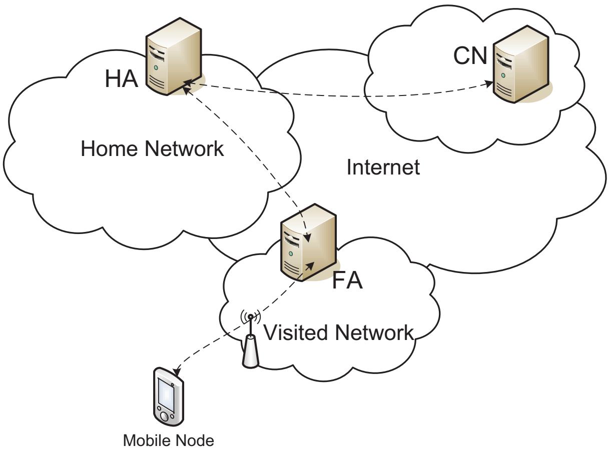


FIGURE 8 Mobile IPv4 topology

MIPv6 still has HA, and the location is always updated to HA by using binding update messages. The mobile node can get CoA from the visited network by using the IPv6 neighbour discovery and the address auto-configuration mechanism and it does not need any support from the visited network routers. MIPv6 doesn't use FAs either, and the problem of triangular routing is solved by using the route optimization mechanism. With this mechanism, the mobile node is able to register current CoA directly with the other nodes (Correspondent Node), which enables direct IP flows between the two nodes without routing them via HA. Similar route optimization is possible in MIPv4 as well, but in MIPv6 route optimization is not an extension to the protocol but a fundamental part of it. MIPv6 can also use IPv6 headers for routing instead of the tunneling mechanism, which will reduce the overhead of sent packets.

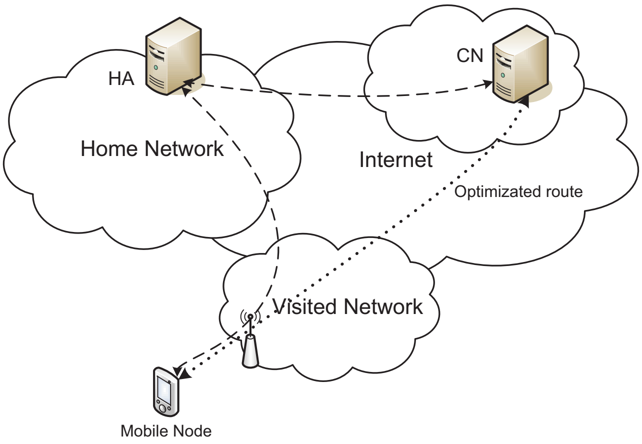


FIGURE 9 Mobile IPv6 topology

To support both IPv4 and IPv6 a dual stack support for Mobile IPv6 [69] has been introduced. With this mechanism Mobile IPv6 support can be extended also to the nodes still using the IPv4 protocol.

Fast handovers for Mobile IPv6 [31] is one mechanism based on the mobile IP. In Fast Mobile IPv6, a router located in the access network may work as a proxy for the mobile node. During a handover, the previous access router (PAR) acts as a proxy for the mobile node and uses tunneling to route the packets to a new access router (NAR) in a new access network, see Figure 10. The advantage of fast mobile handovers is the support they provide for routing the data to the new upcoming point of attachment before the connection will break due to the handover. This obviously reduces packet loss caused by the handover and helps the mobile node maintain higher Quality of Service level.

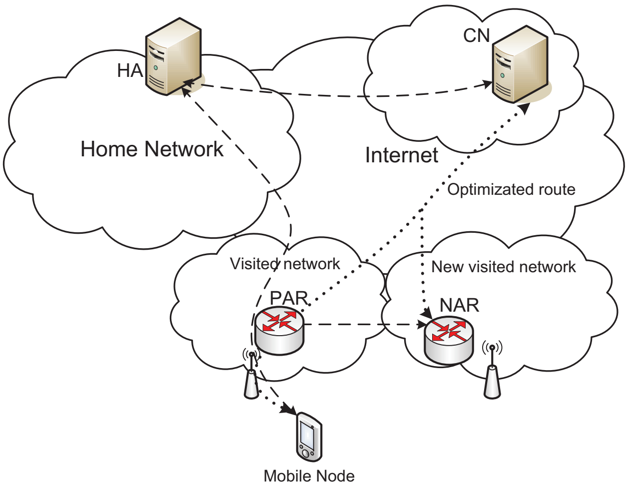


FIGURE 10 Topology of fast handovers for Mobile IPv6

Hierarchical Mobile IPv6 [70] is an extension to MIPv6. It introduces a new entity called Mobility Anchor Point (MAP) that acts as a local home agent for the mobile node. When the mobile node moves inside the MAP domain, signaling from the Mobile node is needed only with the first MAP, which is expected to be closer than the Home Agent. Thus the signaling time can be reduced and MAP can help MIPv6 to achieve seamless mobility [81]. An example topology for Hierarchical Mobile IPV6 is shown in Figure 11 . MAP can get all traffic destined to the mobile node from HA and will forward the packets to the mobile node. Route optimization is also supported and can be done in two ways. The mobile node can optimize a route directly with the correspondent node or with MAP. In this way, the traffic does not have to go via HA.

Proxy Mobile IPv6 [19] is designed to provide the network based IP mobility support, and tries to minimize the signaling between the mobile node and home network. The main idea is that the network entities will track the mobile node movement and trigger the necessary action for the handover without having any involvement of the mobile node in the L3 signaling. Proxy mobile IPv6 introduces an entity called Mobile Access Gateway (MAG) that is responsible for detecting

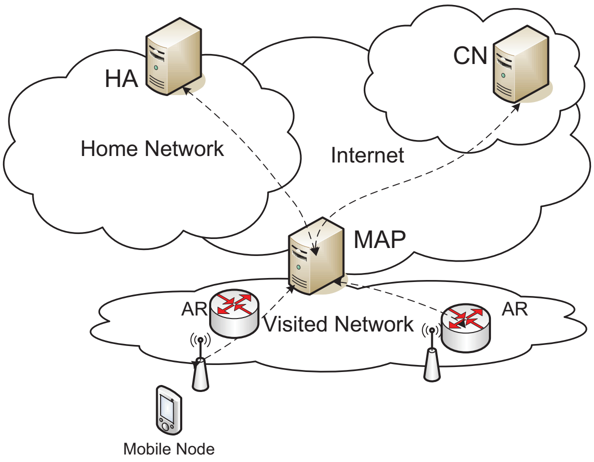


FIGURE 11 Topology of Hierarchical Mobile IPv6

and managing the mobility of mobile nodes inside a proxy Mobile IPv6 domain. MAG is responsible for handover signaling on behalf of the Mobile Node. MAG updates the location of the mobile node with the Local Mobility Anchor (LMA) that acts as a topological anchor point to mobile nodes (similar to MIPv6 HA). A data flow destined to or from mobile nodes will go via LMA, which routes the packets to the mobile node through MAG. LMA has the same functionalities as HA in Mobile IPv6. There may be several anchor points in the Proxy Mobile IPv6 domain responsible for keeping track of different groups of mobile nodes. The basic topology of Proxy Mobile IPv6 is shown in Figure 12. Proxy mobile IPv6 supports both IPv4 and IPv6 or dual-stack (IPv4/IPV6) protocols and it has also support for multihoming.

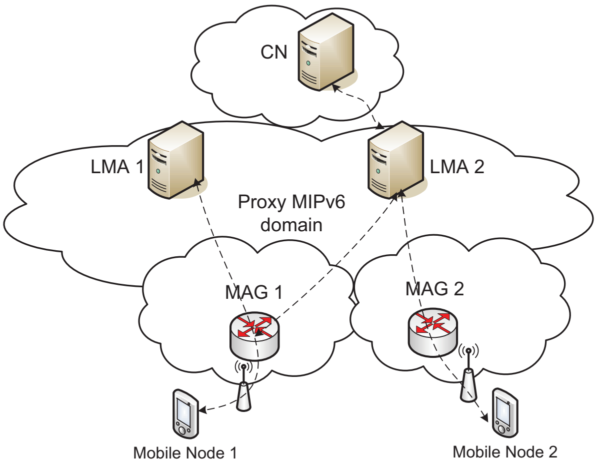


FIGURE 12 Proxy Mobile IPv6 topology

One interesting protocol is the Host Identity Protocol (HIP) [45, 48, 20] that also has a mechanisms for mobility support. HIP proposes a new Host Identity layer between the network and transport layers. With this layer it is possible to map transport connections to Host Identifiers (HI) instead of IP addresses. Unlike IP addresses, Host Identity stays constant if a host is changing the network. Host Identity is a unique public cryptographic key of a public/private key pair. The use of cryptographic keys is also the security cornerstone in the HIP architecture together with the IPSec Encapsulating Security Payload that is used for data traffic between HIP nodes. One of the key mechanisms is the HIP base exchange, which is used to generate and exchange the HI keys for communicating. For location updating, HIP nodes can use Dynamic DNS or a special rendezvous server, see Figure 13. In practice, HIP nodes keep updating their current IP addresses, which are mapped to the HIs. HIP can enable mobility across IPV4 and IPv6 networks and also supports simultaneous multiaccess.

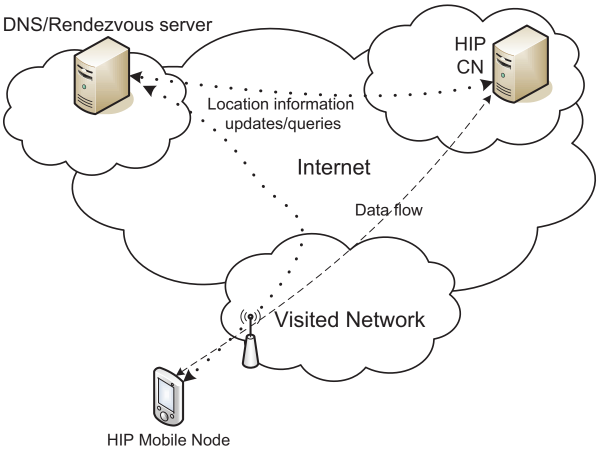


FIGURE 13 Example topology for HIP protocol

Also a whole network can be mobile. For network mobility support there is the network mobility protocol (NEMO) [10]. The basic idea of NEMO is to use a gateway called mobile router (see Figure 14)) to provide Internet access for a moving network. Only the mobile router will change its point of attachment to the Internet. NEMO is used to hide mobility, and other nodes see the moving network as a normal sub-network and use the Mobile Router as a default gateway. One good example of a moving network is the network inside a moving train.

Even there are a number of mechanisms and protocols available to enable IP mobility. There is still a number of drawbacks, such as high latency or packet loss, caused by the procedures that are needed during the handover execution phase [22]. In addition to the protocols described in the previous section, there are also dedicated solutions introduced which may be suitable e.g. for smaller scale networks. One example is the DRiWe protocol described in Section 2.5.

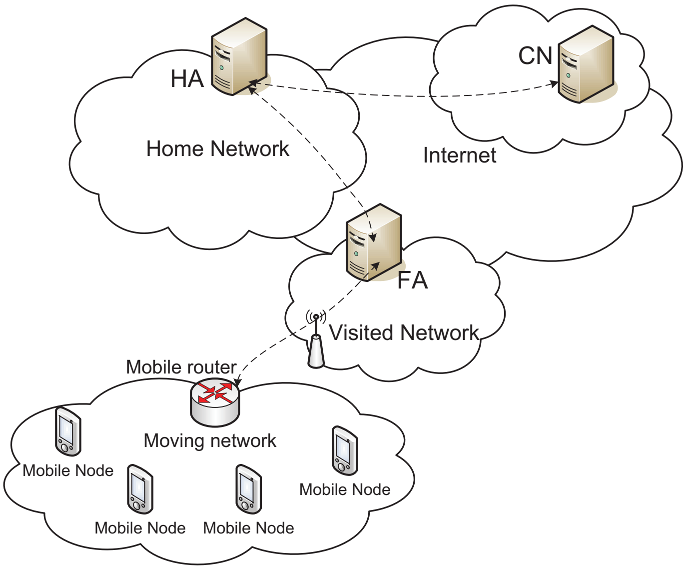


FIGURE 14 Example topology of Network Mobility

2.5 DRiWE mobility protocol

This section introduces the DRiVE protocol for handling the movement in IP-networks [PI, 37]. This protocol introduces mechanisms for handling the problems of routing for the mobile devices. The protocol considers only the mechanisms for handling the OSI Layer 3 movements. The DRiWE protocol is also one of the very first achievements in the author's research, forming a part of a wider mobility management research.

2.5.1 DRiWE protocol functionality

The DRiWE protocol is based on the idea that routing decisions take place in routers. Those routers, which participate in routing data flows for mobile nodes, know the location of the mobile node in question. Besides normal network based routing information, the routers use host-specific information about mobile nodes in their routing tables. The protocol also introduces mechanisms to avoid any gratuitous growth of the routing tables. The routing table's growth is controlled by allowing the dropping of the routing information of the mobile node and getting it by a dedicated mechanism if needed. The protocol also includes an advertising mechanism, so that routers can propagate the information about the location of mobile nodes to other routers.

To achieve a soft handover, the mobile node uses multiaccess functionality where two interfaces are used to communicate with both access routers (AR1 and AR2, in Figure 15 at the same time during the handover. The connection to the new access router (AR2, in Figure 15) is formed before the old connection breaks.

During the handover, the mobile node accepts incoming packets from both of its interfaces. Therefore, there is no need to stop the data flow at any time. Both interfaces are used only during the handover so that the current connection is used until the new connection is totally established and ready for the data flow. [PI]

2.5.2 DRiWE protocol handover mechanism

The DRiWE handover mechanism must be supported by the access routers, mobile nodes and also by the intermediate routers between access routers. The correspondent node that communicates with the mobile node does not need any support for the protocol or for the handover mechanism. The intermediate routers need to support it because of the behavior of the introduced update mechanism (see Section 2.5.2.1). In the example scenario in Figure 15, the mobile node is attached to AR1 and it is moving to the AR2's access network. When a mobile node is going to connect to another access network it has to complete a Layer 3 handover to enable the IP traffic. For enabling the connection in the foreign network the mobile node needs a temporary IP-address (care-of address) from the new access router (AR2, in Figure 15). The care-of address query takes place through the current access router (AR1, in Figure 15).

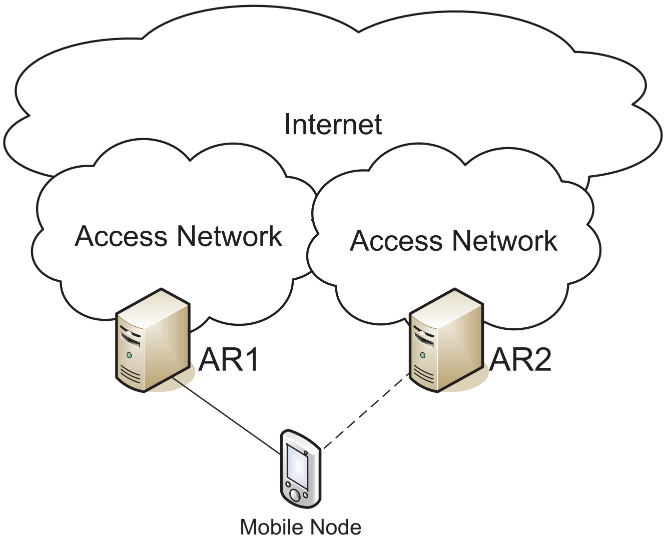


FIGURE 15 DRiWE protocol handover mechanism

To enable the use of the HoA (the Home Address) of the mobile node as a source address for the outgoing packets, tunneling is used between the mobile node and the access router. The packets go tunneled to the access router by using the care-of address as the source and the IP address of the access router as the destination for the tunnel (outer) packet. The access router decapsulates the packet and sends the original packet to the destination with the source address of the mobile node's HoA. Packets destined to the mobile node are routed to their destination by using the HoA as the destination and will be routed to the current access router. The access router knows all the mobile nodes that are currently attached to its network and will deliver the packets to them.

After the mobile node is attached to the new access network and the new access router (AR2, in Figure 15) the updating of the new location of the mobile node at the previous access router (AR1, in Figure 15) starts. When the previous access router (AR1, in Figure 15) gets the information about the new location of the mobile node it starts to forward packets destined to the mobile node to the new access router (AR2, in Figure 15). After the successful update of the location between access routers the mobile node changes its default outgoing route to go through the tunnel towards the new access router (AR2, in Figure 15) and the handover has successfully finished. Figure 16 shows the needed messages for getting the care-of address (CoA) and updating the location in a successful handover.

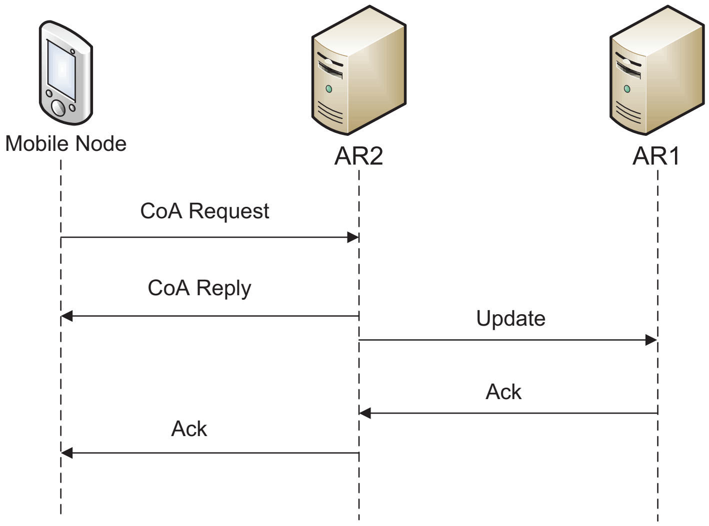


FIGURE 16 DRiWE protocol handover signaling

The DRiWE protocol introduces a mechanism for keeping connections up between mobile nodes and access routers even when there are no packets to send or receive. It is done by the exchange of timely Keep-Alive messages. The mechanism gives information about link breaks to the mobile node. Without this, when there are packets to send, the broken connection may not be detected for quite a long time, e.g., when using User Datagram Protocol (UDP) [61] for data delivery. The mobile node can also be prepared for situations where it is ping-ponging between two access routers or needs to use the older connection again, especially when the current connection is lost and the older connection is still available. The use of this mechanism should be very carefully considered as it can generate extra traffic in the network or increase power consumption of the mobile node when trying to keep the interfaces up.

2.5.2.1 DRiWE location update mechanism

The behavior of the mechanism used to update the location of the mobile node between access routers needs the update message to be handled at every router between the access routers. This is necessary because those intermediate routers have to update their information about the location of the mobile node accordingly. Another solution for sending the update messages is to use tunneling between the old and new access routers to forward the packets between them (AR1 and AR2, in Figure 15). The DRiWE protocol introduces mechanisms, which allow routers to acquire the location of mobile nodes when needed to track down their current position for routing purposes. The Update message is sent to the next router on the route to the destination (AR1, in Figure 15). The first intermediate router adds the HoA of the mobile node included in the message to its routing table and the next hop will be the router from which this message was received. Each intermediate router between the access routers will do the same. When the update message reaches its destination (AR1, in Figure 15) the access router changes its routing table and starts forwarding the packets to the mobile node through the new access router (AR2, in Figure 15). The update message is also acknowledged to the sender and to the mobile node. If the sender (AR2, in Figure 15) doesn't get the acknowledgement, it sends the information about erroneous updating to the mobile node and the recovery mechanism will be started. The recovery mechanism recovers the location information in the affected routers back to the form it was before the update. The recovery works like the update mechanism but is started from both access routers (AR1 and AR2, in Figure 15). So the recovery will propagate from both directions and will recover the situation back even when there is a link break between the access routers.

2.5.3 Implementation and evaluation

Tests were conducted in two types of environment. First the protocol was tested in a small wired test network shown in Figure 17. Ordinary PCs, with Linux as their operating system, were configured to work as routers. The mobile and correspondent nodes were also ordinary PCs. The routers and the correspondent node were computers with Intel Pentium4 2 GHz CPUs, 256 MiB RAM, Samsung 10 GiB HDD, Intel PRO/100 VM and three 3Com PCI 3c905C Tornado network cards. The mobile was a computer with Intel Pentium3 733 Mhz CPU, 128 MiB RAM, Samsung 10 GiB HDD, Intel PRO/100 VM and 3Com EtherLink XL 10/100 PCI TX (3c905B) network cards. The correspondent node ran Windows XP and all others ran Linux as their operating system because the protocol had to run on them (it was implemented for Linux).

The test was made by using a ping and ssh connection between the mobile and the correspondent node. In Figure 17, AR1 and AR2 were functioning as the access routers. R2 and R3 were used to test the functionality of the intermediate routers in the handover. The handover was simulated by plugging the cable of the mobile node out of the old access router and plugging it into a new one. There

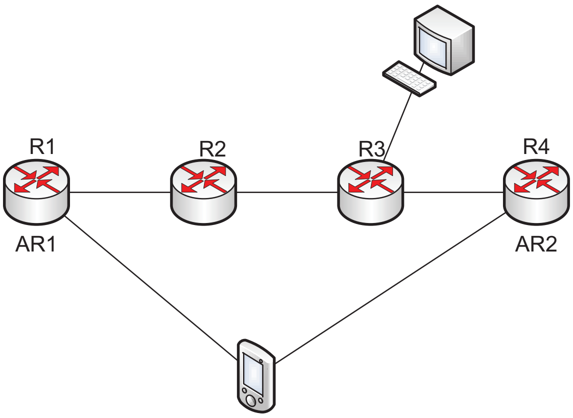


FIGURE 17 DRiWE testbed

was a controller computer connected to all the nodes via a switch. This computer was used to connect to the mobile node and instruct it to perform handovers to routers (specified by their IP address). During these tests no extra load, other than the ping packets and ssh connections between the mobile and correspondent nodes, were introduced into the network.

The test consisted of starting a ping command or an ssh connection on either the mobile node and the correspondent node as the destination or vice versa. Two different scenarios were tested. In the first scenario, the correspondent node was directly connected to the access router (AR1) that served as the initial starting point of the mobile node. Handover took place from AR1 (R1) to AR2 (R4). During the handover, no packet loss was reported by ping nor did the ssh connection brake. For the second scenario, the correspondent node was connected to one of the intermediate routers (R3). The main reason for this scenario was to test the update recovery mechanism that deals with handover failures occurring during location updates. The protocol on R3 was stopped during the handover and restarted after. During the handover, the ssh connection did not break, and the recovery mechanism functioned properly.

The same scenario (topology) was used to test continuous movement of the mobile node. The mobile node moved from R1 to R2 to R3 and finally to R4. In general, during handovers no connections were lost even when the active link was plugged out. The protocol recognized it and went back to the previous access router without a problem. Handovers took 5.3 milliseconds on average.

2.6 Comparison between MIPv6, FMIPv6 and DRiWE protocols

Many mobility solutions count on MIPv6 [28], though it is known to be unable to satisfy seamless handovers without any extensions due to the procedures it has to accomplish during its handover phase. Besides the mechanisms to ensure that the IP addresses used are unique on the link (Duplicate Address Detection) and authorizations, it needs to send binding update messages to each of its correspondent nodes (CN) in order to update their knowledge of the actual location of the Mobile Node (MN). As long as the CNs are not updated they continue sending packets to the old Care of Address (CoA) of the MN, thus resulting in possible packet losses if that CoA is not used by the MN any longer. The MN has to register with its Home Address (HoA) too. In the DRiWE protocol, during a handover only one update message is sent out from the new AR to the old AR. This allows the MN to be accessible at the new AR immediately. It is because it can be assumed that it was reachable at the old AR and with the update message it updated the routing information of the old AR to forward packets for the MN towards the new AR. The idea is similar to the fast handovers for Mobile IPv6.

Due to the update mechanism of the DRiWE handover, the MN may not be accessible on the best path from a CN. In MIPv6 it is accessible because every CN knows the CoA of the MN, so they can send packets to it using the best path (according to their and subsequent router's routing tables). In DRiWE, traffic is not routed to the MN by its CoA but by its HoA. It means that after a handover the MN is accessible through its old AR (and the routers between the old AR and the new AR, because they became updated by the update mechanism). For the MN to be accessible on optimized routes it is necessary to update the routers that store information about its previous (or older) location. For this a routing protocol, such as Routing Information Protocol (RIP) [21], which is initiated by the MN through its current AR, can be used.

In MIPv6 the Home agent (HA) is acting as a Proxy for Neighbour Discovery messages destined to the MN. It is used to make the local nodes in the home network to send packets destined to the MN to the HA's MAC address instead. The same mechanism is also needed for DRiWE for the same reason.

Timers for bindings in CNs and HAs are used by MIPv6 to drop unused bindings from their binding caches. In DRiWE, routers can drop unused routing table entries that correspond to MNs. It can be done either when the routing table size reaches a certain size or by maintaining a timer for unused entries (or both). Keep-alive messages are exchanged between MNs and ARs to keep the MN registrations alive. If the MN does not get keep-alive messages from the current AR in a specific amount of time it tries to change its AR back to the previous one. This is possible because the MN keeps alive its registration to the previous AR.

MIPv6 supports the discovery of the HA by an MN by means of Dynamic Home Agent Address Discovery. This mechanism can be used to DRiWE too. The Home Address Option is used by MIPv6 to inform the recipient of the mobile node's home address and to avoid the tunneling of packets from the MN to CNs. For traffic from CNs to MNs, the type 2 routing header, containing MN's HoA to which the MN's applications are bounded, is used. In DRiWE, tunneling is used to send packets from the MN to CNs. DRiWE uses a tunnel between the MN and the AR when sending packets to CNs outside the visited network. It means that these packets will have the MN's HoA as their source when reaching the CN.

There is an issue when sending packets to CNs located in the same visited network as MN, because in that case the MN would send them using its CoA as the source address, even though the CN might be waiting for a packet from the HoA of the MN (in the case when the CN wants to talk to the MN at its HoA). Therefore, it is assumed that CNs should not use the CoA of the MN. There is no real need for a CN to send packets to the MN using its CoA in DRiWE. The CoA is considered to be known only by the respective access router and the MN. Another assumption is that all the packets sent by the MN have the HoA as the source (they go through the tunnel between the MN and the access router) and the MN is allowed to use its CoA for communication only with the access router. This way CNs always talk to the MN at its HoA. In MIPv6 this is solved by manipulating packets in Layer 3 of the network stack and using Home Address Option and type 2 routing header. Thus all the packets sent to CNs with bindings are seen at their destination as coming from the HoA of the MN.

For the FMIPv6 [31] protocol to work there must be a router in the currently visited network that may work as a proxy for the MN. This proxy router is used to tunnel packets arriving to the MN's CoA in its network to the new AR in the new visited network. As mentioned in Section 2.5.2.1, the update mechanism in DRiWE can be implemented by using tunneling between the old AR and the new AR, thus making the old AR to tunnel packets, which are destined to the MN's HoA, to the new AR. This mechanism is more or less equal to the functioning of FMIPv6. Both protocols have to tear off the tunnel after some time: FMIPv6 when the MN has finished with updating the CNs, and DRiWE after the normal update mechanism (modifying/adding location information of the MN in the old AR and the routers towards the new AR).

The MIPv6 L3 handover starts after the MN has detected movement (already moved) to a new network (by receiving Routing Advertisements with unknown prefixes). For a DRiWE handover to start, the MN has to know the IP address of the AR beforehand. The same applies to FMIPv6. MIPv6 by default supports MNs with one interface. DRiWE needs at least two interfaces (multiaccess support) of the same type. These two interfaces must be of the same type to provide a smooth handover (one is used while another one is being configured). FMIPv6 supports one interface by default, although to receive L2 information about the new AR while still connected to the old one it may need a second interface.

DRiWE implements fast and smooth handovers. Fast means that the handover needs less signaling than handovers for MIPv6, and smooth means that no or a minimal number of packets get lost during the handover thanks to the use of multiple interfaces. Even though the handover is faster than that of MIPv6's, the routes for reaching the MN after a handover may not be the optimal ones. For this an additional location advertisement is necessary (MN routing advertisement). In MIPv6, CNs have knowledge about the exact location of MNs by binding update messages. In DRiWE, the knowledge in routers about MNs location may not be optimal. This is caused by the way the MN's location is updated during handovers. That is, to provide routers with exact location information of an MN, the MN has to advertise its location (MN routing advertisement). This advertisement can be matched with the sending of binding update messages in MIPv6.

FMIPv6 provides fast handover for MIPv6 by reducing the necessary number of signaling during the handover process until the MN regains IP connectivity at the new AR. But after the fast handover the MN needs to send BU messages to CNs in order to use optimized routes. Therefore, it has a similar kind of handover style as DRiWE. It has two phases, the first phase is to provide fast establishment of IP connectivity and the next is to "build" optimized routes to the MN. Although the second phase of FMIPv6's handover may be faster in large networks (with lots of routers) than that of DRiWE's, DRiWE may need less signaling in small networks with lots of CNs.

In FMIPv6, when the new AR receives the Handover Initiate message from the old AR, it starts to defend the new CoA of the MN until the MN arrives at the new network. The same kind of mechanism is needed in DRiWE for the same purposes as in FMIPv6. That is, to defend the new CoA from being used by another node in the network of the new AR until the MN arrives there. This means that the new AR must work as a proxy for neighbour discoveries for the MN's new CoA.

The DRiWE protocol was designed with tools existing for user space protocol implementation and because of this there are some shortcomings which could be solved by redesigning the protocol to work more tightly in Layer 3 (now only the routing table and interface address modifications are done using netlink sockets and ioctl calls in Linux). Security has not been designed into DRiWE yet, though it is obviously needed. For example, only authenticated MNs must be allowed to register to ARs and to instruct them to start e.g. the update mechanism. For protocol messages sent between ARs and routers the same security considerations may apply as for normal routing protocols. DRiWE might not be as scalable as standardized protocols such as Mobile IPv6, but is suitable for smaller scale networks.

2.7 Handover decisions

When moving in the area of heterogeneous IP networks, the selection between different networks that may be used is needed. Usually, this is based on the information about the network availability which is the main criterion. There is also other criteria which may trigger the handovers like supported QoS or the cost of using certain networks. This trigger information and the collection of it is discussed more in Section 3. Whatever the trigger information might be, there needs to be some intelligence in the network nodes to enable the decision on handovers. The decision on handovers can be made in the mobile node or it can be network initiated and in an optimal case it will be a distributed decision between the mobile and the nodes in the network side. In mobile devices, the decision depends much on the services and applications that the user wants to use. In the network side the load of the network may be triggering the handovers. In the following section, the problem of decision-making is discussed with the description of work accomplished when defining the vertical handover VERHO system [PII]. The VERHO system is an entity in the mobile node, making the decisions about which available access technology should be used.

2.7.1 VERHO system for mobile nodes

The goal of the VERHO work was to design a system that realizes intelligent decisions based on the available information, controls the MIPv6 handovers and achieves seamless vertical handover by performing a proactive handover. Other research such as [80, 4, 7, 79] have also promoted the use of several information sources, and it can be argued that mechanisms for handover decisions are necessary. This kind of a system itself can provide the seamless handover by not only controlling the handovers but also triggering, e.g., the search of new possible access points.

The purpose of the VERHO system in the mobile node is to allow also the users to present their own demands for the connection. For example, users can predefine the cost of each technique in terms of time as one of the decision parameters. Additionally, things like power consumption in the mobile device should be minimized, which can be another parameter. The applications have their own demands for, e.g., needed bandwidth or maximum delay of the link. The most significant information is of course the availability of the link and access network. But, as described, the information may be originated in different protocol layers (cross-layer information). Figure 18 illustrates the VERHO system position with different layers of information. The main idea is that valuable information for the handover decision can be collected from different sources, and based on this decision, the handover process, provided with the used mobility protocol, will be initiated. The cross-layer information used can be provided via the trigger management framework [PV].

The system is working completely in the mobile node, and it does not necessarily need support from the network side. The original idea of the VERHO system was to use the information that is available only in the mobile node side. However, further studies have showed that also network can provide valuable information. For example, the congestion level in an access point is very valuable information and may be the trigger for handovers as presented in [PVIII, 24].

2.7.2 Decision process in VERHO system

The decision-making in the VERHO system is based on predefined rules compared to the properties of the interface of the available technique. The rules may be divided into critical and non-critical rules. The critical rules are those rules that have to match before the interface is enabled for the decision process. The necessary critical rules are based on the state of the interface and link (Link Up

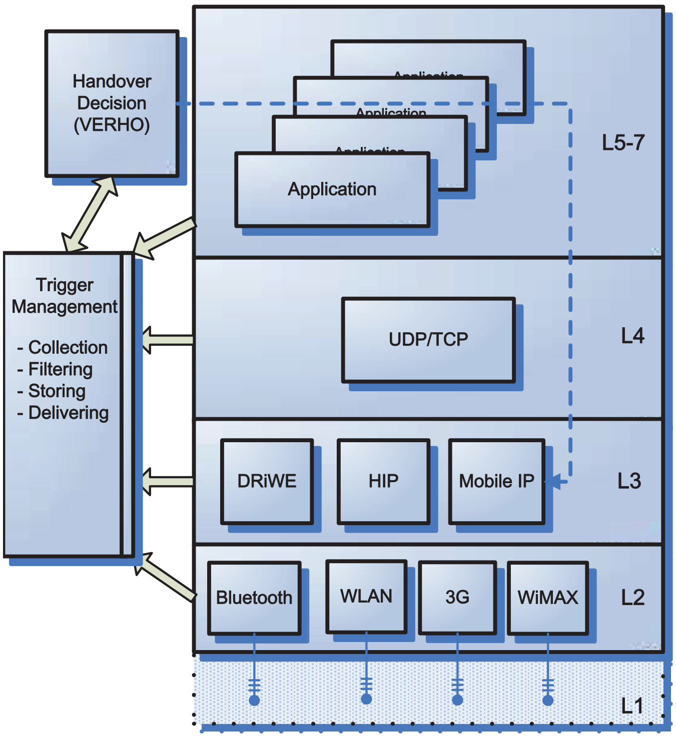


FIGURE 18 Cross-layer architecture

and Link Down). The system also provides a possibility for the user to give some critical rules concerning, e.g., the type of the interface or even the maximum cost of the interface to be used. If the requirements of the critical rules match, the interface will be available for the decision. A rule can be like this: Interface state = UP AND Interface type = IEEE 802.11b. This rule means that the user wants to use WLAN 802.11b interface if possible.

The implementation of decision mechanism may be rather simple when the decision is based just on a numerical preference value. In this simple case, each interface will get a calculated preference value based on the values of different parameters. The parameters have weighted numerical values so that the most significant parameters will have greater weight. Each interface has several parameters such as cost, power consumption, support for wireless QoS or bandwidth. If a parameter is lacking in a specific interface due to its different technology, that parameter can have a default constant value. The value of the preference number is calculated from each interface's parameter values. The measured QoS parameter's value can be one group of parameters. The QoS parameter value can be an IP-level End-to-End quality measured with a separate software tool like QoSMeT [64]. Each measured parameter - e.g., delay, RTT and jitter - can then have the weighted value for the calculation of the preference value of the interface. There are also parameters like the type of interface without any numerical value. Those parameters can be assigned with a default numerical value. Deriving of the value can be based also on the user demands or, e.g., on the speed of the connection. By following the current state of a specific interface's properties, the system is able to choose whether there is another interface available that matches better the specified rules. But as mentioned the presented way of using the calculated preference value is just one simple example of deriving final decisions and does not exclude the use of other more advanced algorithms with the limits of the device's processing power.

2.7.3 Information sources used in VERHO system

The VERHO system uses the information deduced from the decision process and controls the MIPv6 handover. The decision when handover should occur depends on the information collected from each available technology and data flows in situations where several different access technologies are available. The most significant information is the information received from the link layer, e.g., link up/down. The link layer offers different sets of information depending on the technologies available. Using the link layer information also improves the Mobile IP handovers. The system may also use several other sources of information, e.g., user specified information and the Quality of Service requirements of applications.

The VERHO system is mainly making the decision about which one of the interfaces available should be used. But by using the link layer information, the system can make the decision and control, e.g., Mobile IP handovers in a more sophisticated way. It has already been shown that the use of link layer information improves the Mobile IP handover, e.g., in [4]. The paper in question introduces a policy- based handover decision based on the link layer information. Their solution introduces a generic link layer between layer 2 and layer 3 and shows that a system with link layer information outperforms Mobile IP handovers.

The VERHO system uses two triggers, Link Up and Link Down, for triggering the handover. The Link Up trigger event corresponds to the establishment of a new Layer 2 link, which allows IP communication over it. This is typically a new connection between the Mobile Node and the Access Point. The Link Down trigger event corresponds to a layer 2 link that has been broken down. This typically occurs when a current connection between the Mobile Node and an access point has been terminated. The interface, which generates this trigger, cannot be used for communication until a connection with an AP is made (Link Up). There are also many Layer 2 parameters that can be used as hints, for example, signal strength, signal to noise ratio, interference, etc. By using these triggers and hints it is possible to make a proactive handover that allows the creation of a new connection by using another available interface before the current connection breaks. It need to be noted also that it is not only the handovers and mobility protocols that benefit from the triggers and hints but also other protocols and applications may get benefits. For example, the studies such as [9] have shown that Transmission Control Protocol (TCP) congestion control [3] may react more rapidly and avoid unnecessary congestion situations when it gets the information about the new upcoming link and changes in the used end-to-end path used. Also multimedia applications [30, 59] benefit from it, for example applications may start adapting or changing their codecs in a proactive way if they get the information about the upcoming changes on the link after the handover.

The information that can be used as hints for the upper layers much depends on the used hardware and software used. The RFC 4957 [32] describes the behaviors of GPRS, 3GPP2 (cdma2000) and IEEE 802.11. It suggests that Link Up and Link Down triggers can be created based on the PDP context activation/ deactivation in GPRS and bringing down/up the PPP connection in 3GPP. In IEEE 802.11 it is suggested that a Link Up trigger can be created when the Mobile Node receives a (Re)Association Response message. The Link Down trigger of IEEE 802.11 is created if the Mobile Node gets a De-authentication message or a Disassociation message from its access point.

The behavior of the IEEE 802.11b and Bluetooth technologies to create triggers were studied for the purposes of VERHO. The system needs two different mechanisms to get the information from layer 2. One way to get the information is by listening events that layer 2 techniques offer. This is the way how information is gathered, e.g., in IEEE 802.11. The Link Up and Down triggers of IEEE 802.11 for the VERHO system can be created based on the signaling messages between the Mobile Node and Access Point, in a way used in [32]. The Link Down triggers should be created for VERHO system also if the link is put down from the Mobile Node side.

Another way is to use polling for those access techniques that do not generate events or triggers. The system will use the polling mechanism and monitor the state of the interface for the Link UP and Link Down triggers. The polling mechanism can be used also for getting the values of the link layer parameters from drivers. Those parameter values can be taken into account, e.g. when calculating the preference values of a specific interface. For example, in Bluetooth polling is needed and the creation of Link Up or Down triggers is based on the available connection state parameter.

Signal Strength is considered to be a reliable hint that can be used for triggering the handover. For example, if the signal strength of an interface that is being used falls below a defined value and there exists another interface with an acceptable level of signal strength, the system should switch over to that interface. The change of the signal strength can also trigger other mechanisms like the scan mechanism in IEEE 802.11 for finding a new possible access point.

Information collection from various sources can be done by using the triggering framework (TRG) described in Section 3 and illustrated in Figure 18. When the VERHO system was designed TRG was not available and other mechanisms, such as Link Layer Provider (LIP) described in [PIII], were used directly with the VERHO system for validation purposes. The LIP, as it is designed, is capable of extracting and collecting link layer information from terminal interfaces. Even the LIP can have an own interface with the VERHO system for providing link layer information. It is possible to deliver this information also via TRG using just a single trigger format and interface for getting the trigger information.

The integration of LIP with TRG and VERHO is easy since LIP can act as one of the producers in the TRG framework as described in Section 3. What is most important, however, is that the design of the VERHO system is such that it can support or can be easily modified to support possible future systems. The future systems and extensions to VERHO, possibly on the network side, may provide the handling of the authentication, security or the mechanisms for better Quality of Service.

2.8 Summary

This chapter discussed the mobility management in heterogeneous IP networks. There was a brief description of different mobility protocols, including that of the DRiWE protocol, which forms a part of the author's research. A concept for vertical handover called VERHO that supports intelligent decision algorithms was introduced as well. For collecting mobility related information and feeding mobility management related decision processes, a triggering framework (TRG) was suggested. It can be argued that the introduced components, VERHO and TRG, together with a suitable mobility protocol (for example DRiWE or Mobile IPv6) provide the way to handle mobility in IP networks in a more optimized way. The next section further discusses the mobility event collection and distribution and gives a more detailed description about and evaluation of the suggested triggering framework.

3 MOBILITY EVENT COLLECTION AND DISTRIBUTION

This chapter addresses one of the key challenges in mobility management, i.e. mobility event collection and distribution and also presents the background and the main results of the publications PIV,PV,PVII and PVIII.

Managing different events and triggering information originating from a network and from different layers of the protocol stack is very challenging due to the variety of information. Mobile devices are capable of running demanding network applications and have support for multiple access technologies. In a heterogeneous network environment it is mandatory to get the necessary information for triggering handovers. In their current state without any extensions, the mobility protocols can handle only a small amount of this information.

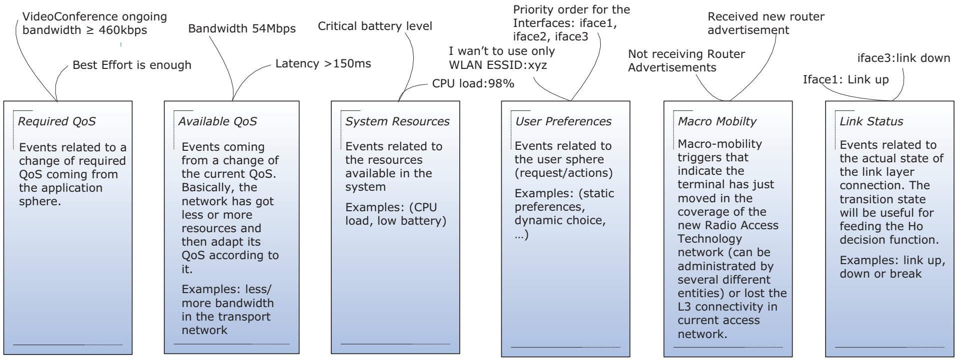
A significant part of this dissertation work was to design a framework that supports event collection and processing, and trigger distribution possibly from hundreds of different sources. Also other studies, such as in [80, 4, 7, 79] claim that in heterogeneous network environments several sources of events and context information should be consulted in order to achieve seamless connectivity and develop swift mobility management mechanisms. Furthermore, related work in other event/notification systems [35, 13, 42], which introduces mechanisms on how to implement such systems, along with the evaluated event generation cases was very encouraging and complimentary to effort in defining the trigger management framework (TRG) [PV] as a specialized notification system for mobility-related events which originate from the entire protocol stack.

TRG is designed to be open, flexible, with a low overhead, and incrementally deployable. TRG is capable of handling a large set of notifications and enables efficient use of cross-layer information for mobility management. Notifications may originate from any mobility related entity, not only from the lower layers of the protocol stack (physical, data link, and network) but also from the upper layers. Typically, notifications are conventional mobility events, such as the availability of a new network access, received signal strength indications, network capacity load, but can be also higher level events, such as security alerts, policy violations, end-to-end quality of service deterioration, and network access cost changes. TRG can act as a part of the mobility management, together with the mobility management protocols and applications as described earlier in Section 2.7.1.

3.1 Mobility triggers

After surveying the literature (see, for example [63, 14, 79, 27, 7, 6] and based on the author's and his colleagues expertise, it was possible to identify a large number of different types of network events related to mobility management. Triggers can be clustered, regardless of the underlying communication technology, based on groups of events related to changes in network topology and routing, available access media, radio link conditions, user actions and preferences, context information, operator policies, quality of service (QoS) parameters, network composition [29], and security alerts.

Figure 19 illustrates six different trigger groups as boxes. The "offshoots" on the top point to example triggers belonging to each group. The rightmost group includes representative link layer "up/down" triggers (irrespective of the radio access technology). The leftmost group includes triggers originating from the application layer. In this example, certain triggers originate from the node ("System Resources") while others originate from the network ("Macro Mobility"). The "origin" corresponds to the entity that produces the trigger, for example, the radio access component. An advantage of the grouping approach is that it allows us detect relations between otherwise disparate triggers. This prevents the generation of excessive transient triggers based on, for example, the same root-cause event, such as a link outage, and reduces the number of events that need to be processed. For Eexample, in Figure 19, the link status trigger "link up", could be advantageous for a mobile node if it starts sending router/agent solicitation



Required QoS

Events related to a change of required QoS coming from the application sphere.

Available QoS

Events coming from a change of the current QoS. Basically, the network has got less or more resources and then adapt its QoS according to it.

Examples: less/ more bandwidth in the transport network

System Resources

Events related to the resources available in the system

Examples: (CPU load, low battery)

User Preferences

Events related to the user sphere (request/actions)

Examples: (static preferences, dynamic choice, ...)

Macro Mobilty

Macro-mobility triggers that indicate the terminal has just moved in the coverage of the new Radio Access Technology network (can be administrated by several different entities) or lost the L3 connectivity in current access network.

Link Status

Events related to the actual state of the link layer connection. The transition state will be useful for feeding the Ho decision function.

Examples: link up, down or break

FIGURE 19 An example of trigger groups

messages immediately after receiving the trigger, thus getting the router advertisements earlier than if it were passively waiting for them. This is especially the case for vertical handovers; see for example [43]. Conversely, the mobile node does not need to wait for router advertisements after a "link down" trigger.

Event sources need to be able to deliver notifications to interested applications and other system entities in a uniform and concise manner. This approach simplifies notification handling considerably, while guaranteeing sufficient diversity for event separation and classification. The trigger management framework (TRG) was developed to manage and efficiently propagate triggers originating from a variety of sources. TRG lays the foundation upon which sophisticated handover operations can be performed. The aim was to establish an extensible framework where new sources of triggers can be defined and included in a system implementation as necessary.

3.2 Trigger management framework

The main elements of the trigger management framework are detailed in [PV, 39] and include the entities which generate events (producers) and entities that use the trigger information (consumers). The trigger management framework is capable of collecting event information from various producers by using a specific collection interface, process them, convert them to a unified trigger format and distribute the triggers created to interested consumer entities. Producers as well as consumers can be any entities implementing the collection interface, and they can be located in the same or in a different node in the network. It should be noted also that the same entity can act both as a producer and a consumer; and also, a TRG entity may act as a consumer for another TRG entity.

The trigger management system is shown in Figure 20 and comprises (a) sources (or producers) feeding relatively fast-changing information about events; (b) trigger consumers, which receive notifications in the form of unified triggers about events they are interested in; and (c) the implementation of TRG, which includes data stores and internal logic. The TRG implementation processes the events received from the producers, generating triggers based on consumer provided (filtering) rules and making sure that all system wide policies are enforced.

A central part of the design is designating entities as producers and consumers of triggers. Consumers must state their need to receive triggers and can choose to stop receiving them anytime. The same entity may be simultaneously acting as a consumer and a producer. For example, it can receive all triggers originating from radio access events, but opt to receive only the upper layer triggers associated with security policy violations. In the former case, the consumer takes advantage of the trigger grouping and classification functionality; in the latter case, it additionally requests trigger filtering. Consumers can, of course, use these triggers to generate their own events and serve as a producer for other entities. It is expected that TRG will be used to guide handover decisions and executions via systems like VERHO, as described in 2.7.1. In particular, consumers can use triggers to find out whether the mobile device is moving within a single network or crossing different access technology boundaries and whether the addressing scheme, trust and provider domains should be changed accordingly.

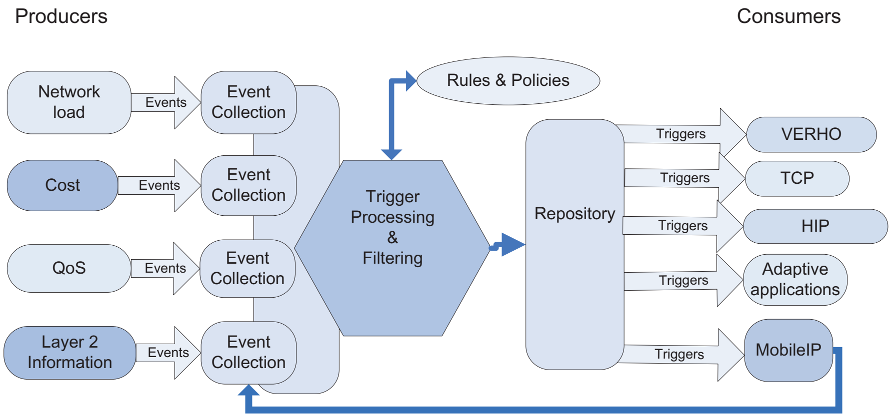


FIGURE 20 TRG-Architecture

The TRG architectural requirements address functional, performance, and security issues. As shown in Figure 20, the core TRG implementation is partitioned to three major components, namely, trigger event collection, processing, and storage. The figure also depicts examples of TRG event sources (QoS, Layer 2 Information, Cost, and Network load) and of TRG consumers (applications, Mobile IP, HIP and VERHO). As mentioned earlier, the same component may act as a trigger source and consumer at the same time, as in the case of MobileIP in the figure. In short, events are collected from the corresponding sources and are handed over to the trigger processing engine, which is responsible for timestamping and reformatting triggers (if necessary), and assigning them to the appropriate group. Consumer-specific rules for filtering and the filtering itself are handled during trigger processing. Processed triggers typically have an expiration time, after which they are automatically removed from the active triggers repository. TRG supports the application of different triggering policies (see Section 3.2.4), defined as a set of classification, filtering, trust, and authorization criteria/ rules. This allows our implementation to enforce a different policy at different times or when the node operates in different contexts.

The design and realization of the TRG functionality is flexible in the way that it supports dedicated mechanisms to handle the prioritization of events or ensuring the trust between different entities (consumers and producers). This helps to address one important issue, which several papers, see for example [35, 82, 11] focus on, the temporal event ordering and other time-related issues. Most importantly older event should not overrun newer or current valid event information.

3.2.1 Triggering Events Collection

Triggering events collection is a function in TRG, which receives events from various sources in the network system via the trigger collection interface. A TRG implementation may contain several event collectors, which may be distributed and be responsible for collecting different types of events. The need for different event collectors arises from the fact that the origin of an event source can be a hardware device, a system component implemented in kernel space, or an application implemented in user space. For example, each device driver could implement its own event collection functionality, which could be capable of handling triggering events produced by the specific device only. Moreover, sources can also be located in the network, at active network elements or at the user's home network, for example. Finally, a particular TRG implementation can act as a consumer of another TRG located in a different node. Thus, orchestrating the collaboration of several collection entities, in order to efficiently gather a larger amount of events.

Having dedicated collectors for different event sources enables the use of TRG in different operating systems as well. The collector can format the events to the format that TRG understands and there is no need to modify the core of the TRG functionality; instead the collector can be modified as necessary. This is also one of the key points in the architectural design of TRG that enables it to handle cross-layer information by having a collector at different layers as needed. For example, TRG can get connectivity information in FreeBSD through a collector that uses Route Socket and in Linux through a similar collector that uses RTnetlink socket. Obviously these collectors need to have their own implementation. The core of TRG could be implemented in kernel space for performance reasons, allowing for direct access to lower layer information. On the other hand, TRG can be implemented in application space, allowing for greater flexibility and easing the implementation and code evolution. The proof of concept validations of TRG described later in this dissertation follow the latter approach.

3.2.2 Trigger Processing

TRG handles triggers using a common format and reformats any "legacy" ones into a chosen format once they arrive via the triggering event collection.The trigger format is illustrated in the Table 1.

This common format for triggers allows the use of already existing event sources, which are not yet compatible with the TRG, and will be instrumental in migrating "legacy" systems to the new framework. New sources should implement the trigger event collection functionality and use the trigger collection interface in order to register their triggers and to make them available to consumers. Consumers can subscribe by specifying a set of triggers (and, optionally, filtering rules) and are expected to unsubscribe when they do not wish to receive them any longer. For each consumer subscription, the TRG processing compo nent ensures that filters are formatted correctly. It may also supply default filters

TABLE 1 Trigger format

Field	Type	Description
`id`	integer	Trigger identifier, same as producer identifier. Mapping producer name to identifier.
`type`	integer	Specific to the trigger identifier. Mapping producer information to type.
value	std:string	Specific to trigger type.
timestamp	time_t	Time when a trigger enters the TRG repository.

for certain consumers, and performs the actual filtering. Basic rules can also be used as building blocks or for crafting more sophisticated rules.

3.2.3 Trigger Repository

The repository is designed to meet the stringent requirements placed by the mobility management, but it can be used to store non-mobility triggers also. The basic primitives include adding, removing, updating, and disseminating triggers in a standardized format. Each stored trigger has an associated lifetime and is removed automatically once it expires.

3.2.4 Policies and rules in trigger management

TRG supports different triggering policies, defined as a set of classification, filtering, trust, and authorization criteria/rules. This allows TRG implementation to enforce a different policy at different times or when the node operates in different contexts. The availability, on the one hand, of a system-wide policy and, on the other, consumer-supplied filters lies at the centre of our TRG design.

System policies ensure that only designated consumers can receive certain groups or types of triggers. For example, a node may operate under different policies depending on whether the user is on a business or a leisure trip. Policies can also establish different trigger classifications and groupings in different contexts and are typically stored in a separate repository accessible to the TRG implementation. Filters allow the consumer to focus on a trigger subscription. For example, a monitoring application may be interested in receiving all network utilization measurements, while a VoIP application may be interested in receiving a trigger only when the utilization exceeds a certain threshold and the user is on the call. In fact, a VoIP application can even opt to be an intermittent trigger consumer, subscribing and unsubscribing to receive certain triggers solely when needed. TRG implementation uses access control policies to define:

- Which producers are allowed to register and send triggers to TRG. Producers are identified by the trigger IDs they register, and can be chosen on a system basis. For example, a policy allows only specific producers to register with TRG.

- Which consumers are allowed to subscribe to triggers. Policies can be very specific, specifying which consumers can receive certain triggers and from which producers.

Consumers are identified by their locator (typically a host address). For example it is possible to enforce a policy that dictates that triggers from a producer with ID=50 are allowed to be subscribed only from "localhost" entities. The Policy Manager applies access control using policies described in XACML (OASIS eXtensible Access Control Markup Language) [49].

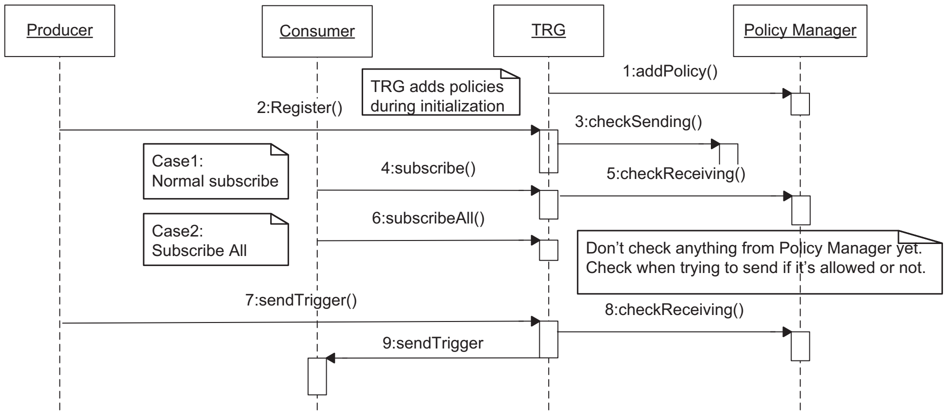


FIGURE 21 TRG-Policy Manager Message Sequence Diagram

Figure 21 shows which policy manager functions are called when a producer registers or a consumer subscribes. Typically the decision on whether to allow producer registrations and consumer subscriptions is made immediately based on the system policies, and the result is returned to the initiating entity. If a consumer attempts to subscribe to all triggers, the decision may be deferred until triggers become available. That is, the subscription for "all triggers" effectively becomes a subscription for "all triggers allowed" when system policies dictate so. In the prototype implementation, policies are described using access control lists read from a configuration file. Policies also define which consumers are allowed to subscribe and for which trigger.

3.3 Trigger management framework and 802.21 Media independent handover

The IEEE 802.21 Media Independent Handover (MIH) [26] working group is the standardizing body for an information service that will facilitate media independent handovers. The scope of the IEEE 802.21 standard is to develop a mechanism that provides link layer intelligence and other related network information to upper layers to optimize handovers (HO) between heterogeneous IEEE 802 systems and facilitates HOs between IEEE 802 and cellular systems. IEEE 802.21 helps with HO initiation, network selection and interface activation. The purpose is to enhance the experience of mobile device users. The standard supports HOs for both stationary and mobile users. For mobile users, HOs are usually needed when the wireless link conditions change. For stationary users, HOs are needed when the surrounding environment changes. Both the mobile node and the network may make decisions about connectivity. The HO may be conditioned by measurements and triggers supplied by the link layers on the mobile node. The IEEE 802.21 standard defines services that enhance HO between heterogeneous access links. Event service, Command service and Information service can be used to determine, manage and control the state of the underlying multiple interfaces. By using the services provided by the MIH function, users, such as Mobile IP, are able to better maintain service continuity, service adaptation, battery life conservation, networks discovery and link discovery. The MIH function also facilitates seamless handovers between heterogeneous networks.

Both TRG and the IEEE 802.21 MIH function seem quite similar, aiming to improve mobility management performance. But IEEE 802.21 stops short of allowing upper-layer entities to provide events that can drive a HO. It was impossible to compare the performance of the implementations since no MIH implementation was available when the tests with TRG were performed. The TRG approach emphasized standardized ways for consumers to receive a trigger from a variety of sources. Easy application registration to TRG permits them to get the information they want from different sources. Event generation, on the other hand, is by its very nature a distributed process and, without a central agent, all sources and consumers are forced to create a fully meshed topology. By introducing TRG, event collection becomes straightforward and trigger distribution standardized. Instead of using only the services provided by the 802.21 MIH functionality it is suggested that the system should use the TRG framework together with 802.21 services. 802.21 can be, for example, the source entity that provides the lower layer information for TRG framework since TRG can be used to collect and distribute the upper-layer information.

While IEEE 802.21 is designed only for mobility management, TRG can be used for other purposes as well. The design of TRG is not limited to guide only mobility management and handover decisions. For example, TRG is used as an important building block for the Ambient networks Information Service Infrastructure (ANISI) [18]. ANISI is an information service infrastructure designed to provide services and applications at different layers of the protocol stack with support for network information gathering, correlation and intelligent decision-making in support of enhanced mobility management and context-aware communications. There, as part of ANISI, TRG allows the system components to manage context and triggers comprehensively. TRG and IEEE 802.21 can also provide cross-layer information to help adapting QoS-sensitive applications and extend the traditional multimedia signaling protocols as presented in [59, 72, 5].

It is also studied how TRG can provide hints about moving the communication endpoint from one device to another, as explained in Section 3.4.1

3.4 Validation and experimental evaluation

The validation and performance analysis of TRG was carried out with a real TRG implementation with several different use case scenarios. TRG architecture design, with the possibility to use separate event collectors in different environments, makes a TRG implementation portable, and it has been integrated in several prototypes. Also the user-space C++ implementation of TRG has been tested with several different operating systems: FreeBSD release 6.1, Linux Fedora Core with kernel 2.6.12, Ubuntu Linux 8.04.3 LTS and 10.04 LTS with kernels 2.6.24-5mip6 and 2.6.32.15+drm33.5, Windows XP, Mac OS X with kernel 10.7.0 and Linux familiar v.0.8.4 with kernel 2.4.19. The user space TRG implementation can be compiled to support also other operating systems, such as Maemo, for example.

TRG has been tested with different mobility protocols with the focus on using the Mobile IPv4 [57], Mobile IPv6 [28] and Host Identity (HIP) protocols [45]. TRG is also a part of the Ambient Networks Architecture [47] and a prototype as discussed in [50, 67, 53]. TRG and Mobile IP integration with the use of network information presented in [34, PVIII] showed the benefits of using TRG for the Mobile IP when network is congested. HIP integration with TRG and test evaluations presented in [51] showed as well that TRG processing forms only a small factor (less than 9 percent) of the total processing time, namely, trigger collection, processing and dissemination.

TRG has been validated with the Ambient Networks mobility management with moving network support and also with a session mobility scenario [PIV, PV] as well as a part of the Ambient Networks gateway selection [36]. TRG is a main component of the distributed control and management framework as described in Section 3.5. In addition to TRG with mobility management, it has been investigated how TRG can help scalable video adaptations [59]. And last but not least the performance of the TRG implementation was tested with dedicated scenarios [PVII].

The work contributing towards this dissertation focused on the mobility management, and the following sections summarize the TRG validations and tests related to this area.

3.4.1 Session mobility with trigger management framework

Instead of limiting the validations to traditional mobility management protocols it was decided to implement a new novel mobility scenario where TRG was employed to enable streaming video session handovers between two mobile devices.

In the scenario of Figure 22, the user starts watching a video streamed to his

FIGURE 22 The video is streamed to the laptop (left), followed by a session handover to the PDA (center), and a handover back to the laptop (right)

laptop. His PDA is nearby, and the user decides to move to another room but would like to keep watching the video on the way. The PDA is fitted with a SoapBox multi-sensor device (detailed in [74]), which produces sensor messages for TRG. A sensor producer reads raw sensor data from SoapBox and refines them to sensor messages like "acceleration X sensor = 1.62". Sensor data is collected and forms higher level context information messages like "device orientation = downwards". These are sent to the TRG via a trigger collection interface. The decision to use SoapBox as a TRG producer was based on the fact that sensors are becoming more common in everyday life and new smartphones come with several sensors that can measure, for example, illumination. These sensors are well suited to act as trigger sources. When the user picks up the PDA, a "vertical orientation" trigger initiates a session handover from the laptop to the PDA; the two have to coordinate and make arrangements for the transfer of a video streaming session (Figure 22). A successful session handover allows the user to receive the streaming video on the PDA over the IEEE 802.11 seamlessly. The user can also explicitly initiate a session handover by pressing a PDA button. In this example, TRG handles triggers associated with the mobility, orientation, and user preferences, keeping the video flowing smoothly while changing the communication end-point.

Figure 23 illustrates the logical topologies of two different test cases, which were evaluated in a laboratory environment, based on the scenario shown in Figure 22. On the right-hand side of Figure 23, all devices are connected using IEEE 802.11 in an ad-hoc mode, as if the user were streaming the video from his digital collection at home. On the left hand side, the server is located in a different network, as would be the case when watching a video from a service provider over the Internet. In both setups, a ten minute video encoded at 576 kb/s was streamed over UDP. At t = 3 min, a session HO from the PDA to the laptop is triggered, and at t = 7 min, the session is "moved" back to the laptop. The quality of the video stream was qualitatively examined, and any losses of frames were noticed. It is important to note that the focus of these experiments is not on showcasing the session handover, the above being simply a particular application of triggers leading to a HO. These experiments aim at assessing the feasibility of introducing a TRG implementation in small handheld devices. The purpose is not attempt to formally draw peak signal to noise ratio (PSNR) curves but point

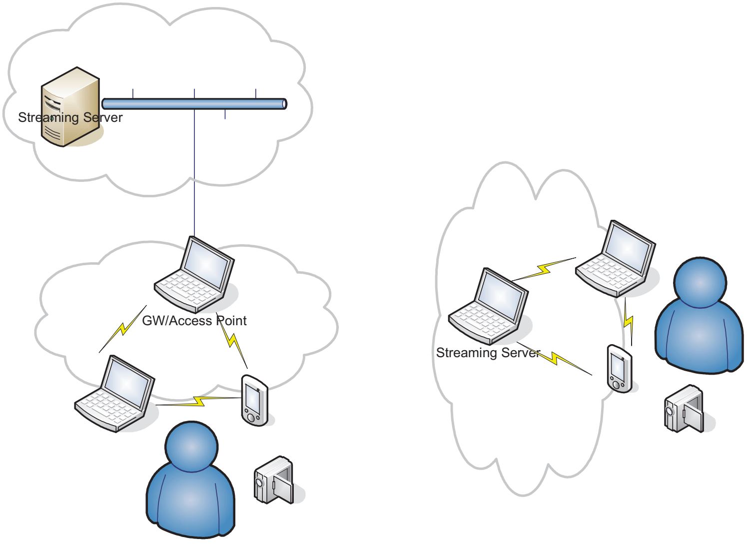


FIGURE 23 Topology of the session handover testcases

out that the absence of packet loss in our evaluation scenarios is a good indication about the ability of the current TRG implementation to expeditiously deliver triggers and enable video session handover.

Traffic traces were captured by using tcpdump during the experiments. All packet IDs sent by the video server were compared with the packet IDs received by the Laptop. The PDA video clients showed that no packet losses occurred during the session handover. The effect of TRG signaling and the actual session handover on packet delay was negligible, compared to the packet delays before and after the session handover. Figure 24 illustrates the packet interarrival time as recorded by tcpdump at (a) the streaming server, (b) the receiving laptop and (c) the receiving PDA. The left-hand side shows the packet interarrival time measured at the streaming server, laptop and the PDA during the delivery of the 10-min video stream. The solid, dumpy line is caused mainly by the measurement of the server and the most important information is above or below that line. In Figure 25 we zoom in at around t=3min when the first session HO is triggered from the laptop to the PDA. As Figures 24 and 25 illustrate, only few packets have >0.1 s interarrival time. These results are very promising, taking into account that this is a prototype implementation and because the PDA was running the video client and captured packets using tcpdump throughout the experiment, leaving few spare system resources available.

This work has focused on the empirical validation and evaluation of TRG. The theoretical aspects (scalability, security, reliability) have been partly addressed elsewhere [58] and any further analysis is also part of a future work agenda. It is important to note that these sets of experiments go beyond showcasing the concept of TRG-assisted session HOs. This is simply a particular application of triggers leading to a handover. Instead, they aim at assessing the feasibility of

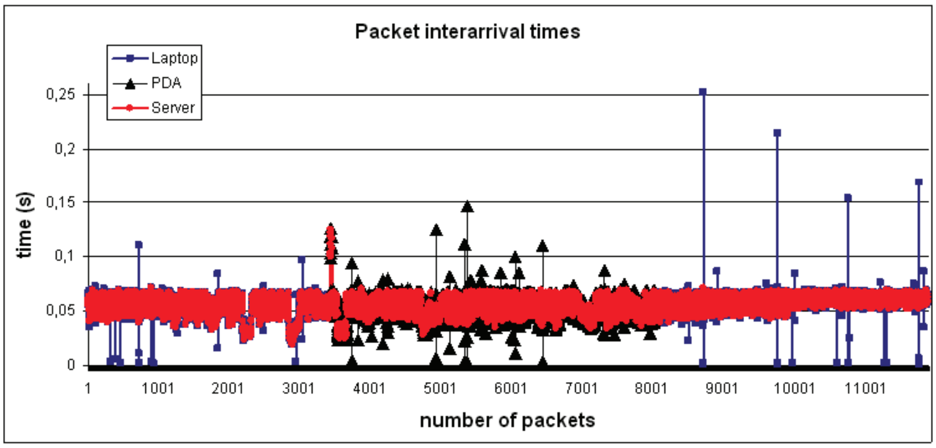


FIGURE 24 Experimental results when triggering a session HO

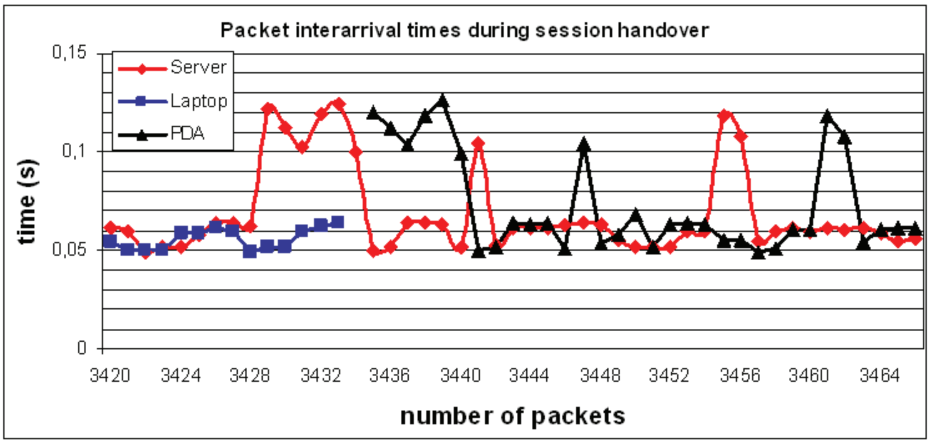


FIGURE 25 Zoom in to experimental results when triggering a session HO

introducing a TRG implementation in small commercial, off-the-shelf handheld devices.

3.4.2 Performance analysis of trigger management framework

The development and evaluation of TRG continued with the new test cases where an updated implementation of TRG, enhanced with web service interfaces, was used. In these tests 100 000 triggers were submitted from several sources to TRG, and TRG delivered them to different consumers. There were two different test cases, aiming to of quantify TRG performance under stress (and perhaps clearly "unrealistic") conditions. Test case 1 employed n producers connected with m consumers via TRG. During the test, each producer sent 100 000 back-to-back triggers and all the triggers were distributed to all m consumers. This means that TRG needed to process $n \times 10^5$ triggers and deliver $n \times m \times 10^5$ triggers. Figure 26 illustrates an example case where each of n=3 producers A, B and C sent 100 000 triggers (trigger IDs 51, 52 and 53, respectively), to m=4 consumers (I, II, III, IV). That is, in this particular scenario, each of the four consumers received 300 000 triggers from TRG.

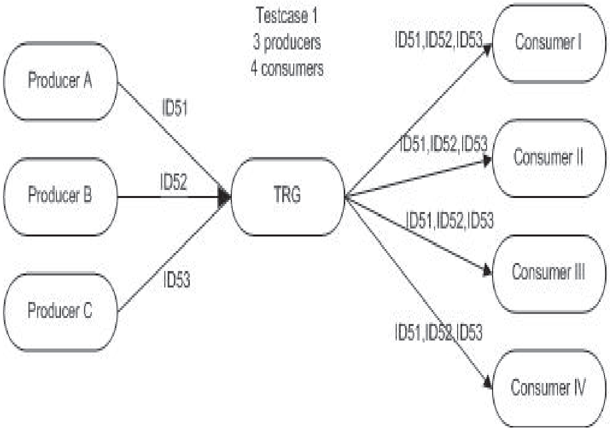


FIGURE 26 Triggers in Test Case 1

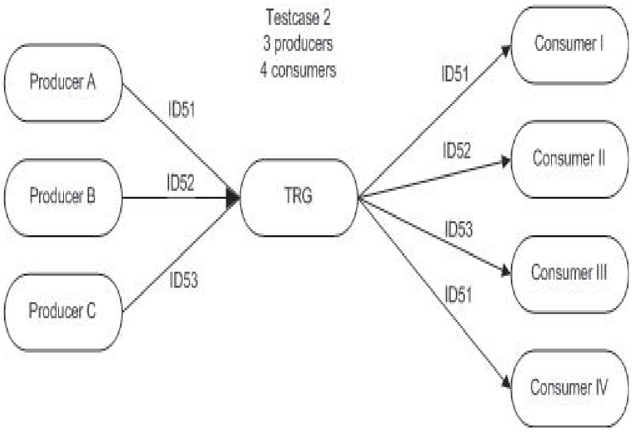


FIGURE 27 Triggers in Test Case 2

Since there are several possible scenarios about how triggers are distributed between producers and consumers, also a test case 2 setup, where each consumer has only one dedicated producer, was defined (Figure 27), In this setup TRG needs to process $n \times 10^5$ triggers and deliver $m \times 10^5$ triggers. If there are more producers than consumers, triggers will be distributed evenly between the available consumers. As mentioned, tests were carried out using the C++ implementation of TRG with web service interfaces for producers and consumers. For the performance tests a laptop with an Intel Pentium M 1.70 GHz PC with 1 GB RAM, running FreeBSD release 6.1 was used.

Table 2 shows the number of delivered triggers with average processing times in milliseconds for each trigger received by TRG from the producers in test case 1. In this case, only the number of consumers has a major effect on the processing time of each trigger. This indicates that TRG can cope with several registered producers even when there is no subscribed consumer for certain producers. Moreover, the average trigger processing time is only few ms/subscribed consumer in this prototype implementation stress test.

TABLE 2 Total number of delivered triggers and average processing time (in ms) per trigger in test case 1

	Producers				
Consumers	**1**	**2**	**3**	**4**	**5**
1	100k, 1.7 ms	200k, 1.7 ms	300k ,1.8 ms	400k, 1.7 ms	500k, 1.8 ms
2	200k, 2.3 ms	400k, 2.5 ms	600k, 2.4 ms	800k, 2.5 ms	1000k, 2.4 ms
3	300k, 3.2 ms	600k, 3.2 ms	900k, 3.2 ms	1200k, 3.1 ms	1500k, 3.3 ms
4	400k, 3.7 ms	800k, 3.8 ms	1200k, 3.8 ms	1600k, 3.8 ms	2000k, 3.8 ms
5	500k, 4.5 ms	1000k,4.6 ms	1500k, 4.7 ms	2000k, 4.7 ms	2500k, 4.5 ms

Figures 28 and 29 show the total processing time of Test Case 1, with and without the TRG filtering mechanism. It can be seen that when the number of consumers and producers increase the total processing time will also increase in a linear fashion. This is expected since the number of processed triggers increases when more consumers and producers are added. The costs of adding consumers and producers are both linear, but the cost of adding consumers is greater than the cost of adding producers.

To give an example, when making a simple comparison with the calculated slope $k = \Delta y / \Delta x$ of the curves of total processing time, with and without filtering, it shows that the processing time increases faster when there are more consumers (the slope of the curve with 1 consumer k = 177,6 and with 5 consumers k = 454,9 in the Test Case 1 without filtering). This behaviour can be explained as a cost of the duplication of triggers because the number of triggers that have to be duplicated and delivered to the consumers increases when more consumers are added. However, this does not increase the average processing time of one trigger. The number of producers also has an effect on the total processing time but not as marked as that due to the number of consumers.

The cost of using the filtering function of TRG was evaluated first with the Test Case 1. Now each of the 5 producers who registered for sending triggers

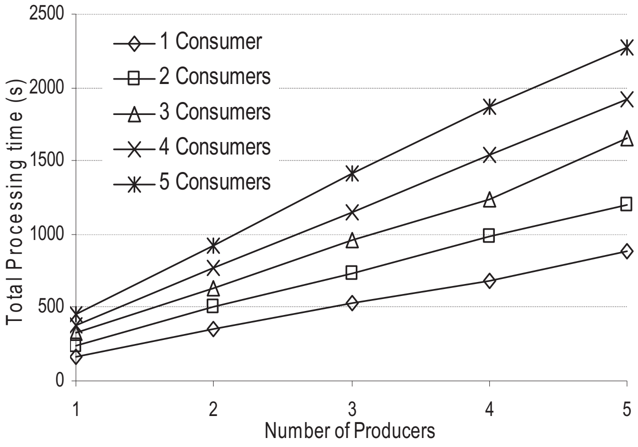


FIGURE 28 Processing time in Test Case 1 without filter processing

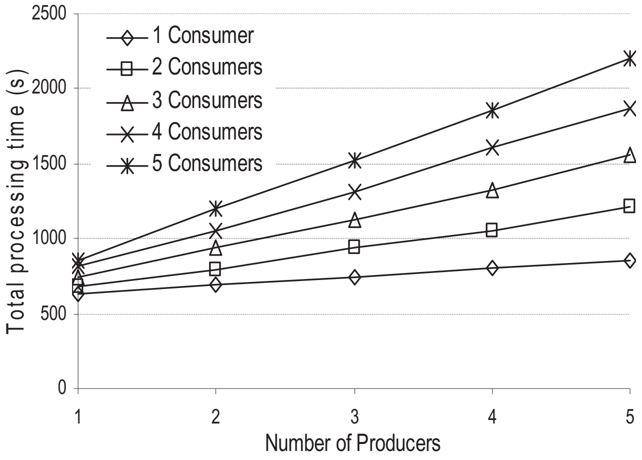


FIGURE 29 Processing time in Test Case 1 with filter processing

sent 100 000 triggers. Thus TRG received a total of 500 000 triggers during the test. Figure 29 shows the total processing times when the filtering mechanism was used. When there is 1 producer, the other 4 producers are filtered away, and the triggers from the sole producer are duplicated and delivered to all four consumers. In the case with 2 producers, the rest 3 are filtered away, and so on. The results show that it takes more time to process all triggers, but this is not caused by the filtering mechanism itself. When comparing the total processing times, as the triggers from 1 producer are delivered to consumers (Figure 29), the total processing time is increased, but this is because now there are 5 times more triggers received by the TRG than was the case without the filtering mechanism, since all 5 producers are sending 100 000 triggers all the time during the test. When the filtering mechanism is not used, the number of producers is controlled by making a new registration for each producer.

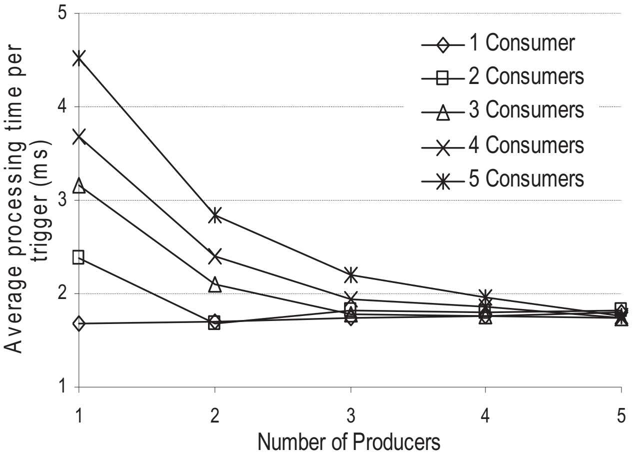


FIGURE 30 Average processing time per trigger in Test Case 2 without filter processing

To further quantify system behaviour when filtering is employed, we consider Test Case 2. When evaluating the filtering function in Test Case 2, each consumer had a filtering rule that was true for all triggers, allowing the distribution of all triggers to the subscribed consumers. By having this "receive all triggers" rule it was possible to test the effect of the filtering mechanism: every time when a trigger is produced, TRG needs to run the filtering mechanism before passing the trigger to the consumers, even though none of the triggers are filtered away. The purpose was to test the effect and cost of running the filtering function. The filtering mechanism itself does not have a major effect on the processing time when compared to the effect of increasing the number of consumers.

When comparing the processing times in Test Cases 1 (Table 2) and Test Case 2 (Figures 30 and 31), we see that the duplication of each trigger to every consumer, which is needed in Test Case 1, increases processing times. In Test Case 2, when the numbers of producers and consumers are equal the differences of the processing times can be measured in microseconds, since now there is no need to duplicate any triggers. The differences in the test cases are shown in Figures 26 and 27.

The test and evaluation cases show that it is, in fact, the duplication of triggers and number of messages that have the major effect on the processing time

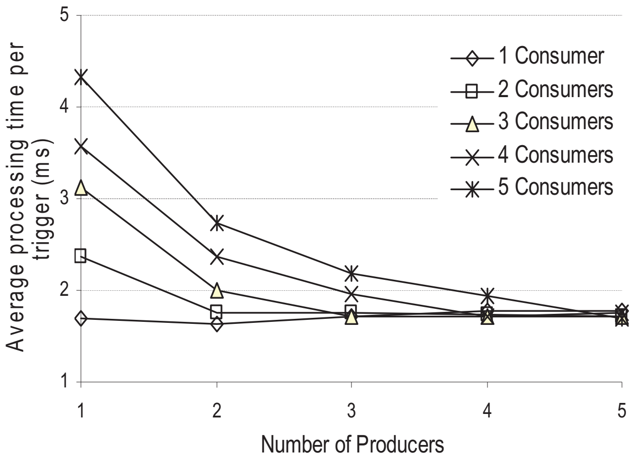


FIGURE 31 Average processing time per trigger in Test Case 2 with filter processing

of triggers. It can be seen in Figures 30 and 31, for n=5 producers and m = 5 consumers, that the processing time does not depend on the number of consumers. In this case there is no duplication of triggers since each consumer has only one dedicated producer the events of which are not distributed to other consumers.

The feasibility of using the TRG framework to process, filter and disseminate a huge amount of triggering events was shown in practice. Each case showed that the processing time of one trigger does not increase, even when processing a huge number of triggers. Even though the test cases are clearly unrealistic with the huge number of triggers employed, they prove that using TRG doesn't cause any major delays to the handover time. On the contrary, TRG enables the handover mechanism to react more rapidly to more events. It is also important that filtering mechanisms do not have any major effect on the processing time since this allows handover decision mechanisms to react faster to important events. It was shown that the cost of having more consumers and producers increases the processing times linearly, and the cost of using the filtering function has only a marginal effect on the processing times. Of course, more total processing time is needed for processing and disseminating all triggers in the test cases. However, implementation of grouping and classification of triggers [PV] and a mechanism, e.g. in the TRG source for prioritizing the trigger traffic, allowing the critical triggers to be processed and distributed faster, TRG is ready to process the triggering events.

3.5 Distributed mobility management in heterogeneous networks

As this dissertation focuses on the IP based mobility management especially on the heterogeneous network environment, the author and his colleagues built a heterogenous networks testbed in VTT's Converging Network Laboratory [8] with WiMAX 802.16d, WLAN 802.11g and 3G/HSDPA technologies. In the test there were two different test setups; one described in the Section 3.5.2 and another in [24]. With the tests, it was possible to evaluate also the mobility of an entire network by introducing a mobile router for a mobile network. The behaviour of the mobile router with the Mobile IP protocol is similar to that in the single mobile node case. One difference is that the mobile router acts as a gateway router for other nodes in the mobile network.

As a part of the evaluations in a real environment, a new idea and concept about distributed mobility management was created. The idea is based on the fact that, for mobility management, there may be many different nodes involved and those nodes should work in a cooperative manner. For example, in the network side there are nodes that know best their own status, e.g. the level of congestion in the access point or routers. In the similar way, the mobile node obviously knows best its own status and preferences. For enabling cooperation and information distribution, a Distributed Control and Management Framework (DCMF) [PVIII, 38] was defined. The work related to DCMF is a part of a still ongoing research, but the following section will give a brief introduction to DCMF.

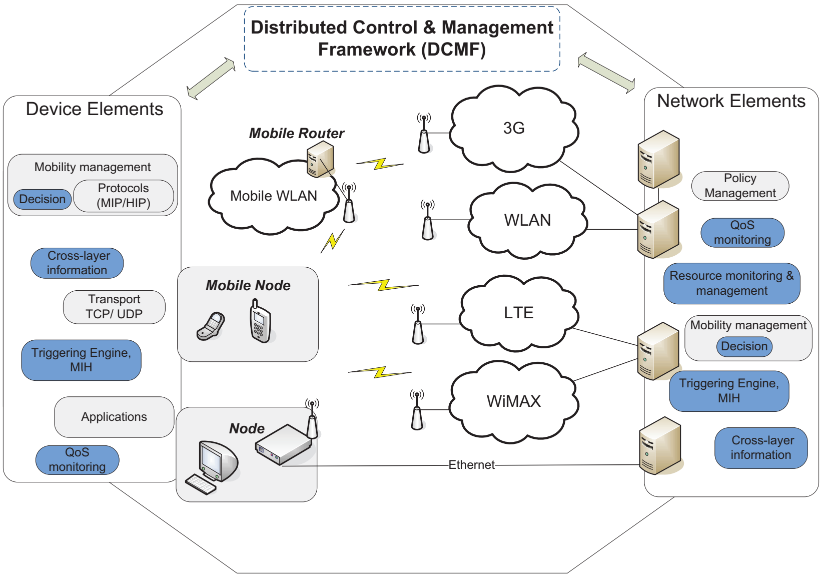


FIGURE 32 The Distributed Control and Management Framework (DCMF)

3.5.1 Distributed Control and Management Framework (DCMF)

Figure 32 illustrates the Distributed Control and Management Framework (DCMF). The illustration also shows examples of the software modules and functionalities that are typically located at mobile terminals (left-hand side of Figure 32) and in core network nodes (right-hand side of Figure 32). DCMF is designed to support distributed information exchange between operators, enable advanced access control, and allow for different optimization techniques such as balancing traffic load between access points. The basis of this framework is the TRG functionality (see Section 3.2) that provides a unified signaling architecture between network nodes, capable of serving different entities, located either in the network or in the terminals and controlled by different players (terminals by the user, network by the service provider). The network side elements (right-hand side of Figure 32) that will benefit from such a signaling framework include, for example, network/operator aided mobility management and resource management. On the terminal side (left-hand side of Figure 32), mechanisms such as mobility management or transport protocol optimization solutions need access to extensive amount of information related to network access characteristics and roaming.

The information that is made available through DCMF by employing TRG for signaling between nodes can be used for optimizing network and terminal side operations in terms of application QoS requirements, energy efficiency, security, and network load balancing. In order to obtain relevant information for the decision-making process, we need to rely on, for example, operator policies, QoS measurements, and various cross-layer events (e.g. those defined and grouped in Section 3.1.

The components of the DCMF may vary depending on the architecture and the needed setup. One possible setup and architectural view of the components is shown in Figure 33. In this setup the main components are TRG, Network Expert System (NES), Mobile Expert System (MES), and network resource probes.

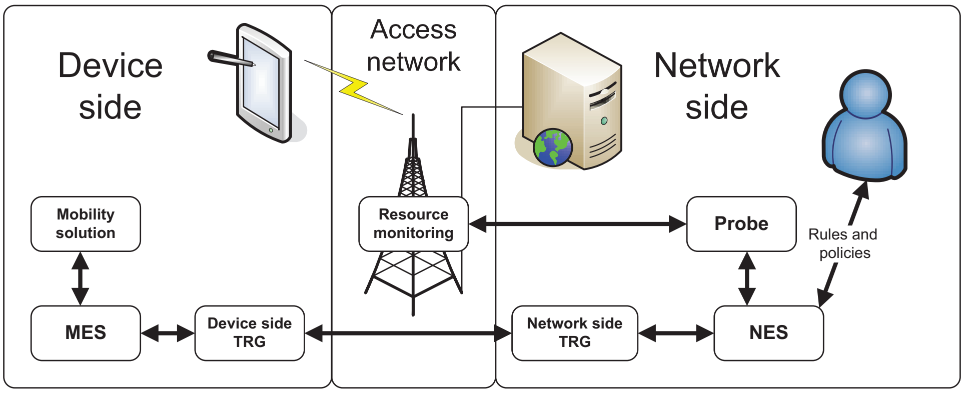


FIGURE 33 Main components of Distributed Control and Management Framework

The role of TRG in the DCMF is to act as an interface between the information sources and the information consumers. TRG provides an information dissemination mechanism between the core network and mobile devices through the wireless access network. Information dissemination can use the cascaded functionality of TRG. In the cascaded mode, one TRG network entity collects information from the network and acts as an information source to another TRG entity.

Other network side components are the Network Expert System (NES) and network resource probe(s). Network resource probes are the components that are collecting available information from the network side. A network probe can, for example, collect valuable information from WLAN access points using the Simple Network Management Protocol (SNMP [33]). This information is further processed by NES, which is a Self-organized Map (SOM) [78] based decision-making component. NES is capable of analyzing the level of congestion in an access network. Human experts can guide the decisions of NES by using different training data and by defining different sets of rules and policies to guide the decision-making (for more information about the functionality of NES, the reader is referred to [24]). This processed information is further collected by the network-side TRG which provides the information (in the form of triggers) to the TRG residing on the mobile node. Communication with TRG uses reliable Web services or plain TCP sockets depending on the source or the consumer. The communication between TRGs is always Web service based. Figure 34 shows the messages in a simple example case where congestion information is gathered from a WLAN access point to the mobile terminal.

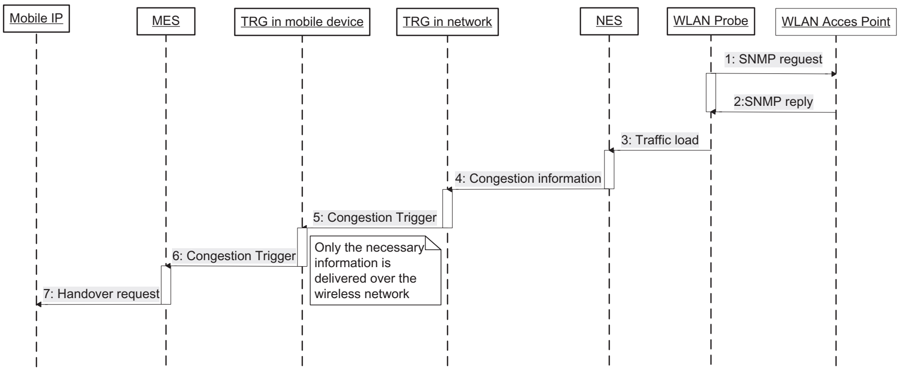


FIGURE 34 Message flow between the components

By using this approach, it is possible to minimize the access network traffic as well since only the necessary information is transferred to the mobile node over the wireless network (message 5 in Figure 34). The information can be further analyzed by the Mobile Expert System (MES) on the mobile node. The MES in the mobile node can be a similar entity as the VERHO system, described in Section 2.7.1 The main idea for this approach is that each node and network knows best its own conditions, characteristics, and resources and can help other entities to make their own decisions as well.

3.5.2 Validation in heterogenous networks testbed

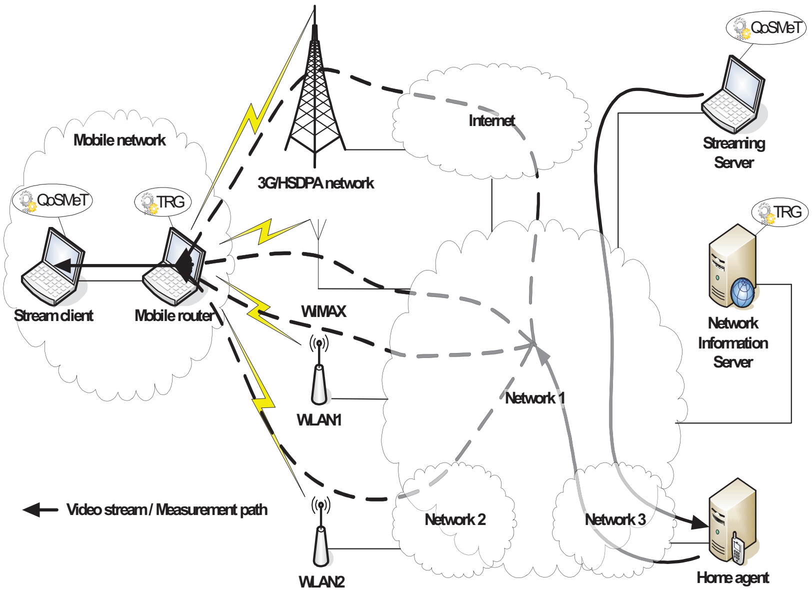


FIGURE 35 Heterogeneous networks testbed and the DCMF evaluation scenario

Figure 35 illustrates the testbed and a scenario for evaluating and validating the DCMF operation and mobility management. With this testbed it was possible to measure and demonstrate the performance benefits achieved in a mobile multiaccess environment. The testbed, described next (see Figure 35), includes a mobile router (MR) providing access to a mobile client. In the beginning of the scenario, a MR is connected to a WiMAX network. In the next step the MR switches its connection to WLAN1. While connected to WLAN1, the MR moves away from the WLAN1 access point, and, as the WLAN1 signal weakens below a threshold, the MR makes a handover to WLAN2. While in WLAN2, the MR moves out of the coverage of both WIMAX and WLAN2, and, as WLAN1 gets suddenly congested, the MR makes a handover to a 3G network. During the scenario, the client connected to the MR continuously streams a video from an Internet service.

The testbed includes a 3G/UMTS cell, two WLAN access points, and one WiMAX cell (see Figure 35). Using this testbed we show how DCMF can help mobility management and can significantly improve the end-user experience. The configuration consists of three network servers, four access networks to which the mobile router is connected to, and a client connected to the mobile router (MR). For mobility management, the MR uses Mobile IPv6 (UMIP implementation) [76] enhanced with DCMF and acts as a router to the mobile network. DCMF provides the triggers detailed in Table 3 to the Mobile IP (MIP) client running on the MR. These triggers are provided locally on every network interface addition or removal, interface IP connectivity change, and routing table change. DCMF provides also triggers concerning WLAN access point operational or congestion status changes and radio interface quality changes.

The decision logic added to the Mobile IP client uses the triggers listed in Table 3 to evaluate all available networks and steer the MR to choose the best possible network.

TABLE 3 Example triggers in the testcase

Name	Description
INTERFACE_ADD	Network interface added to the local machine
INTERFACE_REM	Network interface removed from the local machine
INTERFACE_CONN	Local network interface gained IP connectivity
INTERFACE_DISC	Local network interface lost IP connectivity
ROUTE_ADD	Route added to the local routing table
ROUTE_DEL	Route removed from the local routing table
WLAN_OPER_REM	WLAN access point operational status change
WLAN_CONG_REM	WLAN access point congestion status change
WLAN_QUAL_REM	WLAN radio interface quality status change

In the tests the QoSMeT measurement tool was used [64], which is running on the streaming server, and the client, were used. QoSMeT can measure one-way QoS characteristics experienced by a video stream from the server to the client. QoSMeT is a QoS measurement tool which was developed by VTT and which uses GPS synchronization to accurately measure one-way QoS characteristics such as packet delay or jitter of an end-to-end link. The video used in the experiment is streamed over UDP (with an MTU of 1358 bytes) at an average bit rate of 485 kb/s.

3.5.3 Results

Figure 36 illustrates three aspects related to the experiment. First, the delay experienced by the video stream in the testbed with the presence of DCMF; second the network which is actively used for streaming the video to the client; and third the point at which a vertical handover occurs. The MR is initially connected to the WiMAX network (A). The WiMAX network is then disconnected and the MR makes a handover to the WLAN1 network (point B in Figure 36). When the WLAN1 network gets congested (by adding UDP traffic with a bit rate of 60 Mb/s to another node in WLAN1) the MR makes a handover to WLAN2 (point C). Finally, the WLAN2 network's transmission power is lowered gradually until the MR makes a handover to the 3G network (point D).

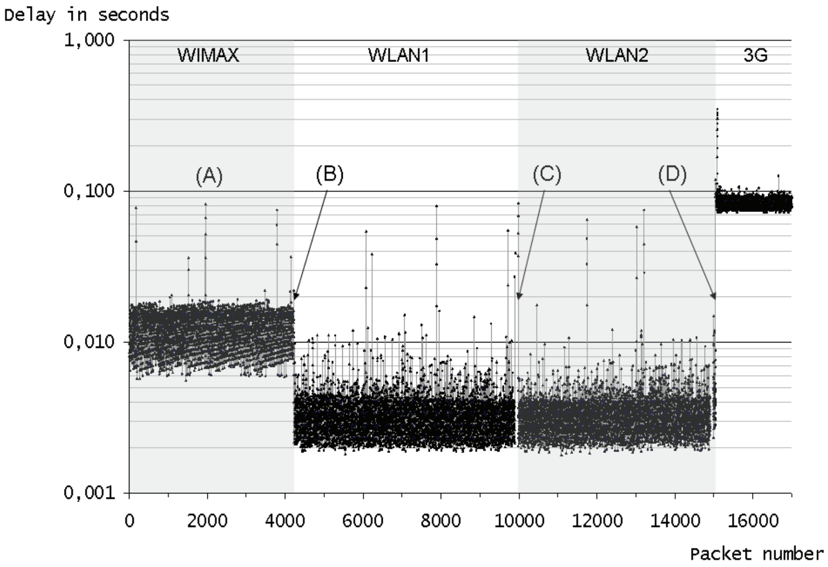


FIGURE 36 One-way packet delay and handovers in our testbed evaluation

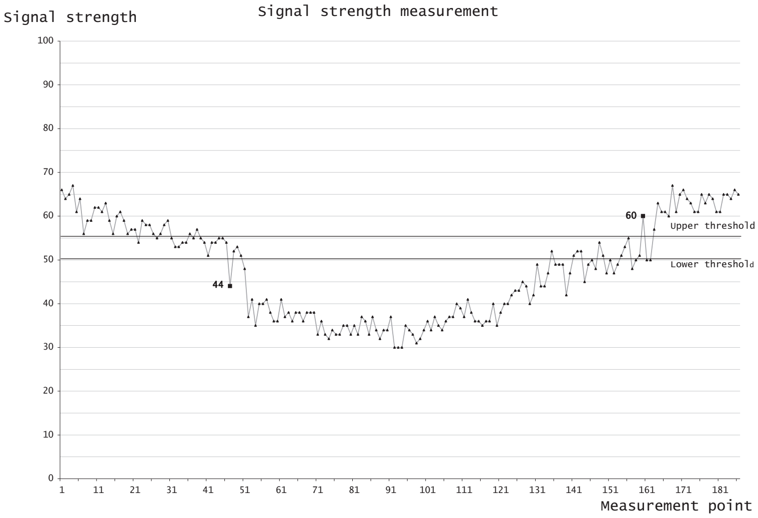


FIGURE 37 Signal strength measurement

Although a network card is able to maintain a connection with an access point, packets can still be lost due to low signal strength. To address this, DCMF provides triggers based on the wireless network adapter's signal strength. Signal strenght with some threshold and hysteresis values is a very commonly used trigger for handovers. The threshold and hysteresis help to avoid unnecessary handovers [60, 52]. Figure 37 illustrates a signal strength measurement by an information source in the DCMF. When the signal strength drops below the lower

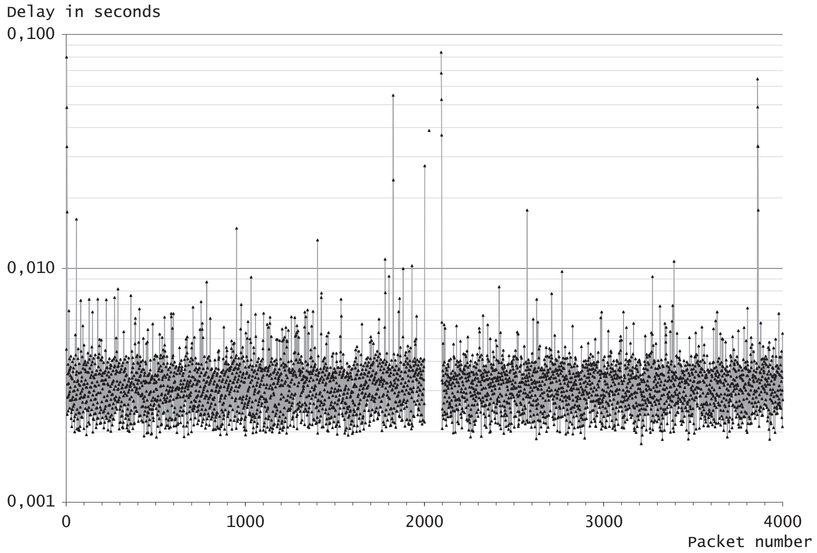


FIGURE 38 Network congestion in WLAN1 in case of DCMF assistance

threshold set in the information source, the source sends a trigger informing interested parties that the interface in question is considered to be down due to low signal strength. Again when the signal strength rises above the upper threshold a trigger indicating that the interface is up is sent. The interface down trigger based on the signal strength triggers the handover (D) in Figure 36.

The MR by itself can only assess the quality of a WLAN connection mainly based on the received signal strength. DCMF can provide additional information from the network to the MR via the TRG. The handover (C) in Figure 36 is a result of such information. At that point in our scenario WLAN1 is heavily congested with additional traffic from another client in the network. The WLAN probe in the network information server (NIS) notices this and sends a trigger to the TRG running in the NIS which forwards the trigger to the TRG in the MR which in turn sends it to the decision logic guiding the MIP client. The decision logic instructs the MIP client to switch to the WLAN2.

To better illustrate the advantage of DCMF in mobility management we can focus on the handover (C) in Figure 36. The packet delay in this situation is graphed in Figure 38 and as a reference without the help of DCMF in the same situation in Figure 39. In Figure 38 with the help of DCMF the packet delay remains constant and only 90 packets are lost while the MR makes a handover to the WLAN2. This packet loss is caused by the delay between the start of the congestion and the initiation of the handover. Visually, a casual viewer experiences a small glitch in the video just before the handover is made. Figure 39 shows the loss of 1703 packets and no handover at all is made by the MR during this time. Eventually, after 6238 lost packets, the MR makes a handover which is caused by the MR not being able to update its binding with its HA because of the con-

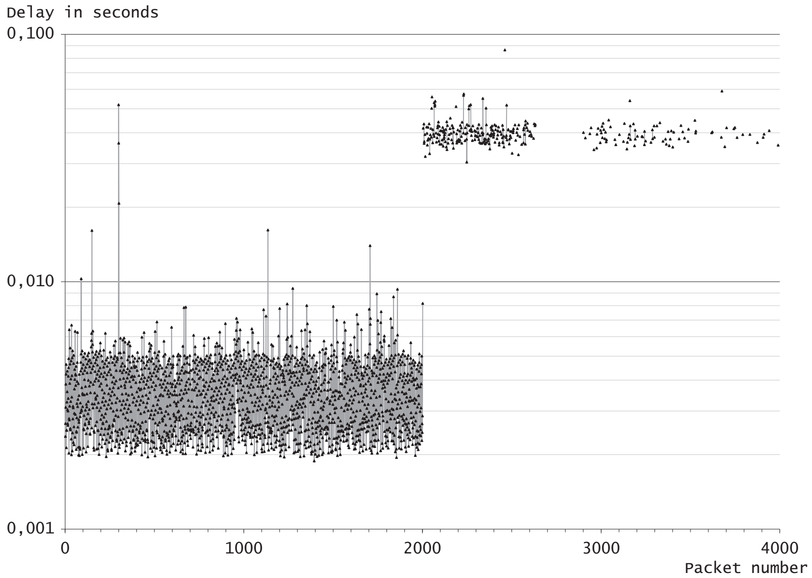


FIGURE 39 Network congestion in WLAN1 without the assistance from DCMF

gestion. Visually this corresponds to 50 seconds of unintelligible video. During this time only occasional broken frames of the video are displayed to the person viewing the video.

The Table 4 shows a classification of the packets in Figures 38 and 39 based on the delay. The table also shows the number of lost packets. As can be seen from the table, DCMF enhanced system offers better QoS in terms of packet loss, the packet loss being 2.2% with DCMF compared to 42.6% without DCMF. Also during the congestion, the delivered packets without DCMF have significantly longer delay, which can be seen as the number of packets with delay over 0,01 sec.

TABLE 4 Packet delay distribution (sec)

	Delay $<$ 0,004	**0,004 $<$ Delay $<$ 0,01**	**Delay $>$ 0,01**	**Packets lost**
DCMF	3369 (84.2%)	518 (12.9%)	24 (0.6%)	90 (2.2%)
No DCMF	1303 (32.6%)	690 (17.2%)	305 (7.6%)	1703 (42.6%)

The measurements, during which DCMF was not used, were made using the default settings of the MIP client. In this case the most relevant setting was the interval at which the MR renews its binding with the Mobile IP Home Agent (HA). The default of this setting on the MIP client used is 60 seconds. This means that the MR's network connection will be down 30 seconds on an average in the case of a total connection loss or severe network congestion as in the example. This problem can be diminished by lowering the binding renewal interval but this also puts unnecessary strain on the network as the communication between the MR and the HA increases. Still, with less severe congestion it is possible for the MR to experience lowered QoS and packet losses for indefinitely long periods of time as long as the MIP client is able to renew its binding with HA.

Figure 40 shows a concrete example of an end user experience about a video stream will in a congested network. This shows quite nicely that also mobility management may have a significant role in the end user experience.

FIGURE 40 User experience in handover case

The test case evaluations, as seen from the evaluations and figures, showed that there is a clear benefit of using DCMF to assist mobility management protocols. This evaluation case and the implementations used are also steps towards the media-independent seamless handovers in the mobile wireless world.

3.6 Summary

This chapter focused on the mobility related events and presented a novel TRG framework for managing mobility related triggers. It described the functionalities for collecting information from various event sources originating not only from the lower layers of the protocol stack (physical, data link, and network), but also from the upper layers and processing the collected events in a standardized trigger format. By using the defined mechanism, the TRG framework enables easy and efficient use of cross-layer and cross-domain information. This framework was implemented and evaluated by performing tests in a real environment with several operating systems (Linux, FreeBSD, Windows, Linux Familiar for the PDA) to prove its robustness and to measure its performance.

Experiments with the TRG framework, including the performance tests and evaluations, showed that the implemented TRG functionalities are very promising. TRG can be run in a small device with a very limited processing power, and it can enable lossless session handovers between devices. Stress tests showed that the TRG filtering mechanism does not cause delay for processing time and that TRG can be used to filter and disseminate a huge number of triggers from several information sources.

It was also discussed how the mobility management can work in a distributed manner. The new Distributed Control and Management Framework (DCMF) enables an efficient way of exchange and use of cross-layer information in heterogeneous multiaccess environments. DCMF enables information flows between network and mobile entities. Validation tests showed how distributed mobility management can enable cooperation and information distribution between different nodes. They also showed that there are performance benefits to be gained from using a network-aided information service to assist in handovers. Validation test results further showed that a real benefit can be gained from enabling efficient and cooperative use of information as a part of mobility management and that there is also a need for it.

4 CONCLUSIONS AND FUTURE WORK

This dissertation deals with research challenges and problems caused by the heterogeneity of the networks. Its main focus is the functionality of the current mobility management and protocols in the access networks. This dissertation also presents new approaches, which will help mobility management functions to achieve seamless mobility for the users. The functionality and the problems of the different mobility management protocols and their behavior, especially in the heterogeneous environment, is discussed. There is a special focus on solving the problems caused by mobility protocol handovers, handover decisions and collection and distribution of mobility related information. The investigation also sheds light on how the mobility management can work as a distributed solution in current and future network architectures.

The study of the mobility protocols consists of the evaluation of several mobility protocols, their functionalities and drawbacks. Even though there are several protocols suitable for vertical handovers and mobility management in the heterogeneous environment, seamless mobility is not possible without any extensions to these protocols. The study consists of a definition for the new DRiWE protocol. The main effort with the DRiWE protocol was to study and evaluate mobility management in devices with several access technology interfaces available. This study with a real testbed implementation gives a very good starting point to develop the other mechanism needed for better mobility management.

The research includes a definition and implementation of the VERHO system described. The VERHO system is designed for vertical handover control and is capable of managing the mobile node's network interfaces. The system is also designed to control the mobility protocols and is capable of doing the handover decisions. The design and implementation was aimed at controlling the Mobile IPv6 protocol. VERHO can use different trigger information for decision-making. The information used by the VERHO system can be provided also for the use of the different applications. The VERHO system was validated with a real implementation by using the link layer information provided by the dedicated link layer information provider. Additionally the upper level triggers, collected e.g. by the defined trigger management framework (TRG), may be used as a input for

VERHO.

The very challenging work of collecting and distributing the different events and triggering information originating from network and different layers of the protocol stack is presented as well. The need for this information is due to new sophisticated mobile devices. Current and future mobile devices are capable of running demanding network applications and may have support for multiple access technology. Thus it is mandatory to get the necessary information, e.g. for triggering handovers. This research problem is approached with the definition of the new triggering management framework (TRG). By using the defined mechanism, TRG framework enables easy and efficient use of cross-layer and cross-domain information. This framework was implemented and evaluated by performing tests in a real environment with several operating systems to prove its robustness and to measure its performance. The TRG framework experiments, with their related performance tests and evaluations, indicate that the implemented TRG functionalities are very promising.

As a part of the future mobility management and the validation work, a new concept called distributed control and management framework (DCMF) was created. The DCMF is designed to support distributed information exchange between operators, enable advanced access control, and allow for different optimization techniques such as balancing traffic load between access points. The basis of this framework is the TRG functionality that provides a unified signaling architecture between network nodes, capable of serving different entities, located either in the network or in the terminals and controlled by different players (terminals by the user, network by the service provider). The DCMF concept with the TRG implementation was validated in a real heterogeneous network. The validation tests and evaluations show that the use of mobility management related information from both the mobile node and network will give a significantly better quality of experience for the users.

As a part of the author’s current work and future agenda, the development of the TRG and DCMF functionalities and concepts will continue. The aim is to extend the solutions to serve better the future internet architectures and intelligent cognitive network architectures. This work has already started, and for example TRG is already a part of a multiaccess dissemination networks solution, defined in the 4WARD [2] project and described in [54, 68]. The work for intelligent and cognitive network management has started as well, for example, within the frameworks of the Univerself [77] Project and the Future Internet Research Programme [17].

YHTEENVETO (FINNISH SUMMARY)

Eri teknologioilla toteutetut langattomat verkot mahdollistavat palveluntarjoajille ja käyttäjille uusia mahdollisuuksia erinäisten langattomien palveluiden toteuttamiseksi ja käyttämiseksi. Yhtenäistä näille palveluille on niiden saatavuus mobiillilaitteilla, missä ja milloin tahansa käyttäjät haluavat.

Eri verkkoteknologioiden moninaisuus asettaa haasteita yhteyden muodostukselle. Uusimmissa älypuhelimissa on käytettävissä useita eri liityntätekniikoita. Jotta palvelut olisivat saatavilla, kuten sanottu milloin ja missä tahansa, täytyy laitteiden pystyä käyttämään eri verkkoteknologioilla toteutettuja verkkoja. Palvelun katkeamattomuus verkkojen vaihtojen välillä ja niin sanottu saumaton yhteys nousee avainasemaan liikuttaessa eri verkkotekniikoiden välillä. Saumattoman yhteyden totetutus eri tekniikoiden välillä on eritttäin haasteellista ja tarvitsee protokollia ja tekniikoita, jotka mahdollistavat yhteyden vaihdon tekniikasta toiseen.

Tämä väitöskirja, jonka nimi on Liikkuvuuden hallinta heterogeenisissä IP-verkoissa, tutkii liikkuvuuden hallintaa eri teknologioiden välillä. Työ keskittyy IP-protokollalla toimivien verkkojen toimintaan ja niissä liikkuvuuden tueksi tarvittavien protokollien ja menetelmien tutkimiseen. Väitöskirjassa käydään läpi useita eri liikkuvuutta tukevia protokollia ja menetelmiä. Väitöskirja esittelee työn tuloksena syntyneet liikkuvuudenhallintaprotokollan, järjestelmän yhteyden vaihtamisen kontrolloimiseksi eri verkkotekniikoiden välillä sekä kehitetyn mekanismin liikkuvuuden hallinnassa käytettävän tiedon keräämiseksi, prosessoimiseksi ja levittämiseksi. Mekanismien tavoitteena on edesauttaa saumatonta liikkumista eri verkkoteknologioiden välillä sekä tarjota käyttäjille aina paras ja optimoitu yhteys. Lisäksi työssä esitetään menetelmä hajautetun liikkuvuuden hallinnan toteuttamiseksi.

REFERENCES

[1] 3GPP TS 23.234. 3GPP system to Wireless Local Area Network (WLAN) interworking; System description, Available: http://www.3gpp.org/ftp/Specs/html-info/23234.htm. Referenced 15.6.2011.

[2] 4WARD project, Available: http://www.4ward-project.eu/. Referenced 15.6.2011.

[3] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. Number 2581 in Request for Comments. IETF, April 1999.

[4] S. Aust, D. Proetel, N.A. Fikouras, C. Pampu, and C. Görg. Policy based Mobile IP Handoff Decision (POLIMAND) using Generic Link Layer Information. In *the 5th IEEE International Conference on Mobile and Wireless Communication Networks (MWCN 2003)*, October 2003.

[5] F. Cacace and L. Vollero. Managing mobility and adaptation in upcoming 802.21 enabled devices. In *Proceedings of the 4th international workshop on Wireless mobile applications and services on WLAN hotspots*, WMASH '06, pages 1–10, New York, NY, USA, 2006. ACM.

[6] E. Casalicchio, V. Cardellini, and S. Tucci. A layer-2 trigger to improve QoS in content and session-oriented mobile services. In *Proceedings of the 8th ACM international Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 95–102, Montral, Quebec, Canada, October 2005.

[7] H. Chaouchi and Antunes. P.M. Pre-handover signalling for QoS aware mobility management. In *International Journal of Network Management*, 2005.

[8] Converging Networks Laboratory CNL, VTT, available: http://www.cnl.fi. Referenced 15.6.2011.

[9] L. Daniel and M. Kojo. Employing cross-layer assisted tcp algorithms to improve tcp performance with vertical handoffs. *International Journal on Communication Networks and Distributed System*, 1(4/5/6):433 – 465, 2008.

[10] V. Devarapalli, Wakikawa. R, A. Petrescu, and P. Thubert. Network Mobility (NEMO) Basic Support Protocol. RFC 3963 (Proposed Standard), January 2005.

[11] C. Dousson, K. Pentikousis, T. Sutinen, and J. Makela. Chronicle recognition for mobility management triggers. In *Computers and Communications, 2007. ISCC 2007. 12th IEEE Symposium on*, pages 305 –310, july 2007.

[12] R. Droms. Dynamic Host Configuration Protocol. Number 2131 in Request for Comments. IETF, March 1997.

[13] H.A. Duran-Limon, G.S. Blair, A. Friday, T. Sivaharan, and G. Samartzidis. A Resource and QoS Management Framework for a Real-time Event System in Mobile Ad Hoc Environments. In *Int. Workshop Object-Oriented Real-Time Dependable Systems (WORDS)*, pages 2175–224, 2003.

[14] J. Eisl and et al. Mobility architecture & framework - d4.2 core report, 2005. IST-2002-507134-AN/WP4/D4.2.

[15] S. Frei, W. Fuhrmann, A. Rinkel, and B.V. Ghita. Improvements to inter system handover in the epc environment. In *New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on*, pages 1 –5, feb. 2011.

[16] S. Fu and M. Atiquzzaman. TraSH: A Transport Layer Seamless Handover Scheme for Space Networks. In *Nasa's earth science technology conference*, 2005.

[17] Future Internet, Finnish ICT SHOK Research Programme, Available: http://www.futureinternet.fi/. Referenced 15.6.2011.

[18] R. Giaffreda, K. Pentikousis, E. Hepworth, R. Agüero, and A. Galis. An information service infrastructure for ambient networks. In *Parallel and Distributed Computing and Networks'07*, pages 19–25, 2007.

[19] S. Gundavelli, K. Leung, Devarapalli.V, K. Chowdhury, and B. Patil. Proxy Mobile IPv6. Number 5213 in Request for Comments. IETF, August 2008.

[20] A. Gurtov, editor. *Host Identity Protocol (HIP): Towards the Secure Mobile Internet*. Wiley & Sons, 2008.

[21] C. Hedrick. Routing Information Protocol. Number 1058 in Request for Comments. IETF, June 1988.

[22] T.R. Henderson. Host mobility for IP networks: a comparison. *Network, IEEE*, 17(6):18 – 26, nov.-dec. 2003.

[23] R. Hinden and S. Deering. Internet Protocol Version 6 (IPv6) addressing Architecture. Number 2373 in Request for Comments. IETF, April 2003.

[24] S. Horsmanheimo, J. Eskelinen, and H Kokkoniemi-Tarkkanen. Nes - network expert system for heterogeneous networks. In *IEEE ICT2010 International Conference on Telecommunications 2010*, Doha Qatar, April 2010.

[25] J. Huusko, J. Vehkaperä, P. Amon, C. Lamy-Bergot, G. Panza, J. Peltola, and M.G. Martini. Cross-layer architecture for scalable video transmission in wireless network. *Elsevier Signal Processing: Image Communication 22*, April 2007.

[26] IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services, IEEE 802.21.

[27] S. Ishihara, K. Koyama, G. Miyamoto, and M. Kuroda. Predictive rate control for realtime video streaming with network triggered handover. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 3, pages 1335 – 1340 Vol. 3, March 2005.

[28] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. Number 3775 in Request for Comments. IETF, June 2004.

[29] C. Kappler, P. Mendes, C. Prehofer, P. Pöyhönen, and D. Zhou. A framework for self-organized network composition. In *Lecture Notes in Computer Science*, number 3457, pages 2288–2304. Springer, 2006.

[30] BumWook Kim, SungHoon Seo, and JooSeok Song. An effective codec adaptation for mih based vertical handover in heterogeneous wireless networks. In *Information Science and Engineering, 2008. ISISE '08. International Symposium on*, volume 2, pages 80 –83, December 2008.

[31] R. Koodli. Fast Handovers for Mobile IPv6. Number 4068 in Request for Comments. IETF, July 2005.

[32] S. Krishnan, N. Montavont, E. Njedjou, S. Veerepalli, and A. Yegin. Link-layer Event Notifications for Detecting Network Attachments. Request for Comments. IETF, August 2007.

[33] D. Levi, P. Meyer, and B. Stewart. Simple network management protocol (snmp) applications. *Internet Engineering Task Force, RFC 3413*, 2002.

[34] M. Luoto and T. Sutinen. Cross-layer Enhanced Mobility Management in Heterogeneous Networks. In *International Conference on Communications*, Beijing, China, May 2008.

[35] C.H. Lwi, H. Mohanty, and R.K. Ghosh. Causal Ordering in Event Notification Service Systems for Mobile Users. In *International Conference on Information Technology: Coding and Computing (ITCC)*, volume 2, pages 735–740, 2004.

[36] M. Majanen, K. Pentikousis, and J. Mäkelä. Ambient networks gateway selection architecture. *International Journal On Advances in Networks and Services*, 1(1):52 – 63, 2009.

[37] J. Mäkelä. Dynamic routing for the wireless devices in IP-networks. Master's Thesis, University of Jyväskylä, 2003.

[38] J. Mäkelä and M. Luoto. Distributed Control and Management Framework for mobile and multi-access communication. In *Proceeding of 5th International Mobile Multimedia Communications Conference, MobiMedia 2008*, July 2008.

[39] J. Mäkelä, K. Pentikousis, M. Majanen, and J. Huusko. Trigger Management and Mobile Node Cooperation. In M. Katz and F. H.P Fitzek, editors, *Cognitive Wireless Networks; Concepts, Methodologies and Visions Inspiring the Age*

of Enlightenment of Wireless Communications, pages 199–211. Springer-Verlag, January 2007.

[40] J. Manner and M. Kojo. Mobility Related Terminology. RFC 3753 (Informational), June 2004.

[41] M.G. Martini, M. Mazzotti, C. Lamy-Bergot, J Huusko, and P. Amon. Content adaptive network aware joint optimization of wireless video transmission. *IEEE Communication Magazine, Special Issue on "Advances in Visual Content Analysis and Adaptation for Multimedia Communication*, January 2007.

[42] C. Mingardi, G. Nunzi, D. Dudkowski, and M. Brunner. Event handling in clean-slate future internet management. In *Integrated Network Management, 2009. IM '09. IFIP/IEEE International Symposium on*, pages 275 –278, June 2009.

[43] N. Montavont, E. Njedjou, F. Lebeugle, and T. Noel. Link triggers assisted optimizations for mobile lPv4/v6 vertical handovers. In *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC 2005)*, pages 289 – 294, 2005.

[44] G. Montenegro. Reverse Tunneling for Mobile IP, revised. IETF Request for Comments: 2344, January 2001.

[45] R. Moskowitz and P. Nikander. Host Identity Protocol (HIP) Architecture. Number 4423 in Request for Comments. IETF, May 2006.

[46] N. Niebert, A. Schieder, H. Abramowicz, C. Prehofer, and K. Holger. Ambient Networks – An Architecture for Communication Networks Beyond 3G. *IEEE Wireless Communications (Special Issue on 4G Mobile Communications – Towards Open Wireless Architecture)*, 11(2):14–21, April 2004.

[47] N. Niebert, A. Schieder, J. Zander, and R. Hancock, editors. *Ambient Networks; Co-operative Mobile Networking for the Wireless World*. Wiley & Sons, 2007.

[48] P. Nikander, J. Ylitalo, and J. Wall. Integrating Security, Mobility, and Multihoming in a HIP Way. In *Network and Distributed Systems Security Symposium (NDSS03)*, San Diego, CA, USA, February 2003.

[49] OASIS eXtensible Access Control Markup Language, OASIS specification, Available: http://www.oasis-open.org/committees/xacml/. Referenced 15.6.2011.

[50] P. Pääkkönen, P. Salmela, R. Agüero, and J. Choque. An Integrated Ambient Networks Prototype. In *2007 International Conference on Software, Telecommunications and Computer Networks*, Split - Dubrovnik, Croatia, September 2007.

[51] P. Pääkkonen, P. Salmela, R. Aguero, and J. Choque. Performance analysis of hip-based mobility and triggering. In *World of Wireless, Mobile and Multimedia*

Networks, 2008. WoWMoM 2008. 2008 International Symposium on a, pages 1 –9, June 2008.

[52] K. Pahlavan, P. Krishnamurthy, A. Hatami, M. Ylianttila, J.P. Makela, R. Pichna, and J. Vallstron. Handoff in hybrid mobile data networks. *Personal Communications, IEEE*, 7(2):34 –47, April 2000.

[53] K. Pentikousis, R. Agüero, J. Gebert, J.A. Galache, O. Blume, and P. Pääkkönen. The Ambient Networks heterogeneous access selection architecture. In *First Ambient Networks Workshop on Mobility, Multiaccess, and Network Management (M2NM)*, pages 49–54, Sydney, Australia, October 2007.

[54] K. Pentikousis and T. Rautio. A multiaccess network of information. In *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a*, pages 1 –9, June 2010.

[55] C. Perkins. IP Encapsulation within IP. Number 2003 in Request for Comments. IETF, October 1996.

[56] C. Perkins. *Mobile IP: Design Principles and Practices*. Addison-Wesley, 1998.

[57] C. Perkins. IP Mobility Support for IPv4. Number 3344 in Request for Comments. IETF, August 2002.

[58] C. Pinho, J. Ruela, K. Pentikousis, and C. Kappler. A protocol for event distribution in next-generation dynamic networks. In *Fourth EURO-NGI Conference on Next Generation Internet Networks (NGI)*, Krakow, Poland, April 2008.

[59] E. Piri, T. Sutinen, and J. Vehkaperä. Cross-layer architecture for adaptive real-time multimedia in heterogeneous network environment. In *Wireless Conference, 2009. EW 2009. European*, pages 293 –297, May 2009.

[60] G.P. Pollini. Trends in handover design. *Communications Magazine, IEEE*, 34(3):82 –90, March 1996.

[61] J. Postel. User Datagram Protocol. Number 768 in Request for Comments. IETF, August 1980.

[62] J. Postel. Internet Protocol - DARPA Internet Program Protocol Specification. Number 791 in Request for Comments. IETF, September 1981.

[63] P. Prasad, W. Mohr, and W. Konhäuser. *Third Generation Mobile Communication Systems*. MA: Artech House Publishers, Boston, 2005.

[64] J. Prokkola, M. Hanski, M. Jurvansuu, and M. Immonen. Measuring WCDMA and HSDPA delay characteristics with QoSMeT. In *ICC'07*, Glasgow, Scotland, June 2007.

[65] P Ruuska, J Mäkelä, M Jurvansuu, J Huusko, and P Mannersalo, editors. *Roadmap for communication technologies, services and business models for 2010, 2015 and beyond*. Tekes, 2010.

[66] J. Sachs and M. Olsson. Access network discovery and selection in the evolved 3gpp multi-access system architecture. *European Transactions on Telecommunications*, 21(6):544–557, 2010.

[67] C. Simon, R. Rembarz, P. Pääkkönen, and et al. Ambient Networks Integrated Prototype Design and Implementation. In *16th IST Mobile Summit*, pages 1–5, Budaoest, Hungary, July 2007.

[68] M. et al. Soellner. Mobility in the future internet: the 4ward innovations. In *Second Future Internet Cluster Topic Workshop on Mobility and Security*, June 2010.

[69] H Soliman. Mobile IPv6 Support for Dual Stack Hosts and Router. Number 5555 in Request for Comments. IETF, June 2009.

[70] H. Soliman, C. Castelluccia, K. El Malki, and L. Bellier. Hierarchical Mobile IPv6 Mobility Management (HMIPv6). Number 4140 in Request for Comments. IETF, August 2005.

[71] J. Solomon. Mobile-IPv4 Configuration Option for PPP IPCP. Number 2290 in Request for Comments. IETF, February 1998.

[72] T. Sutinen and H. Rivas. Cross-layer Assisted Network Interface Selection for Multi-interface Video Streaming. In *The IEEE International Conference on Computer Communications and Networks (ICCCN 2011)*, 2011.

[73] The 3rd Generation Partnership Project (3GPP), Available: http://www.3gpp.org/. Referenced 15.6.2011.

[74] E. Tuulari and Arto Ylisaukko-oja. SoapBox: A Platform for Ubiquitous Computing Research and Applications. In *Lecture Notes in Computer Science 2414: Pervasive Computing*, pages 125–138, August 26–28, Zürich, CH, August 2002. Springer.

[75] V. Typpö, O. Gonsa, T. Rinta-aho, B. B. Tharon, and S. Bonjour. Triggering Multi-dimensional Mobility in Ambient Networks. In *14th IST Mobile & Wireless Communications Summit*, June 19–22, Dresden, Germany, June 2005.

[76] UMIP: USAGI-patched Mobile IPv6 for Linux, Available: http://umip.linux-ipv6.org/. Referenced 15.6.2011.

[77] Univerself Project, Available: http://www.univerself-project.eu/. Referenced 15.6.2011.

[78] J. Vesanto and E. Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on neural networks*, 11(3), 2000.

[79] P. Vidales, J. Baliosian, J. Serrat, G. Mapp, F. Stajano, and A. Hopper. Autonomic system for mobility support in 4g networks. *IEEE Journal on Selected Areas in Communications*, 423(12):2288–2304, 2005.

[80] P. Vidales, R. Chakravorty, and C. Policroniades. Proton: a policy-based solution for future 4g devices. In *Policies for Distributed Systems and Networks, 2004. POLICY 2004. Proceedings. Fifth IEEE International Workshop on*, pages 219 – 222, june 2004.

[81] I. Vivaldi, B.M. Ali, H. Habaebi, V. Prakash, and A. Sali. Routing scheme for macro mobility handover in hierarchical mobile ipv6 network. In *Telecommunication Technology, 2003. NCTT 2003 Proceedings. 4th National Conference on*, pages 88 – 92, January 2003.

[82] E. Yoneki and J. Bacon. Determination of time and order for event-based middleware in mobile peer-to-peer environments. In *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, pages 91 – 96, March 2005.

[83] H. Zimmermann. Osi reference model–the iso model of architecture for open systems interconnection. *IEEE Transactions on Communications*, 28(4):425–432, 1980.

ORIGINAL PAPERS

PI

SOFT HANDOVER AND ROUTING MECHANISMS FOR MOBILE DEVICES

by

J. Mäkelä, G. Fekete, J. Narikka, T. Hämäläinen and A-M Virkki 2004

Proceedings of The 15th IEEE International Symposium on Personal, Indoor and
Mobile Radio Communications (PIMRC 2004)

SOFT HANDOVER AND ROUTING MECHANISMS FOR MOBILE DEVICES

Jukka Mäkelä[+], Gábor Fekete[++], Jorma Narikka[++], Timo Hämäläinen[+], Anna-Maija Virkki[++]

[+]University of Jyväskylä, Department of Mathematical Information Technology, P.O. Box 35, FIN-40100 Jyväskylä, FINLAND
Email: jumakela@cc.jyu.fi, timoh@cc.jyu.fi
[++]Jyväskylä Polytechnic, School of Information Technology, Piippukatu 2, FIN-40100 Jyväskylä, FINLAND
Email: feketga@delfin.unideb.hu, narjo@jypoly.fi, anna-maija.virkki.iin@jypoly.fi

Abstract- In this paper, we introduce mechanisms, which offer soft handover support between two different IP subnets for the mobile devices. The mobility support protocol that was developed tries to solve the same problem as Mobile IP, that is, how to manage up to date information about the location of a moving node and to keeps its connections established during its movement. It uses soft handover and relies on accurate routing information of routers participating in the protocol. The protocol work entirely in user space (Layer 7, 6 and 5 of the OSI model) and considers the handover only in layer 3.

Keywords: Network protocol, handover, routing, mobility.

I. INTRODUCTION

The need for handling mobility in IP networks to serve real-time traffic has increased nowadays. In the future there will be more and more mobile devices that will use IP based applications. Mobile IPv4 [1], Mobile IPv6 [2] [3] and Hierarchical Mobile IPv6 (HMIPv6) [4] [5] [6] are proposals for techniques to handle mobility. But it is still unsolved how to minimize the handover time when moving between two access routers that reside in different subnets. In mobile IP it is needed to send Binding Update messages to the home agent when doing the handover. The home agent can be quite far from the mobile node and that causes the signaling time to increase. During the past several years, there has been a lot of research about handover. It is proposed that mobile node can update the information first with the access routers to increase the speed of the handover [3]. The current IETF draft about fast handover [7] is developing the idea further so that the handover latency could be minimized and the handover would be more beneficial to real-time traffic. The proposed extension HMIPv6 to Mobile IPv6 introduces a new local home agent called mobility anchor point (MAP). The MAP is supposed to be closer to the mobile node than to its original home agent. The mobile node can do the updating first with this local MAP and after that with its own home agent. So the signaling latency during the handover can be reduced. Mechanisms where the mobile node can receive packets during signaling the update are also introduced. However, the protocol still cannot meet the requirement for traffic that is delay sensitive, such as voice, especially in macro mobility management [8].

In this paper, we present ideas about a protocol that tries to solve mobility problems in IP networks (first presented in [9] and [10]). This protocol has a handover mechanism, which achieves the soft handover without additional buffering. The basic idea is that in IP networks there are several possible techniques available to make the IP based connection (e.g. WLAN, GPRS or EDGE). For using different techniques, the mobile node should also have different interfaces for each technique. These interfaces can be used simultaneously during the handover. The handover is made only between the access routers. The mobile node does not have to register its new location with the HA all the time. This paper introduces also some ideas how the routing could be done in IP networks so that the location of the mobile node is updated by the routers when needed. Changing the way of the routing offers lot of possibilities to increase the speed of the handover but still needs more research

II. DRIWE PROTOCOL

The DRiWE (Dynamic Routing in Wireless Environment) protocol tries to solve the problem of the handover and updating the location information about mobile nodes that arise when mobile devices are changing their location while they are connected to the network. The protocol considers only the mechanisms to handle the movement but the issues, as authentication, are out of scope of this paper. The protocol considers only Layer 3 handovers.

To make the soft handover possible, the mobile node uses two different interfaces during the handover. The two interfaces are used only during the handover so that the current connection is used until the new connection is totally established and ready for use for data flow. The possibility to use two interfaces is based on the idea that when performing the vertical handovers the mobile node has many interfaces e.g. WLAN and GPRS.

The routing part of the protocol is based on the idea that the routing decisions take place only in routers and every router, which participates the protocol, knows the location of the mobile node when needed. The way to have the current information in routers is to use a host specified entry about the mobile nodes in the routing tables. In addition, a mechanism called Mobile Node Discovery is introduced so that the location of the mobile node can be found even when the routers drop some of the host specified routes from their routing tables to avoid large routing tables. Mobile Node Discovery is a necessary mechanism if routers are allowed to drop unused routes from the routing table so that the route to mobile node could be found even when the routers don't have the information about the location of the mobile node. The protocol includes also the advertising mechanism so that routers can propagate the information about the location of the mobile nodes to other routers.

By using this kind of information and techniques in routers, it is not necessary to have any support for the protocol in corresponding nodes. A mobile node registers itself with the possible new access router. The current access router has the information about every mobile node in its routing table that is currently connected to it, in its routing table.

III. FUNCTIONALITY OF THE PROTOCOL

When the mobile node is connected to an IP network from the same network where it has the fixed IP address, the packets are routed with normal routing protocols without any problems. When the mobile node needs to move and connect to other access router on a different network (AR2, in Fig. 1) it has to complete a Layer 3 handover and it will need a care-of address from the new access router (AR2, in Fig. 1) in new network. The DRiWE protocol was tested in a wired network so the care-of address query takes place through the current access router (AR1, in Fig. 1).

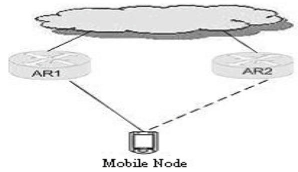


Fig. 1. The handover.

After the mobile node has got the care of address from the new access router (AR2, in Fig. 1), it creates a tunnel towards the new access router (AR2, in Fig. 1). Now the mobile node can use its fixed IP address as the source for outgoing packets. The packets go tunneled to the access router by using the care-of address as the source and the access routers IP address as the destination for the tunnel packet. The access router decapsulates the packet and sends the original packet to the destination. Those packets that are destined to the mobile node, are routed to their destination by using the fixed IP address of the mobile node as the destination. The access router doesn't have to tunnel those packets. During the handover, the mobile node accepts incoming packets from both of its interfaces. Therefore, there is no need to stop the data flow at any time.

When the mobile node has attached to a new access router (AR2, in Fig. 1) the new access router will send an update about the new location of the mobile node to the current access router (AR1, in Fig. 1). Now the current access router updates its routing table and forwards packets destined to the mobile node to the new access router (AR2, in Fig. 1). The new access router routes those packets to the mobile node. The mobile node also changes its default outgoing route to go through the tunnel towards the new access router (AR2, in Fig. 1). The mobile node can keep the old interface and connection up. So it can "prepare" itself to situations like changing back to the old access router (AR1, in Fig. 1) (where it may be able to use already allocated resources), especially when the current connection is lost and the older connection is still available. When another handover is needed all allocated resources from the old interface is released and that interface is used for configuration with the new access router.

In order to keep the connections up, the mobile node sends Keep Alive messages to the access routers and the mobile node can easily notice when the link is actually down. TCP protocol sees this link failure only as lost packets although the rate of sending Keep Alive messages should be carefully chosen not to cause network congestion.

Using a mechanism where the routers send advertisement to their neighbors can handle the mobile node's location advertisement. The decision when advertisements are sent should be made by mobile node because if the mobile node is moving only inside a small area, it's probably enough to update the location only in and between access routers by using the Update mechanism.

A. The mobile nodes functionality in the handover

It is assumed that the mobile node needs to connect to the new access router (AR2, in Fig. 1) and the mobile node also knows the new access router's IP address by some kind of movement detection. The mobile node must also have two interfaces.

When the mobile node wants to connect to the new access router it can ask the registration and a care-of address from the new access router (AR2, in Fig. 1) by sending a CoA Request message to it. The mobile node waits for a specified time for the new care-of address to arrive included in the CoA Reply message. If the time specified by the timer elapses or the mobile node receives an error message as a reply, then the registration and allocating the new care-of address fails and the mobile node cannot connect to that access router. The mobile node can then try to connect to other possible access routers by sending the CoA Request message to them.

When the registration is successful and the mobile node has the care-of address, it establishes a tunnel towards the new access router (AR2, in Fig. 1). The tunneling is used for the outgoing packets destined to other destinations than the access router. After that the mobile node sends an Update Request message to the new access router (AR2, in Fig. 1). By sending this message, the mobile node asks the new access router to change its routing table so the access router will start forwarding the packets directly to the mobile node and sends the update about the location of the mobile node to the old access router (AR1, in Fig. 1). The mobile node waits for an acknowledgement message of the Update Request. Now the mobile node can receive packets from both access routers. When the mobile node gets the acknowledgement it can change the default route to the new access router (AR2, in Fig. 1) and handover ends. The older connection can be kept alive by using a Keep-Alive mechanism introduced later in this document.

If the mobile node does not receive the acknowledgement in time or it receives an error message, it sends an Update Request message to the current access router (AR1, in Fig. 1) so it can start a mechanism called Update Recovery. This mechanism is used to recover the routing tables of intermediate routers, between the new and the current access router that may have been changed because of a failed handover (a possibly failed Update mechanism). The Update Recovery at the current access router (AR1, in Fig. 1) is started by the mobile node. The new access router (AR2, in Fig. 1) starts it itself if it detects that the Update mechanism has failed. This mechanism is presented later in this paper. In this case, the mobile node doesn’t change the default route and the resources reserved for the new connection can be released.

In the situation when the mobile node is moving back to the earlier access router (AR1, in Fig. 1) and has both the old and the current connection available, it can use already allocated resources. The mobile node can just send the update request to the earlier access router (AR1, in Fig. 1) that will send an update to the intermediate router between the access points. This mechanism can have also timer included so that messaging in the situation called ping-pong can be decreased. In ping-pong situation, when the mobile node is moving back and forth between two access routers, the update mechanism may take too much time and is unnecessary. In this case the default route is not changed and the update message is not sent immediately, only after some time, so that the mobile node actually uses the same route all the time, if possible.

When new handover is needed and all interfaces are used, the mobile node releases the resources from the interface used for connecting to the old access router (AR1, in Fig. 1) and uses this released interface to establish the new connection.

When the mobile node is connected to the access routers, it uses the mechanism called Keep-Alive to keep the connections established. This mechanism is quite simple. The mobile node sends Keep Alive messages to those access routers it wants to keep the registration alive with. The mobile node waits for similar messages from the access routers. When it fails to receive Keep Alive messages within the specified time, the mobile node treats the connection as broken. The mobile node starts this mechanism after it has established the connection and tunnel with access router. Access routers send Keep Alive messages to all mobile nodes attached to them.

B. The access routers functionality in the handover

When the access router receives the CoA Request message from the mobile node, it registers the mobile node and allocates the care-of address for it. If the mobile node has already registered (the fixed IP address in the message is the same as already registered) then the access router can use already allocated resources for the mobile node. The access router sends a CoA Reply message to the mobile node as a response. The care-of address is included in that message. If the registration or allocating the resources fails, the access router sends an error message to the mobile node as a response. The Keep Alive mechanism is started after the access router has sent the CoA Reply message. If the mobile node managed to establish the connection, the new access router should get the Update Request and Keep Alive messages from the mobile node.

When the access router receives the Update Request message it adds the fixed IP address of the mobile node into the routing table and starts forwarding the packets directly to the mobile node. The access router also sends an Update message about the new location of the mobile node destined to the previous access router (AR1, in Fig. 1). In the DRiWE protocol, the behavior of the mechanism used needs the Update message to be handled at every router between the access routers. This is necessary because those intermediate routers have to update their information about the location of the mobile node accordingly. This processing of the Update message by the intermediate routers could be dropped if tunneling would be used, instead of normal routing, to forward packets from the old access router (AR1, in Fig. 1) to the new access router. The Update message is started from the new access router (AR2, in Fig. 1) and sent to the next router on the route to the destination (AR1, in Fig. 1). When the first intermediate router receives this message it adds the fixed IP address of the mobile node included in the message to its routing table. The next hop for the entry will be the source IP address of the router from which this router received the message. Then the router sends the packet to the next router in route to the destination. Every intermediate router will do the same. The IP address of the earlier access router (AR1, in Fig. 1) is also included in the message so that routers can lookup the next hop for it in their routing tables.

When the previous access router (AR1, in Fig. 1) gets this Update message it changes its routing table and starts forwarding the packets to the mobile node through the new access router. It will also send an acknowledgement, about receiving the Update to the new access router (AR2, in Fig. 1). After the update mechanism has finished, every intermediate router has the information about the location of the mobile node and the previous access router (AR1, in Fig. 1) can immediately forward the packets to the new access router (AR2, in Fig. 1).

The new access router (AR2, in Fig. 1), which sends this Update message, waits for an acknowledgement sent by the earlier access router (AR1, in Fig. 1) as a response. When it receives the acknowledgement it also sends an acknowledgment to the mobile node that is waiting for the acknowledgement of the Update Request. If the access router fails to get the acknowledgment it sends an error message to the mobile node. It also automatically starts the Update Recovery mechanism to recover the situation back to the form it was before the handover in access routers and in routers between them.

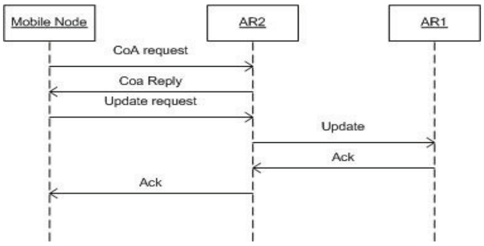


Fig. 3. Successfully handover.

If the update mechanism fails it can cause the situation that the previous and some intermediate routers have not received the information about the new location of the mobile. The Update Recovery mechanism is introduced to solve this situation. This mechanism is similar to sending an Update message but in the reverse way. The new access router (AR2, in Fig. 1) first changes its routing table to undo the effect of the Update Request and sends the Recovery Update message destined to the previous access router (AR1, in Fig. 1) and waits for an acknowledgement. All routers that get this information change the next hop value of the entry in their routing tables for the mobile node in question to the value, which is the next hop towards the destination of the Update Recovery message (AR1, in Fig. 1). Every router waits for an acknowledgement and if it fails to receive the acknowledgement or the message cannot be sent further, it sends an acknowledgment back to the router it received the message from.

Also the mobile node initiates the Update Recovery when it receives an error message or fails to receive the acknowledgement in time from the new access router (AR2, in Fig. 1) as a response to the Update Request message. In this case the mobile node sends an Update Request to the current access router (AR1, in Fig. 1). So the recovery may propagate from both directions, from the current access router (AR1, in Fig. 1) and from the new access router (AR2, in Fig. 1). It is done this way because the route between them might be damaged, since the update mechanism has failed and one recovery message may correct only one half of the route. If the two messages overlap, it does not cause problems, only some overhead and redundancy in the recovery.

C. Advertising mobile node locations

This paper considers the handover more than other parts of the protocol, so that soft handover can be used and packet loss is minimized during the handover. Some ideas about handling the routing are described but the fact is that more research is needed to solve the problems that arise when the mechanisms introduced in this paper are used.

The information about the location of the mobile node is advertised to other routers by using a fairly simple mechanism. The basic idea is that routers should always have or be able to get the information they need about a mobile node's location. Routers can get the information needed by Update messages sent by access routers, by receiving advertisements about the location of the mobile node or using the Mobile Node Discovery mechanism. Every router, which needs the information about a mobile node's location, must have the support for this protocol.

The decision when the advertisements start to propagate is made by the mobile node. When the mobile node is moving fast it is not needed to send advertisements after every handover because the information the advertisement would contain would become out of date quite soon. The mobile node should manage a counter that counts the number of hops it has moved since it last requested the sending of an advertisement. A timer can also be used to trigger the advertisement, for example in situations when the mobile node moves to a location which is not far enough to start the advertisement mechanism but the mobile node stays at that position for a longer period.

The advertisement messages can be sent between routers using UDP. A message includes the fixed IP address of the mobile node, the IP address of the sender (the next- hop) and metric information. The metric information shows the number of routers that packet has to pass through on its way to the mobile node (hop count). Every router sends an advertisement message to its neighbors. An advertisement message must also include a timestamp, so that routers can drop those advertisements that are not up to date.

The routing table growth could cause problems when using host specified routes. To avoid this problem the routers should be able to remove information that is not used by ongoing connections or wasn't used recently. Removing the routes can cause the situation that a router fails to know the information needed to forward packets destined to a mobile node. To handle this situation the mechanism called Mobile Node Discovery is introduced. By using Mobile Node Discovery, routers can find the current location of the mobile node. The idea is that, the router asks the wanted information from neighboring routers recursively and the router, which knows the information, sends an answer to the one that needs it.

IV. IMPLEMENTATION AND TESTING

Due to the fact that the routing part still needs more research, only the handover part was implemented. The implementation was made to work entirely in user space and the handover was implemented to be testable (and usable) both with UDP and TCP. The implementation that was tested was the first version so there should be possible speed improvements both in the TCP and UDP versions. Also note that security was not considered. The environment used in the testing was a wired network. Normal PCs, with Linux as their operating system, were configured to work as routers. Mobile and correspondent nodes were also normal PCs.

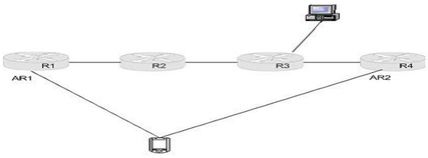


Fig. 3. Test scenario.

Fig. 3 presents testing environment. The AR1 and AR2 were working as the access routers when testing the handover between two access routers. The testing was made by using ping and ssh connection between the mobile and the correspondent node. First the correspondent node was connected directly to R1 that was the initial starting point of the mobile node and after handing over to R4 the communication between the correspondent node and the mobile node went through R3.

The handover worked as expected. The time it takes for the handover mechanism in mobile node was measured with the UDP and the TCP protocols. Handover with UDP took in average 5.3 milliseconds and with TCP 5.6 milliseconds. The connection did not break during the handover.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have presented the mechanisms for handling the handover without losing connections and ideas how the routing could be handled with using host specified routes. To get more information about the effectiveness of the mechanisms introduced, more testing scenarios is going to be studied and tested. The update and the update recovery mechanisms are also going to be tested as a part of the erroneous case of the handover. The handover could be tested as a part of Mobile IP functionality. Also the handling of the update between access routers could be different e.g. using tunneling or this mechanism could be implemented right in the Linux kernel itself below Layer 3. The routing can be handled by using this kind of technique but more research is still needed if this kind of functionality is going to be used. The protocol can be implemented in both IPv4 and IPv6, although it was implemented and tested with IPv4 but the implementation can be modified to work with IPv6.

VI. REFERENCES

[1] C. Perkins, “IP Mobility Support for IPv4”, IETF RFC 3344, August 2002.
[2] D. Johnson, C. Perkins and J. Arkko, “Mobility Support in IPv6”, IETF draft, work in progress, 2003.
[3] C. Perkins and D. Johnson, “Mobility support in IPv6”, In Proceedings of the Second Annual International Conference on Mobile Computing and Networking (MobiCom’96), Rye, New York, USA, November 1996.
[4] H. Soliman, C. Castelluccia, K. El-Malki, L. Bellier, “Hierarchial Mobile IPv6 mobility management (HMIPv6)”, IETF draft, work in progress.
[5] C. Castelluccia, “HMIPv6: A Hierarchical Mobile IPv6 Proposal” ACM Mobile Computing and Communications Review, vol.4, no.1, January 2000, Pages 48-59.
[6] C. Castelluccia, “A Hierarchical Mobility Management Scheme for IPv6”, Proceedings of the Third IEEE Symposium on Computers and Communications 1998 (ISCC ’98), 30 June-2 July 1998, Page(s): 305-309.
[7] R. Koodli, “Fast Handovers for Mobile IPv6”, IETF draft, work in progress, 2003.
[8] I. Vivaldi, B.M. Ali, H. Habaebi, V. Prakash, A. Sali, “Routing scheme for macro mobility handover in hierarchical mobile IPv6 network”, Proceedings of the 4th National Conference on Telecommunication Technology 2003 (NCTT 2003), 14-15 Jan. 2003, Page(s): 88-92.
[9] J. Mäkelä, “Dynamic routing for the wireless devices in IP-networks”. Master’s Thesis, University of Jyväskylä, in finnish, 2003.
[10] G. Fekete, “The implementation of the DRiWE Mobility Support Protocol”. Bachelor’s Thesis, Jyväskylä Polytechnic, 2003.

PII

INTELLIGENT VERTICAL HANDOVER SYSTEM FOR MOBILE CLIENTS

by

J. Mäkelä, T. Hämäläinen, G. Fekete, and J. Narikka 2004

Proceedings of the 3rd International Conference on Emerging
Telecommunications Technologies and Apllications (ICETA2004)

Reproduced with kind permission of elfa s.r.o..

Intelligent Vertical Handover System for Mobile Clients

Jukka Mäkelä, Timo Hämäläinen
University of Jyväskylä, Department of Mathematical Information Technology, PL 35, 40100 Jyväskylä, FINLAND
E-mails: jumakela@cc.jyu.fi, timoh@mit.jyu.fi

Gábor Fekete
University of Debrecen, Institute of Informatics, Egyetem ter 1, 4010 Debrecen, HUNGARY
E-mail: feketga@delfin.unideb.hu

Jorma Narikka
Jyväskylä Polytechnic, School of Information Technology, Piippukatu 2, 40100 Jyväskylä, FINLAND
E-mail: jorma.narikka@jypoly.fi

Abstract

This paper introduces a technical view of handling the change of the available network connections in mobile device with system that offers also mechanisms to provide the unbroken connections in heterogeneous IP-network environment. System is beneficial especially for the streaming softwares when a user is moving through heterogeneous areas where the connection is available during movement but through different techniques.

1. Introduction

The number of mobile devices is rising and the new wireless techniques will offer new possibilities for the users. The rising availability of wireless networks provides possibility to use e.g. some streaming software for e-learning purposes wherever the wireless connections are available. The development for effective use of the advantages of IP-networks requires several new features from the infrastructure. Obviously one of them is that people like to have more freedom considering the time and place, when and where they are using IP-connection. While these requirements are set, one must focus on the issues like how to manage mobility, how to optimize the bandwidth and costs, etc. Future mobile devices will presumably have several different access technology interfaces (e.g. IEEE 802.11, GPRS and Bluetooth) for the connections to the IP-networks. The next generation technology will be introduced in the coming years and it will be one technology more among the others.

The Mobile IPv6 protocol (MIPv6) [5,6] is introduced to handle mobility in future IPv6 networks. Different mechanisms are also introduced to enable the efficient way to change between IP-networks while moving. This change of the connection point of network is called handover or handoff. However, MIPv6 handover suffers from a number of drawbacks such as high handover latency, packet loss, due to the procedures it has to accomplish during its handover phase [14]. The problem in MIPv6 handover is that the connection breaks during the handover, which causes packet loss. In MIPv6 the signaling between the Mobile Node (MN) and home network router called Home Agent (HA) is needed. The distance between HA and MN can be substantial which increases the handover time. A number of research has been done to decrease the handover time. The Hierarchical Mobile IPv6 [3, 4, 7] introduces new Mobility Anchor Point (MAP) that is acting as a local HA for MN. The purpose is that the signaling is needed first only with MAP that is supposed to be closer than HA and the signaling time can be decreased. Fast handovers for Mobile IPv6 (FMIP) [12] is also one introduced mechanism. In FMIPv6 there is a router in the currently visited network that may work as a proxy for the MN. This proxy router is used to tunnel packets to the new AR in the new visited network. FMIPv6 uses also link layer information that allows starting the handover earlier than would otherwise be possible. Both FMIP and HMIP are drafts under the IETF's MIPv6 Signaling and Handoff Optimization (mipshop) working group [10].

We believe that in the future there is more than one access technology available in parallel. Our research work considers handovers between different wireless technologies in heterogeneous environment. This handover between different access technologies is called vertical handover [9]. Our previous research [11] introduces a routing protocol that provides the handover without HA and achieves also the seamless handovers in small area networks, when at least two interfaces can be used. That research was about the mechanisms for the handover and routing and was not considering the decision part of handover at all.

The optimal situation would be that the data flow could be on all the time during the handover. To achieve seamless handovers, the connection to new access point has to be build before the old connection breaks. Unfortunately, many layer 2 technologies require that the connection to the old access point must be dropped before establishing the new.

When moving in heterogeneous wireless IP networks the decision between different techniques that should be used is needed. Normally, this is based on the availability of the network. We believe that in future there are many wireless networks with different techniques available. The decision should be done in mobile devices because the decision depends much on the services and applications that user want; to use. Such facts like actual cost of using some technique may be quite important for customer. Figure 1 Illustrates a situation in heterogeneous environment when Mobile Node is moving through area with GPRS and couple of IEEE 802.11 hotspots available.

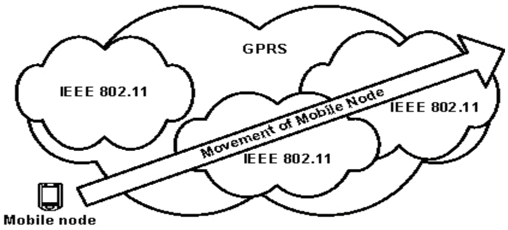


Figure 1. Heterogeneous environment.

In this paper, we introduce the system called VERHO that makes the decision which available access technology should be used. The system uses the information deduced from the decision process and controls the MIPv6 handover. The decision when handover should occur depends on the information collected from each available technology and data flows in situations when several different access technologies are available. The most significant information is the information received from the link layer, e.g., link up/down. The link layer offers different set of information depending on the technologies available. Using the link layer information also improves the Mobile IP handovers. The system will also use several other sources of information, e.g. user specified information and the Quality of Service requirements of applications.

Our goal is to design a system that realize the intelligent decision based on the available information and control the MIPv6 handovers and achieve the seamless vertical handover by performing a proactive handover. The other research like [16] has also introduced the ideas to use several information sources. We agree fully that the solution for decision is needed and it is not too complex to create. We believe also that this kind of system can provide the seamless handover itself by not only controlling the handovers but also triggering e.g., the search of new possible access points.

The rest of the paper is organized as follows: Chapter 2 describes ideas and the functionality of the VERHO system. Chapter 3 introduces the use of the OSI layer 2 information for the movement detection in Mobile Nodes. In Chapter 4, we introduce mechanism for vertical handover. The conclusions and the description of the ongoing research are given in Section 5.

2. Handover decision in mobile client

The system is working completely in Mobile Node side and the system does not need any support from network side. The main idea is to use the information that is available in Mobile Node side. To collect the information for the decision, the system listens e.g., the events in layer 2 and uses the polling for source that does not create any event to listen. The design of the system is going to be such that it can support or can be easily modified to support possible future systems. The future systems can be such systems on the network side, that offer the handling of the authentication, security or the mechanisms for the better quality of service.

If the decision is made in the Mobile Node it allows also the users to give their own demands for the connection. For example, users can predefine the cost of each technique in terms of time. Information can be one parameter for the decision. Also such things like power consumption in the mobile device should be minimized. The applications have their own demands for e.g., needed bandwidth or maximum delay of the link. The most significant information is of course the availability of the link. This availability information can be gathered and derived from the link layer. More detailed description of the link layer information used in our system is described in chapter 3. Figure 2 illustrates the information sources in the Mobile Node for the decision. Other sources of information e.g., location information from the GPS could be also one source for the decision. These other sources need their own mechanisms to produce the information and are out of scope of this paper.

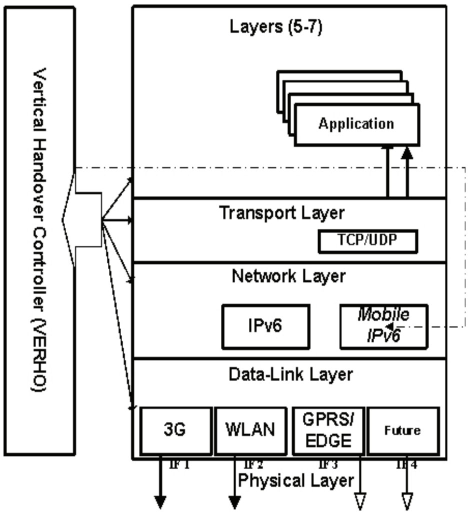


Figure 2. Information sources of the VERHO system.

2.1. Decision mechanism

The decision is based on the defined rules compared to the properties of the interface of available techniques. The rules are divided into critical and non-critical rules. The critical rules are those rules that have to match before the interface is enabled for the decision process. The necessary critical rules are based on the state of the interface and link (Link Up and Link Down). The VERHO system has also its own mechanism to derive the critical information of the link by following the signal strength and the state of the link (see chapter 3). The system provides also the possibility for user to give some critical rules considering e.g., the type of the interface or even the maximum cost of interface to be used. If the minimum requirements of rules match, then interface is available for the decision. Rule can be for example like this: Interface state = UP and Interface type = IEEE 802.11b. This rule means that user wants to use only WLAN. The type of the interface parameter can be also among other parameters with own weighted value. That makes possible to users set the rules like: always use the WLAN, if available.

The decision among those interfaces that match the critical rules are based on the preference number. Each interface will get calculated preference value based on the values of different parameters. Parameters have weighted numerical values so that the most significant parameters will have bigger weight. Each interface has several parameters e.g., cost, power consumption, support for wireless QoS or bandwidth. If some parameter lacks in a specific interface, due to its different technology, that parameter can have a default constant value. The value of the preference number is calculated from the each interfaces parameters values.

One group of parameters can be the measured QoS parameters value. At the time of writing this document, the research of gathering the link quality information is still in progress. But one idea is that e.g., IP-level End-to-End quality can be measured with separate software. Each measured parameter e.g., delay, RTT, and jitter can have then the weighted value for the calculation of preference value of the interface.

There are also parameters like the type of the interface, without any numerical value. Those parameters can be assigned with default numerical value. Deriving the value can be based e.g. on the user demands or e.g. on the speed of the connection.

The system keeps track of the changes in values of each parameter by listening the events and using the polling mechanism. By following the current state of the specific interfaces properties the system is able to choose if there is other interface available that match better with the specified rules.

3. Using Link-layer information

The Mobile IP handover and movement detection is mainly based on the router advertisements and link breaks. The link layer offers the information that can be used as hints for the triggering the handovers. By using the layer 2 information it is possible to achieve more efficient handover. Our system is mainly making the decision, which one of the interfaces available should be used. But by using the link layer information the system can make the decision and control the handovers of Mobile IP in more sophisticated way. It is already showed that the use of link layer information improves the Mobile IP handover e.g., in the [13]. The paper in question introduces a policy based handover decision based on the link layer information. Their solution introduces generic link layer between layer 2 and layer 3 and shows that system with link layer information outperforms Mobile IP handovers.

Our system uses two triggers Link Up and Link Down for triggering the handover. Link Up trigger event corresponds the establishment of a new L2 link, which allows IP communication over it. This is typically a new connection between the Mobile Node and the Access Point. Link Down trigger event corresponds a layer 2 link that has been broken down. This typically occurs when a current connection between the Mobile Node and an access point has been terminated. The interface, which generates this trigger, cannot be used for communication until a connection with an AP is made (Link Up). There are also many L2 parameters that can be used as hints, for example, signal strength, signal to noise ratio, interference, etc. By using these triggers and hints and making proactive handover allows the creation of connection by using other available interface before the current connection breaks.

There is no specification how the link layer triggers should be created or used. The available information that can be used as hints for the upper layers are technique dependent. We have studied the behavior of the IEEE 802.11b and Bluetooth technologies to create triggers for those techniques. The description is in paragraph 3.1. The draft [1] describer the behaviors of GPRS and 3GPP2 (cdma2000) and IEEE 802.11. They suggest that Link Up and Link Down triggers can be created based on the PDP context activation/deactivation in GPRS and bringing down/up the PPP connection in 3GPP. In IEEE 802.11 they suggest that Link Up trigger can be created when Mobile Node receives (Re)Association Response message. Link Down trigger of IEEE 802.11 is created if Mobile Node gets the De-authentication message or a De-association message from its AP.

3.1. Trigger creation in VERHO system

The main research topic is not how to create triggers but some studies have been made as mentioned earlier with IEEE 802.11b and Bluetooth technologies. The system needs two different mechanisms to get the information from the layer 2. The one is way to get the information by listening events that layer 2 techniques offer. That is the way when gathering the information e.g., from IEEE 802.11. The Link Up and Down triggers of IEEE 802.11 for our system can be created based on the signaling messages between Mobile Node and Access Point, like described in [1]. Link Down triggers should be created for VERHO system also if the link is put down from the Mobile Node side.

The other option is to use polling for those access techniques, that NIC does not generate events or triggers. System will use the polling mechanism and monitor the state of the interface for the Link UP and Link Down triggers. The polling mechanism is used also for getting the values of the link layer parameters from drivers. Those parameters are taken into account when calculating the preference value of specific interface. For example, in Bluetooth the polling is needed. We studied the BlueZ [2] protocol stack that provides HCI and L2CAP protocols, which contain information that can be used as triggers or hints. The creation of Link Up or Down triggers is based on the available connection state parameter.

The Signal Strength is considered to be such an reliable hint that can be used also for triggering the handover. For example, if the signal strength of used interface falls below a defined value and there exist another interface with acceptable level of signal strength, the system should change to use that interface. This can also trigger the mechanisms to find the new possible APs in layer 2 e.g., scan mechanism in IEEE 802.11.

4. Vertical handover mechanism

When there is at least two interfaces enabled for the decision (matched with the critical rules) those both can be used for the data flows. In the situations when system notices that the properties of the link that is currently used changes and after calculation of preference value other interface gets better preference value, the system will enable that for use. The system will use only one interface for the data flows. The simultaneous use of interfaces and balancing the load between several interfaces are future topics of our research. When the decision is based on the change of such parameter value that is not critical, (necessary for the interface use) to triggering the handovers should be very careful not to cause any ping-pong effect between interfaces. The handover is necessary if the current connection is lost and it causes the Link Down trigger.

The seamless handover is not possible, if the current connection is lost before new connection establishment, and that is why our system will monitor also the signal strength to provide a proactive handover. If a link goes down or the signal strength falls below acceptable limits, the system should trigger the handover and start to scan for new candidate access points for the interface that fails to match anymore with the critical rules. Reason why to change the interface immediately is because many layer 2 technologies require that the old connection should dropped before creating a new and won’t allow data flows during this scanning phase. If only one interface can be used, the packets are lost during the scanning phase. Some drivers have their own support to handle layer 2 horizontal handovers (between same technology) but unfortunately, there is no mechanism in layer 2 to get the info if also the network is changed. In this situation the layer 3 handover is necessary. Our system is designed to support the vertical handovers when also the access network is changed. It is also possible that there are no new APs available for that technique at all. The supporting of the seamless handover is not the main area in our research but using this kind of behavior in the decision it provides also the seamless handovers if there is no other separate mechanism available.

When trying to scan for new access point, when data flow goes through another interface, might cause interference, when using techniques, which use same frequency e.g., Bluetooth and IEEE 802.11b. The problem is that Bluetooth and IEEE 802.11b don't understand each other. A Bluetooth may start transmitting data while a IEEE 802.11 station is sending a frame. This results in a collision, which forces the 802.11 station to retransmit the frame. This lack of coordination is the basis for RF interference between Bluetooth and 802.11[8]. Fortunately, there are research going on to minimize the interference between different techniques. For example since Bluetooth version 1.2 it's possible to use adaptive frequency hopping to minimize the interference. When interference is minimized the both techniques can be used simultaneously without problems.

5. Conclusions and work in progress

In this paper, we have presented our ideas for the intelligent handover mechanism (VERHO) on the Mobile Node side. The level of QoS can be measured but more research will be made to understand the behaviors of applications and demands during the handover process. The VERHO system will take account more sources of information after some more research work is completed in those fields. Main goal is the full implementation of the described system. The first version of the implementation will use the information sources and mechanism introduced in this paper so that first the efficiency of using Link layer information can be studied. The system will control the MIPv6 handovers and the used implementation of MIPv6 is Mobile IPv6 for Linux (MIPL) [15]. The handover time itself depends on the mechanisms implemented in MIPL. The real efficiency of the introduced mechanism will be learnt during the test phases of the full implementation. The load balancing between the interfaces and the simultaneous use of the interfaces are the future topics of our research.

References

[1] Alper Yegin (Editor), Eric Njedjou, Siva Veerepalli, Nicolas Montavont, Thomas Noel, "Link-layer Triggers and Hints for Detecting Network Attachments", IETF draft, work in progress, 2004.

[2] BlueZ, "Official Linux Bluetooth protocol stack", Sofware, http//www.bluez.org/, 2004.

[3] C. Castelluccia, "HMIPv6: A Hierarchical Mobile IPv6 Proposal" ACM Mobile Computing and Communications Review, vol.4, no.1, January 2000.

[4] H. Soliman, C. Catelluccia, K. El Malki, L. Bellier, "Hierarchical Mobile IPv6 mobility management (HMIPv6), IETF draft, June 2004.

[5] C. Perkins and D. Johnson, "Mobility support in IPv6", In Proceedings of the Second Annual International Conference on Mobile Computing and Networking (MobiCom'96), Rye, New York, USA, November 1996.

[6] D. Johnson, C. Perkins and J. Arkko, "Mobility Support in IPv6", IETF RFC 3775, June 2003.

[7] I. Vivaldi, B.M. Ali, H. Habaebi, V. Prakash, A. Sali, "Routing scheme for macro mobility handover in hierarchical mobile IPv6 network", Proceedings of the 4th National Conference on Telecommunication Technology 2003 (NCTT 2003), 14-15 Jan. 2003.

[8] Jim Geier, "Minimizing Bluetooth Interference", Wi-FI Planet tutorial, July 2002.

[9] J. Manner ed., M. Kojo Ed., "Mobility Related Terminology", IETF RFC 3753, June 2004

[10] MIPv6 Signaling and Handoff Optimization (mipshop), IETF's working group, 2004.

[11] Mäkelä J., Fekete G., Narikka J., Hämäläinen T., Virkki A-M. "Soft handover and routing mechanisms for mobile devices", Proceedings of The 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, September 2004.

[12] R. Koodli, "Fast Handovers for Mobile IPv6", IETF draft, July 2004

[13] S. Aust, D. Proetel, N. A. Fikouras, C. Pampu, C. Görg, "Policy based Mobile IP Handoff Decision (POLIMAND) using Generic Link Layer Information", the 5th IEEE International Conference on Mobile and Wireless Communication Networks (MWCN 2003), Singapore, October 2003.

[14] S. Fu and M. Atiquzzaman, "TraSH: A Transport Layer Seamless Handover Scheme for Space Networks", Nasa's earth science technology conference, 2004.

[15] MIPL, "Mobile IPv6 for Linux", Sofware http://www.mobile-ipv6.org/, July 2004.

[16] Vidales, P., Chakravorty, R., Policroniades, C., "PROTON: a policy-based solution for future 4G devices", Policies for Distributed Systems and Networks, 2004. POLICY 2004. Proceedings. Fifth IEEE International Workshop on, 7-9 June 2004.

PIII

USING LINK LAYER INFORMATION FOR IMPROVING VERTICAL HANDOVERS

by

J. Puttonen, G. Fekete, J. Mäkelä, T. Hämäläinen and J. Narikka 2005

Proceedings of The 16th International Symposium on Personal, Indoor and
Mobile Radio Communications (PIMRC 2005)

Reproduced with kind permission of IEEE.

2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications

Using Link Layer Information for Improving Vertical Handovers

Jani Puttonen*, Gabor Fekete*, Jukka Mäkelä*,
Timo Hämäläinen* and Jorma Narikka†

*University of Jyväskylä, Department of Mathematical Information Technology,
PO.Box 35, 40014 University of Jyväskylä, Finland,
Emails: janput@cc.jyu.fi, gfekete@cc.jyu.fi, jukka.makela@vtt.fi, timoh@cc.jyu.fi

† Jyväskylä Polytechnic, School of Information Technology,
Piippukatu 2, 40100 Jyväskylä, Finland,
Email: jorma.narikka@jypoly.fi

***Abstract*— In this paper we present a common triggering mechanism (cross-layer framework) and logic to gather link state and quality related information from the link layers of different access technologies. This information may be utilized by upper layer protocols and applications to react to changes of the link layer. Examples of such a utilization is movement detection, handover decision and application adaptation. The access technology dependent events and parameters, when necessary, are converted into access technology independent triggers and hints. The presented prototype architecture, Link Information Provider (LIP), is currently tested with the VERHO vertical handover controller, which enables intelligent policy-based handover decisions according to several input parameters from the user, application, link layer, etc. Currently, LIP supports certain general link layer parameters, which can be gathered from the operating system and network interface drivers. The system is designed to be easily extended.**

***Index Terms*— Link layer, Trigger, Hint, Mobile IPv6, Vertical Handover, Always Best Connected**

I. INTRODUCTION

In the last few years, the number of mobile devices as well as access technologies have increased. More importantly, mobile devices are expected to have several integrated access technologies, thus the device is called a multiaccess terminal. Different access technologies have different characteristics related to Quality-of-Service, bandwidth, coverage area, cost, power consumption, etc. [1]. These access networks co-exist (and overlap) and constitute a wireless overlay, a.k.a. heterogeneous network [2]. The access technologies might also provide their specific link layer handover mechanisms. But, for the mobile terminal to be always globally accessible, some upper layer mobility management technique is necessary, such as Mobile IP (MIP) [3], [4]. This is emphasized as the IP protocol seems to be the enabling technology for both applications and access networks [5]. These issues enable the user to be always best connected (ABC) to the services [6].

Mobile IPv6 [4] (MIPv6) is a standard way for IPv6 mobility management for future wireless networks. With the usage of two separate IP addresses, Home Address (HoA) and Care-of-Address (CoA), Home Agent (HA) and IP-in-IP tunneling MIPv6 enables unbroken sessions to applications. However, it does not consider movement detection to support seamless (i.e. no packet loss) handovers, partly because it is purely based on network layer information, such as Router Advertisements (RAs) and Neighbor Unreachability Detection (NUD). The handover delays with MIPv6 in IEEE 802.11 (WLAN) environment can be about 300 milliseconds, depending on the RA interval [7]. Access technology dependent information, such as connection establishment with an access point (AP) and current signal strength, could be utilized to fasten the movement detection process [8]. The information can also enable proactive handovers and provide quality and link feature (e.g. bandwidth, security) related information to be used in handover decision. This is emphasized if we consider interface selection in vertical (i.e. inter-technology) handovers.

There are also related research in progress both at IETF and IEEE. The IETF Detecting Network Attachments (DNA) working group [9] has been working with mechanisms for hosts to detect their IP layer configuration and connectivity status quickly and reliably. Some optimizations were proposed to the current specifications that would allow a host to reconfigure its IPv6 layer faster, such as link layer indications and fast RAs. The IEEE 802.21 working group [10] works on Media Independent Handover (MIH) specifying information sources to enable and assist in handovers and interoperability between IEEE and non-IEEE access technology networks.

In this paper we present *Link Information Provider* (LIP), first presented in [11], to acquire the link layer information from the interfaces of different technologies. Only link layer triggers on the Mobile Node's (MN) end are considered. LIP receives information from link layer events, polls for other related parameters and provides them to upper layer consumers supplemented with trigger and hint indications. This information can be used by different applications and

---

The work of Jani Puttonen is supported by the Nokia Foundation and the Research and Education Foundation of TeliaSonera.

Jukka Mäkelä is now with VTT Technical Research Centre of Finland, PO.Box 1100, 90571 Oulu, Finland

978-3-8007-2909-8/05/$20.00 ©2005 IEEE

TABLE I. AVAILABLE EVENTS FROM LINK LAYER

L2 Technology	EVENT
IEEE 802.11	AP in range
	AP out of range
	Connected
	Disconnected
	Scan Completed
	TX drop
Bluetooth	Connected
	Disconnected

purposes, for example helping in vertical handover decision (i.e. interface selection) or application content adaptation. We have combined the LIP with VERHO [12] VERtical HandOver controller, which is currently under development. This paper expands our previous research related to mobility management techniques [13] and faster handovers for Mobile IPv6 [14].

The rest of the paper is organized as follows. Section II discusses the link layer information in general. Section III describes the architecture and functionality of the Link Information Provider (LIP). In section IV a proof of concept and other possible applications of LIP are discussed. Section V concludes the paper and presents some future work.

II. LINK LAYER INFORMATION

Different access technologies, such as IEEE 802.11 (WLAN) [15], provide link layer handovers, i.e. handover between the Access Points (APs) of the same technology in question. IEEE 802.11 specifies the active and passive AP scanning processes, which provide information about the available APs in the neighborhood. The actual connection with the APs is handled with a two-way association process. Bluetooth^{TM} [16] does not specify any handover mechanisms, although it provides some tools to perform it. By using Bluetooth in the Personal Area Network (PAN) mode and Service Discovery Protocol (SDP) we can implement the Bluetooth handover by ourselves. IEEE 802.3 (later Ethernet) is in general more simple, because a handover consists of pugging in and out the cable. But still there might be problems with switches, because of their Spanning Tree algorithms. Also cellular networks, such as Universal Mobile Telecommunications System (UMTS) and General Packet Radio Service (GPRS), provide IP based communication over them, but due to testing limitations, those remain as future work at this point.

The information that can be gathered from the link layer can be divided into two categories; events and parameters. Event provides information about what happens at the link layer and parameters about the quality and features of the link. Tab. I presents events provided by the IEEE 802.11b and Bluetooth Linux device drivers. The events provided by different IEEE 802.11 Linux device drivers vary a lot (we used a modified version of orinoco-0.15rc2). Also, we are aware of the interoperability problems of Bluetooth and IEEE 802.11 due to same frequency use. But, these problems are out of scope of our research and assumed to be diminished by Bluetooth v1.2 [17].

TABLE II. AN EXAMPLE OF PARAMETER DIVISION

Parameter type	Parameter	Example
Static	IFId	eth0, bnep0, wifi0
	IFType	Ethernet, BT, WLAN
	IFMacAddr	00:60:1D:23:39:C3
Configurable	IFStatus	IFUp, IFDown
	IFIpAddr	3ffe:1::3/64
	SSID	MyNetworkName
	BSSID	00:E0:03:05:3A:CD
	Channel	1-13
	APIpAddr	3ffe:1::2/64
	GWAddr	3ffe:1::1/64
Quality	SigStr	40dB
	PowLevel	
	BitRate	11 Mbps
	Security	WEP,WPA
	AvBandw	6 Mbps
	Price	1 euro/MB

The parameters, which can be gathered from the link layer can be divided into static, configurable and quality related. Static parameters are unchanged variables usually related to physical interfaces, such as interface identification or MAC address. Configurable parameters are usually related to the current link or IP level configurations. Parameters related to quality can be such as signal strength or whatever QoS parameters or features the link offers. Tab. II presents some examples of the parameters that can be bound to a link.

The available bandwidth in IEEE 802.11 is dependent of the number of wireless stations associated to the AP and their data flow amount. So, it is impossible for the station to acquire this information without changes to the infrastructure. Price of using an AP is also quite complicated issue, because of the problem how the MN could get the price information in a dynamic way (i.e. not predefined price/WISP). In [18] it is presented that the price and available bandwidth information could be added to WLAN beacon messages. Also the IP subnet information of the AR could be added to the beacon messages. The Reference [19] presents that the available bandwidth could be evaluated from the IEEE 802.11 MAC layer by listening and collecting the Network Allocation Vector (NAV).

However, different access technologies might provide different events and parameters in different form, or even not every interface will provide all these parameters. For example Ethernet does not have any signal strength value. Thus, it is necessary to unify these parameters for the upper layer, if the parameters are presented in different scale, format, etc. In this context the unified values for events and parameters are triggers and hints. Triggers and hints can be considered to be functional in MN, AP and AR side, but we are focusing primarily on the MN.

Trigger is defined as unified information that presents the events that happen in the link layer, such as association or disassociation to an AP. Hint is more quality related information that can provide certain extra information (e.g. signal strength) to upper layer to be able to anticipate shortly occurring events. The IETF DNA working group [9] has specified two triggers: *Link Up* and *Link Down*. Link Up

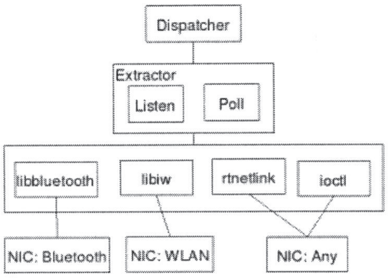


Fig. 1. The Link Information Provider

signifies a state change associated with the interface becoming capable of communicating data packets (e.g. IP packets). Link Down on the other hand signifies a state change assiciated with the interface no longer being capable of communication data packets.

Another question is how the triggers are created from the link dependent events. In [20] the authors, a part of DNA working group, have discussed how the triggers can be formed from GPRS, CDMA2000 and IEEE 802.11 link layers. In GPRS they can be formed from successful activation/deactivation of a PDP Context, in CDMA2000 IPV6CP opened/closed state and in IEEE 802.11 a successfull association/disassociation with an AP. For Ethernet the triggers are now generated immediately after plugin/plugout, but the Spanning Tree Protocol (STP) will cause after this about 30 seconds delay, thus the Link Up should be generated only after the STP is finished.

Hints can be defined as pure parameter values passed on the application (signal strength = 2) or they can be a specified hints according to the parameters (e.g. Link Coming Up, Link Going Down, Link Available).

III. LINK INFORMATION PROVIDER DESIGN

Link Information Provider (LIP) is responsible for extracting and collecting link layer information from the terminal interfaces. Its architecture is depicted in Fig. 1 and it was built to support multiple access technologies. Each access technology has its own module that extracts layer information specific to the technology in question.

LIP is implemented in a Linux environment and currently supports Ethernet, IEEE 802.11 and Bluetooth interfaces. For IEEE 802.11, Wireless Extensions and the *libiw27* library are used. Unfortunately, not all the changes to IEEE 802.11 links are reported through Wireless Extensions as events, so besides listening to events polling is also used. For Bluetooth, the BlueZ protocol stack and the *libbluetooth1* library are used. The change in Bluetooth connections and link information must be collected by polling. Access technology independent information is gathered from the operating system by using Netlink sockets.

The system has two main components, a *Poller* and a *Listener*, these two form the *Extractor*. The Poller is responsible for gathering information at specified intervals and is also used by the Listener. The Listener, as its name suggests, listens to changes (events) occurring to links. These events are reported by the operating system. On each event the Listener calls the Poller. This way the information provided by the two components complement each other.

Two classes of information sources are used, general and technology specific. There is some common information that is provided by all the access technologies but are represented in different formats. In this case, the information is converted to a unified representation. The result of the activity of the Extractor is an information bundle that is fed into a queue for further consumption. This queue is heterogeneous with regards to interface types (technology), that is, it holds information about all the supported interfaces. LIP, by default, generates an event (an item in the queue) whenever a change occurs to any of the supported link information/parameters. That is, an event represents changes in link layer information. One single event may describe changes to more than one link parameter that was detected in the same loop of the extractor. For example, a IEEE 802.11 SNR change could mean also a Link Going Down hint. In such a case, the item holds both the information about the SNR change and the link state change. LIP can also be asked to provide all the current information on links (useful e.g. on application startup).

The queue is handled by the *Dispatcher*. Clients wanting to receive these events can register to LIP. The Dispatcher passes events from the queue to all the registered clients. Clients may be any kind of software that need link layer information (e.g. upper layer protocols, applications). LIP provides various link information and parameters. Some of them are technology specific while others are common to all. The repertoire of supported information is planned to be extended and in the following paragraphs we describe the currently available ones.

Bluetooth specific information include the link quality, transmission power level (dBm) and the BD address (i.e. MAC address) of the peer (access point). The information gathered from IEEE 802.11 interfaces currently include the ESSID (i.e. network name), WEP encryption key, BSSID (remote AP MAC address), bitrate, transmission power level, signal to noise ration (SNR). Additional events received are the completion of scanning, packet dropped on transmission, AP connected/disconnected/in_range/out_of_range, although these are not provided by all of the IEEE 802.11 drivers. The available information got from Ethernet interfaces is the ones that are common to all the access technologies, such as triggers.

The common link information is the following. The transmission power level is not available for Ethernet in LIP, therefore its value is set to a predefined one. For IEEE 802.11 and Bluetooth it is available. The same applies for the bitrate. A signal strength values is calculated from the link quality for Bluetooth and from the SNR for IEEE 802.11. The local and remote (AP) MAC addresses are also stored

TABLE III. UNIFIED SIGNAL LEVEL HINTS

	SNR (WLAN)	Link Quality (BT)	Signal Strength (Hint)	Description
a	0	0	0	No connection
	1-6	1-200	1	Very weak
b	5-12	195-215	2	Weak
	11-16	210-230	3	Good
	15-max	225-255	4	Very good

for Bluetooth and IEEE 802.11, but for Ethernet the remote MAC address is not available. The conversion method for the different technologies is shown in Tab. III. The signal strength is an integer value between [0-4] range. The definition is based on our performed performance tests with Bluetooth and IEEE 802.11b access networks. We performed two types of tests with both access networks and with different link quality values: 1. downloading a file with SCP from the AR and 2. pinging the AR. Only one client was connected to the AP (either IEEE 802.11b or Bluetooth), so it had all the available radio capacity. In IEEE 802.11b tests we used WiFi certified Orinoco Silver PCMCIA IEEE 802.11b card and Orinoco AP-500. For Bluetooth, we used two TDK USB Bluetooth v1.1 dongles, from which one was functioning as PAN User (PANU) and the other PAN Network AP (NAP). For more about the performance test results, refer to Appendix I and Appendix II. The unification thresholds can be changed according to the needs of the consumer. For Ethernet, the signal strength parameter can be set to 4 as default. We have also specified hysteresis thresholds to be able to reduce the possibility of ping-pong effects between APs. The resulting signal strength value is used to provide a hint about getting out of or in reach of APs. The transition between a and b in Tab. III is defined to be the threshold between *Link Going Down* and *Link Coming Up* hints. For example, when the unified signal strength is 2-4, the connection is considered usable. But, when the signal strength reaches value of 2, the consumer of the information may take the necessary actions that it sees proper (e.g start scanning for other available APs).

There are also few hints and triggers provided by LIP. Link Down and Link Up triggers are based on information got from Netlink. That is, when Netlink reports that the link is available (a new link appeared) then LIP generates a Link Up event. Link Down is generated upon receiving a link deletion report from Netlink. The Link Going Down and Link Coming Up hints are based on the calculated common signal strength value as described above. We define also a *Link Available* hint, which provides information of all of the available and usable APs. This can be utilized by an AP scanner module for example.

For testing purposes LIP has a simulator module that is used to feed information into the two event queue manually. The simulator reads a sequence of commands from a named pipe. Commands are available to generate link information for each supported type of interface and to control the simulator.

IV. APPLICABILITY OF LINK LAYER INFORMATION

There are several possible applications that can benefit from link layer information. The most important ones could be more intelligent handover decisions, faster handover processes and link layer adaptive applications, such as connection oriented transport protocols (e.g. TCP) and multimedia streaming applications (e.g. video, voice). The general idea is to adapt in different protocol layers to the constantly changing mobile environment.

In mobility management context, LIP can be beneficial in both horizontal and vertical handovers. LIP allows a host to react to possible link changes faster than utilizing only information provided by the network layer (RAs, NUD). Using Link Layer Triggers it is possible to speed up the movement detection of Mobile IPv6 by immediately sending a Router Solicitation after a Link Up trigger. For more efficient horizontal handover performance the Fast Handovers for Mobile IPv6 [21] or Hierarchical Mobile IPv6 [22] can be utilized.

For interface selection (i.e. vertical handovers) LIP can provide, in addition to link layer connection status and quality, some feature related information (e.g. security) in order to make more intelligent handover decision. Also, proactive vertical handover decisions provided by hints can assist in seamless operations. By using the signal strength as a source for Link Going Down and Link Coming Up hints, vertical handovers can be started before the links really go down (i.e. proactive soft handover). Because of fluctuating variables (e.g. signal strength) hysteresis, timer or moving average may be used to avoid avoid ping-pong movements.

Both transport protocols and different applications can benefit from the link layer information to adapt to the link quality. [23] presents TCP enhancement using link layer information. Also Session Initiation Protocol (SIP) clients could adapt their streams according to the current link.

A proof of concept implementation of LIP is tested with VERHO (Vertical Handover Controller) [12]. VERHO is a system to control the usage of available network interfaces in a mobile host. It receives information about links from LIP and uses a policy based system to make its decisions. Currently VERHO is implemented to control MIPL2, a Mobile IPv6 implementation for Linux. System, user and application policies are considered to control the availability of interfaces (enabled/disabled). Enabled interfaces are then ordered in a preference list. The interface with the highest preference is then selected for use. VERHO and LIP together enable policy-based vertical handovers in heterogeneous environments between network interfaces using Mobile IPv6 (see Fig. 2).

In this paper we present a use case type of analysis of the benefits of LIP in vertical handover context. The interface selection logic is in the VERHO system to which LIP provides the input and Mobile IPv6 handovers are managed by a modified version of MIPL2 RC2. The unifiable values of SNR and signal quality can be adjusted to correspond the current access networks (e.g. the scanning thresholds of IEEE 802.11), as we have done in this test to be able to provide easier testing

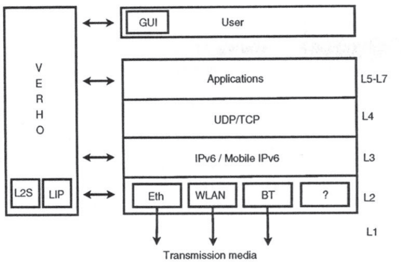


Fig. 2. VERHO Vertical Handover system [12]

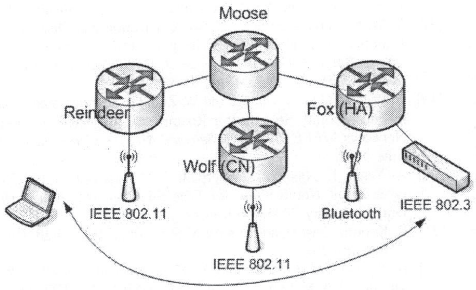


Fig. 3. The test network

possibilities. We rose the unification thresholds to be able to perform handovers with APs close to each other. The topology of the IPv6 test network is presented in Fig. 3. It consists of three ARs and three different access technologies. The AR called Fox functions as the Home Agent (only the interface with Ethernet access). The RA interval was set to 0.05 and 3 seconds. No Route Optimisation was used.

The use case scenario was created on the basis of the network, which is illustrated in Fig. 4, based on the topology in Fig. 3. An employee, who has a multiaccess device with Ethernet, IEEE 802.11b and Bluetooth interfaces, leaves home in the morning and comes back home at the end of the working day. On the way to the workplace there is a IEEE 802.11b hotspot network (ARs Reindeer and Wolf). In the office building there is a Bluetooth (AR Fox) coverage and at the working desk the employee prefers wired Ethernet (AR Fox). The connection was checked by continuously pinging the Home Address of the MN from AR Wolf.

On the way to workplace, the user has only IEEE 802.11b coverage, whose SNR varies around 47. During the horizontal handover between IEEE 802.11b APs (simulated with ESSID change), the MN noticed a BSSID invalidation and Link Down trigger (with SNR 0) followed by a Link Up trigger from the new AP. This triggered the CoA creation and Mobile IPv6 handover. It took around 3 seconds to receive the RAs and to form the new CoA on the new WLAN link. The for the MIPv6 signaling was around 0.3 seconds.

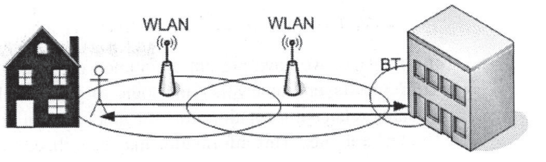


Fig. 4. The use case scenario

When the employee arrives at work with Bluetooth coverage (Link Up from Bluetooth interface), the public IEEE 802.11b SNR is decreasing below the predefined limit (ss < 3) causing a Link Going Down hint. This triggers the Mobile IPv6 handover to Bluetooth. RA based movement detection time was zero because on hint reception the MN has already got RAs on the BT interface. The MIPv6 signaling took around 0.18 seconds.

The wired Ethernet provides more bandwidth (100Mb), thus the user wants to use that where possible. At working desk the user plugs in the cable causing immediately a Link Up trigger at Ethernet interface. The Ethernet segment is built by using a switch and due the Spanning Tree Protocol (STP) the delay between the Link Up trigger and the first RA (link really usable), is a little less than 30s. Thus the Link Up trigger should be formed only after STP is finished. Then an automatic handover happened to Ethernet interface, because VERHO determines a better preference value for it. RA and CoA formation took around 3 seconds and the MIPv6 signaling around 0.01 seconds.

After working day the employee unplugs the Ethernet cable, causing a Link Down trigger to the Ethernet interface and handover to Bluetooth, which is the only usable interface. The MIPv6 signaling took 2s, two BUs were sent, no reply was received for the 1st. Immediately arriving to the IEEE 802.11b coverage, the MN notices a Link Up trigger and performs automatic handover to it because of preference value calculation of VERHO. The time for RA and CoA formation was around 2s and the MIPv6 signaling took around 0.18s. The horizontal handover between IEEE 802.11b APs happened in the similar way as before when the user was going to work.

By using information provided by LIP, it is possible for the user to be always connected to the best interface. And this is achieved with user transparent way, thus the user needs no interaction with the handover processes (except for some user profile at some point). When the employee is arriving to work and plugs in the Ethernet cable, LIP notifies VERHO about 100Mb connection with the best possible signal strength (a set value). This provides better preference value than Bluetooth, and thus a automatic handover to Ethernet. So, the most important benefit of LIP is to provide the link layer related information, and then together with VERHO they can provide the best access. When LIP is extended with parameters such as available bandwidth and VERHO with more intelligent policies, the decision would be even better.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have present a common triggering mechanism (cross-layer framework) and logic to gather link state and quality related information from the link layers of different access technologies. This information may be utilized by upper layer protocols and applications to react to changes of the link layer. Examples of such a utilization is movement detection, handover decision and application adaptation. The presented prototype architecture, Link Information Provider (LIP), was tested with the VERHO vertical handover controller with a use case scenario. The Link Layer information was found to be the enabler of policy based handovers in the context of VERHO.

The further development of LIP includes adding the GPRS/UMTS support. That enables the connections to the WWAN network as well. The current VERHO implementation is somewhat a skeleton one taking only into consideration the LIP inputs. The user and application policies with interface selection algorithms are under research at the time of writing. Also, we are researching scanning possibilities to able to acquire the information about the neighbouring APs. This could provide improved horizontal handover decisions as well. In this paper we have demonstrated the benefits of Link Layer Information considering vertical handover decision. The purpose is also to demonstrate the benefits in providing adaptability to streaming applications.

ACKNOWLEDGMENTS

The authors would like to thank Mr. Michal Rysiak and Mr. Ismo Keränen for their work related to LIP implementation and testing.

REFERENCES

[1] M. Frodigh, S. Parkvall, C. Roobol, P. Johansson, and P. Larsson, "Future-Generation Wireless Networks," *IEEE Personal Communications*, vol. 8, no. 5, pp. 10–17, 2001.

[2] M. Stemm and R. H. Katz, "Vertical Handoffs in Wireless Overlay Networks," *Mobile Networks and Applications*, vol. 3, no. 4, pp. 335–350, 1998.

[3] C. Perkins, "IP Mobility Support for IPv4," IETF RFC 3220, January 2002.

[4] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," IETF RFC 3775, June 2004.

[5] R. Berezdivin, R. Brenig, and R. Topp, "Next Generation Wireless Communications Concepts and Technologies," *IEEE Communications Magazine*, vol. 40, no. 3, pp. 49–55, 2002.

[6] E. Gustafsson and A. Jonsson, "Always Best Connected," *IEEE Wireless Communications*, vol. 10, pp. 49–55, 2003.

[7] G. Daley, B. Pentland, and R. Nelson, "Effects of Fast Router Advertisement on Mobile IPv6 Handovers," in *Proceedings of the Eighth IEEE International Symposium on Computers and Communication 2003 (ISCC 2003)*, 2003, pp. 557–562.

[8] N. Montavont and T. Noel, "Handover Management for Mobile Nodes in IPv6 Networks," *IEEE Communications Magazine*, vol. 40, pp. 38–43, August 2002.

[9] "Detecting Network Attachments," IETF working group, 2004, work in progress.

[10] "Media Independent Handover," IEEE Standard 802.21, 2004, work in progress.

[11] M. Rysiak, "Link Layer Triggers and Hints for Achieving Seamless Handovers," Bachelor's Thesis, Jyväskylä Polytechnic, 2004.

[12] J. Mäkelä, T. Hämäläinen, G. Fekete, and J. Narikka, "Intelligent Vertical Handover System for Mobile Clients," in *Proceedings of the 3rd International Conference on Emerging Telecommunications Technologies and Applications (ICETA 2004)*, September 2004, pp. 151–155.

[13] J. Mäkelä, G. Fekete, J. Narikka, T. Hämäläinen, and A.-M. Virkki, "Soft Handover and Routing Mechanisms for Mobile Devices," in *Proceedings of the 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2004)*, September 2004.

[14] J. Puttonen, A. Viinikainen, M. Sulander, and T. Hämäläinen, "Performance Evaluation of the Flow-based Fast Handover Method for Mobile IPv6 Network," in *Proceedings of the 60th Semiannual IEEE Vehicular Technology Conference (VTC'F04)*, September 2004.

[15] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Standard 802.11, 1999.

[16] "Bluetooth Specification," Bluetooth Special Interest Group SIG, 2004.

[17] D. R.-P. J. Mander and C. Todd, "Evaluating the Adaptive Frequency Hopping Mechanism to Enable Bluetooth - WLAN Coexistence," in *Proceedings of the London Communications Symposium*, 2004.

[18] Y.-W. Lee and S. Miller, "Network Selection and Discovery of Service Information in Public WLAN Hotspots," in *Proceedings of the 2nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, 2004, pp. 81–92.

[19] C. Guo, Z. Guo, Q. Zhang, and W. Zhu, "A Seamless and Proactive End-to-End Mobility Solution for Roaming Across Heterogeneous Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, June 2004.

[20] A. Yegin, E. Njedjou, S. Veerpalli, N. Montavont, and T. Noel, "Link-layer Event Notifications for Detecting Network Attachments," IETF Draft, February 2005, work in progress.

[21] R. Koodli, "Fast Handovers for Mobile IPv6," IETF draft, October 2004, work in progress.

[22] H. Soliman, C. Castelluccia, K. El-Malki, and L. Bellier, "Hierarchial Mobile IPv6 Mobility Management (HMIPv6)," IETF Draft, October 2004, work in progress.

[23] Y. Yu and C. Yin, "A Cross-layer Mechanism for TCP Connection over Wireless Uplink in Cellular Networks," in *Proceedings of the 61st Semiannual Vehicular Technology Conference (VTC'S05)*, May 2005.

APPENDIX I

PERFORMANCE TEST RESULTS FOR IEEE 802.11B WLAN INTERFACE

SNR	Transfer rate (kbps)	Packets lost	Delay (ms)	Wireless events
40	6500	2/19500	1.6	-
35	6500	2/19500	1.6	-
30	6500	3/19500	1.6	-
25	6500	1/19500	1.6	-
20	6500	2/19500	1.6	-
15	6500	3/19500	1.7	-
12	5800	20/17500	2.0	3 Packet Drops
10	3400	26/17000	2.4	-
7	3100	33/15100	5.7	many Packet Drops
4	1450	43/9400	6.8	many Packet Drops

APPENDIX II

PERFORMANCE TEST RESULTS FOR BLUETOOTH INTERFACE

Link quality	Transfer rate (kbps)	Delay (ms)
255	610	29
230	390	41
220	330	42
210	235	-
200	42	107
148	10	-

PIV

PAVING THE WAY FOR FUTURE MOBILITY MECHANISMS: A TESTBED FOR MOBILITY TRIGGERING & MOVING NETWORK SUPPORT

by

J. Mäkelä, R. Agüero, J. Tenhunen, V. Kyllönen, J. Choque 2006

Proceedings of 2nd International IEEE/Create-Net Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (Tridentcom 2006)

Paving the Way for Future Mobility Mechanisms: A Testbed for Triggering & Moving Network Support

Jukka Mäkelä*, Ramón Agüero†, Jari Tenhunen*, Vesa Kyllönen*, Johnny Choque†, Luis Muñoz†
*VTT Technical Research Centre of Finland, PO Box 1100, FI-90571 Oulu, Finland
{jukka.makela, jari.tenhunen, vesa.kyllonen}@vtt.fi
†Department of Communications Engineering, University of Cantabria - Santander, Spain
{ramon, jchoque, luis}@tlmat.unican.es

***Abstract*— Forthcoming wireless communication systems, well represented by the term *Beyond 3G*, are likely to impose some new requirements that go beyond the traditional view on today's networking paradigm. In particular, mobility procedures will no longer be restricted to the change of the *point of attachment to the network*. The work presented in this paper aims at proving, following a fully experimental approach, the feasibility of some architectural components of a *Mobility Control Space*, which has been designed in the context of the *Ambient Networks* project. Especially, in this study we focused on and successfully realized two concepts, a facility for *triggering* mobility events and support for *moving networks*.**

I. INTRODUCTION

The rapid growth of wireless communications devices has paved the way for a significant number of new services and applications. Users are not static anymore, but they are expecting to be connected anytime, anywhere. However, networking solutions are not yet prepared to cope with the potential new characteristics of these forthcoming scenarios.

The Ambient Networks (AN) project [1] aims at providing a common framework to cope with a common interworking architecture, able to embrace heterogeneous networks, different end-user requirements, plug & play support, etc. In particular, mobility aspects are carefully being considered; mobility can no longer be restricted to its current scope, and related mechanisms are likely to go beyond traditional view on mobility as far as not only the devices will be mobile, but also many other entities, such as applications, flows, etc.

Starting from the aforementioned requirement, a Mobility Control Space has been integrated within the overall AN framework. The main aim of this paper is to present a testbed, implemented with *off-the-shelf* technologies, which assess the feasibility of some of the most important components of such architecture. In particular, it focuses on the *Triggering* framework, which can be seen as a core component of the architecture, as it deals with all the potential events that may derive into mobility actions. It also demonstrates the *session handover* concept, which clearly goes beyond the traditional view on mobility. Last, but not least, the testbed provides a framework for recognizing and establishing *moving networks*, which are believed to play a key role in the forthcoming wireless communication scenarios.

The paper is therefore structured as follows. Section II describes the main AN concepts that will be part of the testbed. Section III discusses how they were mapped onto a experimental study scenario, and described the subjacent demonstration flow. Section IV describes the whole platform, including the software components that were implemented. Section V describes the main outcomes of the existing testbed, and propose some topics that could benefit from the existing functionalities within the platform. Finally, Section VI concludes the paper.

II. MOBILITY CONCEPTS IN AMBIENT NETWORKS

A. AN Mobility Overview

AN project has proposed a scalable, flexible architecture, whose main goal is to provide an appropriate abstraction level, so that its functionalities can be used independently of the particular networking operations. Its most important component is the so-called *Ambient Control Space* or ACS, which gathers all the control functions that have been designed in order to fulfill with the identified networking requirements; all these components are structured into the *Functional Areas* (FA), which can entail either protocol-terminated, which need to interact with other peer AN entities, or not.

This paper focuses on a realization of the FA which are related to mobility operations, which form the *Mobility Control Space* (MCS). The MCS high level architecture is shown in Figure 1; as can be seen, it comprises the following components:

- **Triggering.** It collects, processes and filters all types of triggers or/and hints that can impact mobility procedures, and deliver them to the appropriate *customers*.
- **Handover & Locator Management.** It enables the movement of mobile entities between different locations.
- **Rendezvous & Resolution Service.** It ensures that a correspondent node is always able to locate an AN node, updating its *reachability* state.
- **Moving Network Support.** It allows managing a set of nodes that move together as a single entity, taking advantage from this.

The actual state of the mobile object, in terms of how it can be reached, which identifier it is using, which is the next hop towards it, etc, is represented by the boxes on Figure 1. It is worth mentioning that we do not restrict to traditional mobility, but we consider other flavors of it. In this sense, we do not restrict the term mobile object to a device, but we extend it to mobile applications, flows, etc.

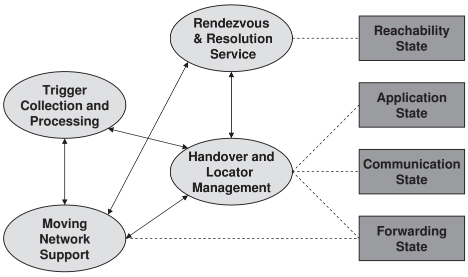


Fig. 1. Mobility Control Space Architecture

This work focuses on two of the aforementioned FAs, the triggering and the moving network support. In the rest of the Section we provide a deeper discussion on these two components. In addition, we will also discuss the different Mobility Dimensions, as used in this work.

B. Mobility Dimensions

Mobility is not anymore just a change of the point of attachment to the network, but also several other dimensions of mobility are possible. In the AN project seven different dimensions for mobility are identified [2]. Figure 2 illustrates examples of these dimensions.

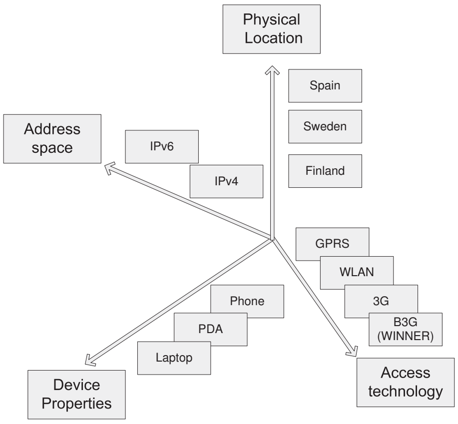


Fig. 2. Examples of New AN Mobility Dimensions

- Physical location: A mobile entity moves between access points within the same radio access technology.
- Access technology: A mobile entity moves from one radio access technology to another.
- Address space: A mobile entity moves between networks/devices with different address space (e.g. IPv4-IPv6)
- Security domain: A mobile entity moves between environments with different level of security.
- Provider domain: A mobile entity moves between networks operated by different provider.
- Device properties: A mobile entity moves from one device to another.
- Time: A mobile entity doesn't move spatially but ongoing communication is suspended for a while and resumed afterwards

In addition to the traditional mobility dimension (physical location), in our study we chose the dimension of device properties as another example. In order to demonstrate it, we implemented a mechanism that can move a video stream session between different user devices.

C. Triggering FA

From the AN perspective the Triggering FA [2] is used for collecting triggering events from different sources like mobility protocols, link-layer information or even external sensors like the SoapBox (Sensing, Operating and Activating Peripheral Box) that was used in this study. The Triggering FA will classify the collected triggers according to definable rules. Classified triggers will be provided to the registered consumers in such way that each consumer will receive only those triggers it is interested in. Figure 3 illustrates the Triggering FA and some possible sources and consumers.

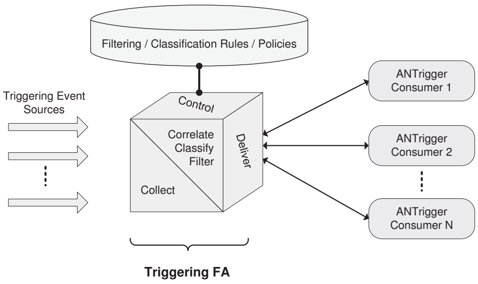


Fig. 3. The Triggering FA and Its Relations to Sources and Consumers

The increasing heterogeneity in current networking environment implies a need for a more generic approach to deal with the collection of pieces of information coming from anywhere in the system. This is the main goal of the Triggering FA; it collects, filters, and delivers the events that may potentially derive into a mobility-related action.

The collected triggers can be used e.g. by different kinds of HO mechanisms to make the decision whether to execute the HO or not. Another consumer of the triggering information in our test prototype is the Moving Network Support FA, described in next subsection.

The Triggering FA consist two main parts [2]: Triggering Event Collection (TEC) and Triggering Events Classification Engine (TECE). Collected triggers can be stored temporarily to the Triggering Event Repository where the registered consumers can fetch the wanted triggers.

D. Moving Network Support FA

In brief, the Moving Network Support (MNS) FA is responsible for the functions of network mobility management. This includes creating and maintaining Routing Groups (RGs) and selecting and maintaining gateways and mobile routers.

In Ambient Networks, a Routing Group [3] can be formed by a group of network nodes that are physically near each other and possibly share similar movement patterns. The RG framework supports multi-hop clusters where nodes may have a special *cluster-head* or *gateway* role.

The simplest benefit of RGs is that they can provide connectivity for nodes that would otherwise be unreachable. The RG structure can, however, enable further optimizations, related to both routing and mobility management [3]. For example, more effective local communication can be achieved or the mobility signaling can be delegated to a designated mobile router.

The algorithm chosen for forming RGs has three phases [4]: *neighbor discovery*, *cluster-head selection* and a *maintenance phase*. Every node running the algorithm is periodically sending *Hello* messages and listening for incoming ones. During neighbor discovery, the nodes identify their stable neighbors within one-hop radius by assigning each neighbor a stability value. The value is updated based on associativity (received *Hello* messages) and link quality (signal-to-noise ratio). Those neighbors whose stability value exceeds a pre-defined threshold are marked as stable.

After discovering these stable links a *cluster-head*, a controlling master-node, is selected. The selection algorithm [4] is a distributed process where each node calculates a suitability value for itself and broadcasts the value to the other stable nodes. Finally, the node with the highest value is chosen as the *cluster-head*. In the maintenance phase, after the RG is formed, it is possible to manage the configuration of the RG by performing operations either on single nodes or whole RGs. That is, single nodes may join or leave an existing RG, and two groups can be merged or one RG can be split into two.

To get external and global connectivity, the RG nodes need to communicate via selected gateway (GW) or mobile router (MR) nodes. To support identification, management and selection of GW and MR nodes, a Gateway Selection Architecture (GSA) [5] has been specified. In addition to ordinary RG, GW and MR nodes, the architecture introduces special Gateway Selector (GWS) nodes. Like *cluster-head* nodes manage RGs, GSW nodes manage the discovery and selection process of the GW and MR nodes.

The gateway discovery process has three phases [5]. First, the GWS node is elected during RG formation. Typically, the chosen *cluster-head* can be assigned also the role of GWS. Next, the potential GW or MR nodes pro-actively advertise their presence and capabilities to the chosen GWS. Finally in the third phase, whenever RG nodes needs the GW service, i.e. communication with nodes outside the RG, they send a request to the GWS, which selects the most suitable GW (or MR) for the requesting node, according to a number of parameters related to topology, physical transmission abilities, and service demands.

III. EXPERIMENTAL STUDY

The basic goal of the study is to assess, using a complete experimental platform, the validity of the concepts that were defined previously. In this sense, our setup can be seen as a proof of the different challenged concepts, rather than a performance evaluation of the subjacent protocols.

In order to build an appropriate platform, the first steps were trying to find a suitable study scenario, so as to mimic the needs that were going to be tested, as well as the different phases that were going to be part of the demonstration flow.

A. Study Scenario

The scenario of the experimental study embraces three different parties:

- *A content provider.* It is a service provider which offers a streaming service to its customers.
- *Mobile users.* Different mobile users that are willing to access the content provided by the previous provider. In addition, they have the capability of using the different AN mobility mechanisms that were described before.
- *Gateway nodes.* These are special nodes that are able to connect the mobile users to an infrastructure networks; furthermore, they are enhanced with the AN mobility procedures.

Furthermore, the scenario will encompass the typical devices that are likely to be part of the forthcoming personal area networks, such as PDA and laptops. In short, the study can be described as follows: some users meet with a friend, a RG is automatically formed, and a GW node, which allows the users to access the content provided by the video server, is selected. One of the users is using both a laptop and a PDA, and using the hint provided by an attached sensor box, the video stream shifts (*session handover*) from one device to the other. Later on, a new potential GW appears. It is offering a better connection, so the RG changes its current GW, without any service disruption. The next section will depict all the steps within this demonstration flow, mapping them onto the concepts that were described before.

In particular, the study will aim at assessing the feasibility of the following challenges:

- Dynamic formation and maintenance of RG.
- Selection and maintenance of GW.
- Dynamic change of GW, without service disruption.
- Demonstration of the Triggering FA, being able to collect, filter (according to predefined policies) and execute the corresponding actions.
- Assess the feasibility of session handover, as a consequence of some of the triggering events.

B. Demonstration Flow

In the following lines, we will describe the demonstration flow that is carried out.

At the beginning, the different nodes (two laptops and a PDA) are switched on. They begin with the distributed interchange of *Hello* messages; from which they are able to assess the associativity (which, to a certain extent, is mapped on the degree of connectivity). This information is then used to create a RG, in which one of the nodes takes the role of *cluster-head*. In addition, one of the devices, which is able to connect to external networks, is selected as GW and therefore, all the other devices use it to access service provider (i.e. the video server); a streaming session is then started from the server to one of the devices. While the video session is ongoing, a new GW appears and joins the RG; according to a set of configurable policies (e.g. price, security, QoS,...), it is offering a better connection than the existing one, so all the devices (RG members) change their GW. The video session that was ongoing, continues through the new GW, without any service disruption, this is to say, the end user does not perceive any halt while watching the video. After a while, the second GW is switched off, and the former one is selected again; this also happens without the user noticing it (i.e. there is no disruption).

Afterwards, the user with the PDA wants to receive the video with it, rather than keep on using the laptop. He moves the PDA to put the screen toward him, and the sensor box attached to it detects this change in device orientation, sending the corresponding event to the Triggering FA, which takes the appropriate steps to perform the session handover. As happened before, there is not any noticeable service disruption in the meanwhile. After a while, the user may decide to continue watching the video with the laptop, so he puts the PDA downwards and the stream is automatically shifted again to the laptop. Furthermore, an additional trigger can be delivered by other means, such as pushing a button in the PDA, which sends the video stream to the other laptop, thus mimicking an action by means of which the user is able to share his video with a friend (another device which is also part of the RG).

IV. TESTBED DEPLOYMENT

A. Hardware

The testbed comprises different types of devices (Fig. 5), as it may be the case in forthcoming personal communications. Laptops (Pentium Mobile) have been used to implement the Video Server, the GW nodes as well as some end user devices. Both Windows and Linux operating systems are used in the platform and all wireless communications is based on 802.11b (*Wi-Fi*) ad hoc mode. In addition, a PDA, the HP iPAQ h5500 was used, using a Linux Familiar distribution.

In the PDA, we had attached a sensor device called SoapBox (Sensing, Operating and Activating Peripheral Box) [6]. It is a small, matchbox sized device packed with processor, wireless and wired communication capabilities and a set of built-in sensors. SoapBox uses commercial components. It has low power consumption, and the RF communication uses an unlicensed radio band (868 MHz). The sensors found in SoapBox are a three-axis acceleration sensor, an illumination sensor for measuring intensity of visible light, an electronic compass for sensing direction and magnetic objects, and an optical proximity sensor for measuring relative distances.

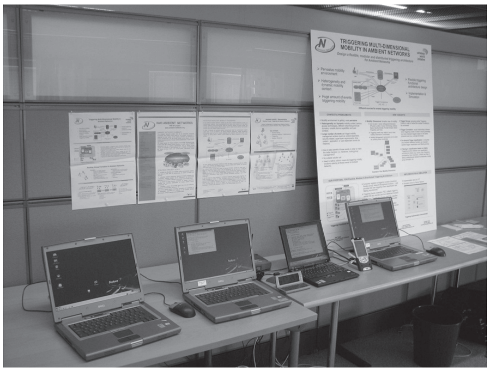


Fig. 5. The Testbed Setup in a Laboratory

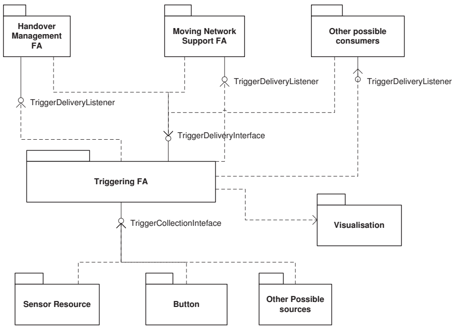
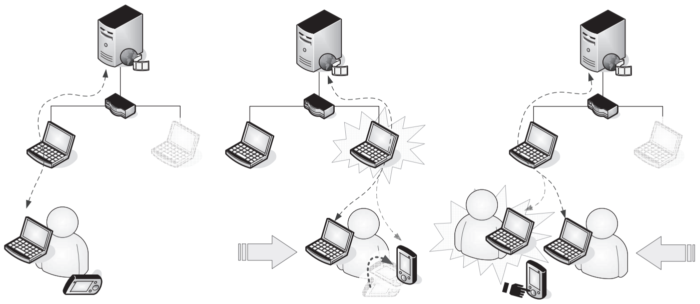


Fig. 6. Overall Software Architecture

B. Implementation Architecture

The primary guidelines for the implementation were extensibility, modularity, cross-platform interoperability, and distributed operation of the components. Therefore, the system was realized using the C++ language, aiming for easy portability by using standard APIs and providing wrappers for any platform-dependent functions.

From the Mobility Control Space, this demonstration required implementing the Triggering FA, parts of the Moving Network Support FA, and the handover functionality. These components and their dependencies are shown in Figure 6.



(1)
The different devices recognize the others and, on a distribute fashion, create a RG

(2)
A new potential GW appears, it joins the RG and all RG members changes the former GW. The video is transferred to the PDA, after the position-based trigger

(3)
A new user joins the RG; the new GW disappears, and the former one becomes GW again. Using press button-based trigger, the video is shared with the new RG member

Fig. 4. Demonstration Flow

C. Triggering FA

In this study the Triggering FA, illustrated in Figure 8, was used to process the triggering information gathered from the sensor, neighbor discovery and user button resources. The processed triggering information was used for making decisions of handover execution. The Triggering FA processed also neighbor discovery information that was used for initiating the RG formation.

The Sensor resource package produces sensor messages for Triggering FA. SoapBoxAPI reads raw sensor data from SoapBox and refines them to sensor messages like ”acceleration X sensor = 1.62“. Sensor Engine has registered these messages from SoapBoxAPI. It collects sensor data and forms higher level context information messages like ”device orientation = downwards“ and sends them to the Triggering Events Collection (TEC) component via Trigger Collection Interface.

TEC gathers triggering events from various sources. It gives time stamps for all received events and sets the time the events are valid in Triggering Events Repository (TER) if not specified.

Classification Engine (CE) registers all events from TEC. CE applies classification rules and policies to incoming events. Classification rules tell how to fill the triggering event attributes, for example ”if the source of the event is ANI, its type will be predicting“. While classification rules ensure correct classification, policies determine whether the events are further processed and apply preferences when several ways of classification could be correct and possible.

TER registers all events from CE. It receives all events entering the Triggering FA except those skipped by policies. TER stores classified events for the time specified for each triggering event.

Handover Decision Engine is a consumer and it registers triggering events from TER with Trigger Delivery Interface according to handover rules. When it has received all required events for a handover, it performs the handover with Handover Tool and Handover Execution components.

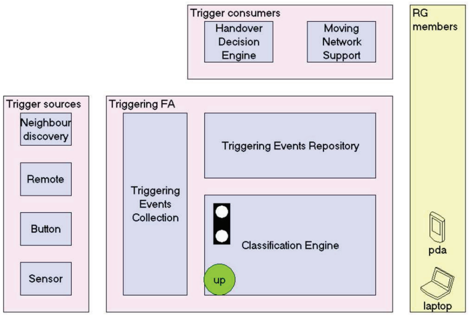


Fig. 7. Screen Capture of the Visualization

The visualization subsystem visualizes the trigger propagation in the Triggering FA. It consists of two subcomponents, a stub and a server. The stub registers events from all components of the Triggering FA so it knows where the triggers are at all times. Every time a trigger moves between any of the monitored components, the stub delivers a corresponding visualization command to the server over a TCP socket connection. Finally, upon receiving these messages the server displays the movements on the screen (Fig. 7). The visualization server utilizes a cross-platform GUI toolkit so it runs on many operating systems, including Linux and Windows.

The components of the Triggering FA are modular, and they communicate by registering triggering events from other components with internal interfaces. Triggering sources and

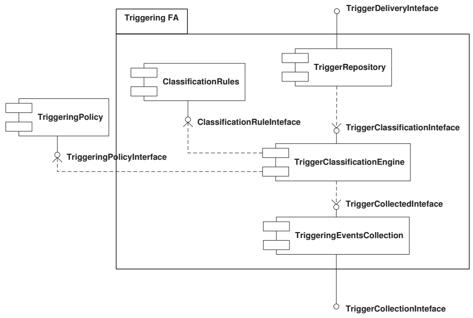


Fig. 8. Triggering FA Software Architecture

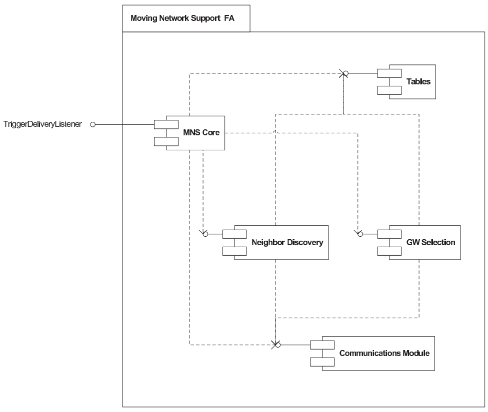


Fig. 9. Components of the Moving Network Support FA

consumers can be distributed to different platforms as they can communicate also with TCP or UDP sockets. Adding new sources and consumers to the system is very easy with this interface.

D. Moving Network Support FA

From the MNS FA we implemented the parts dealing with RG formation and maintenance and GW selection; these are depicted in Figure 9. The FA implementation consists of the core and components rensponsible for communications, storing tables, GW selection, and neighbor discovery.

The very heart of the FA is the neighbor discovery module that monitors the node's one-hop neighborhood and informs other modules about any changes (joining or leaving nodes) by sending triggers through the Triggering FA. The *Hello* messages used to discover the neighborhood and all other advertisements were implemented as broadcast UDP datagrams.

In addition, the *Hello* messages are used to measure the SNR associated to each of the links towards the current node's set of neighbors; in this sense, whenever a *Hello* message is received, the SNR of the link towards the source node is obtained by using the `SIOCGIWSPY ioctl`, which most of existing wireless cards support. This SNR is then mapped onto a quality level, that could be used, e.g. to modulate the associativity values or as another decision factor to be considered while selecting the GW.

The MNS core serves as the interface towards rest of the architecture and is also responsible for starting, stopping, and controlling the other subcomponents. It also performs the message exchange and negotiation needed for RG formation after receiving a trigger from neighbor discovery.

The purpose of the communications module is to provide the other components with a uniform interface for sending and receiving control messages. The communications module adds a common header and inserts a source sequence number to outgoing messages, and also demultiplexes incoming messages to the correct receiving component.

In the MNS FA there is a lot of state information that many components need and can benefit from. To make accessing it easier, the information is stored in one modules and made available to all components via a single interface. Currently, the component stores and shares the tables that have information on all neighbors and RG members.

In the implementation of gateway selection, we chose to not use GWS nodes and used a simpler, distributed algorithm instead. The GW nodes advertise their presence using *Hello*-type broadcast messages and the RG nodes simply choose the one offering the best capabilities by themselves.

This GW selection mechanism also provides elementary support for GW failover. That is, by monitoring the GW advertisement messages of the current GW, the other potential GWs can detect if the current GW disappears and take over the GW role.

E. Session Handover

The movement of a session between different devices is called here a *session handover*. Although sounding similar, it is quite different from the traditional handover (i.e. changing the network access point). In our testbed we were using the currently available techniques for proving this concept. Using the developed mechanism, it is possible to move the end point of the stream between the devices in the established RG in such way that it is totally transparent to the streaming server. The mechanism is based on the idea of using a virtual interface for the stream end point that can be moved and configured to different devices. The target devices should have support for continuing the same session, which in our case was a video stream.

Enabling this type of session movement gives the user a possibility to use the best suitable device for e.g. watching the video stream at all times. For example, at home the user most probably has a possibility to watch the video on a bigger screen and while moving the user can continue watching using a handheld device like a PDA or a smartphone.

V. RESULTS

As mentioned before, the main goal of the presented testbed is to assess the feasibility of the Ambient Networks mobility concepts. The Triggering FA has proven to be an appropriate framework to settle the basis for more advanced mobility mechanisms. It provides an efficient and flexible way of dealing with multiple sources of inputs, and thanks to its easy configurable set of policies, it may be used for other mobility-related procedures.

In addition to this, the testbed comprises a functional method of recognizing a RG. As mentioned before, the RG concept can be used to bring up some optimizations to mobility-related procedures, such as the merge of signaling messages after a handover or to diffuse information to the whole set of RG members on an efficient way.

A. Proof of Concepts

The Triggering framework has been used to initiate mobility-related procedures after detecting inputs coming from different sources. In this work, it has been used to initiate session handover between devices, as well as to initiate the RG formation procedure.

One of the most outstanding issues concerning the mobility within Ambient Networks comes from the fact that we are considering different types of mobility dimensions. The traditional mobility has been shown with the change of GW and, in addition, the session handover is also shown, shifting the video stream for the laptop(s) to the PDA and vice versa. It is important to highlight, concerning session handover, that the end user does not perceive any service disruption.

B. Testbed Extensibility

In addition to have assessed the feasibility (on real platforms) of some advanced mobility concepts, the testbed that has been presented is this paper has been designed and implemented on a modular way, so that it would be quite straightforward to add new functionalities.

First, the Triggering FA has been designed and implemented with the extensibility as a hard requirement. After the architectural design, this component appeared as a central element within the Mobility Control Space, as far as it gathers inputs from various sources and initiates the corresponding mobility procedures. In this sense, it should be able to accommodate new inputs, configure new policies and execute the required procedures. The FA can work either in a centralized way, i.e. all procedures happen within the same device (inputs, mobility actions, etc) or on a distributed fashion, when the triggers are detected by different devices (e.g. an infrastructure node), which send them via the traditional socket interface.

On the other hand, the Moving Network Support FA provides a common framework on which different functionalities can be tested. The basic idea is to manage the whole RG as a single entity, rather than managing all the nodes separately. In this sense, the existing testbed will be extended, using the RG concept to bring up some of the benefits that could be achieved. One example of such benefits may be to use the RG structure so as to propagate some information into the network, and improving some existing mobility protocols (e.g. *Mobile IP* or *Host Identity Protocol*).

VI. CONCLUSION AND FUTURE WORK

In this paper we have addressed the design, development and assessment of a complete testbed that can be used to test some of the potential new features that will pave the way of forthcoming mobility-related challenges. We have discussed why current view on mobility is no longer valid, and we have proposed some new techniques that may help to alleviate the difficulties that will arise. In particular, and using *off-the-shelf* technologies, we have implemented parts of two Functional Areas of a complete Mobility Control Space, namely a common Triggering framework and a set of functionalities to be able to manage Moving Networks on a more efficient way.

The testbed will be extended and improved with new features and functionalities. As has been mentioned, its flexibility and extensibility have been carefully considered since the design phase. Among the aspects that will be added soon to the demonstrator, the integration of a mobility solution based on the *Host Identity Protocol*, support for multi-hop communications within the RG, and distributed triggering protocols are the most appealing ones.

ACKNOWLEDGEMENTS

This work has been carried out in the framework of the Ambient Networks project (IST 507134) that is partially funded by the Commission of the European Union. The views expressed in this paper are solely those of the authors and should not be interpreted as necessarily representing the views of their employers or the Ambient Networks project. The comments and ideas from people involved in the project's work packages one and four are gratefully acknowledged.

REFERENCES

[1] N. Niebert, A. Schieder, H. Abramowicz, G. Malmgren, J. Sachs, U. Horn, C. Prehofer, and H. Karl, "Ambient Networks – An Architecture for Communication Networks Beyond 3G," *IEEE Wireless Communications (Special Issue on 4G Mobile Communications – Towards Open Wireless Architecture)*, vol. 11, no. 2, pp. 14–21, Apr. 2004.

[2] V. Typpö, O. Gonsa, T. Rinta-aho, B. Tharon, and S. Bonjour, "Triggering Multi-dimensional Mobility in Ambient Networks," in *14th IST Mobile & Wireless Communications Summit*, June 19–22, Dresden, Germany, June 2005.

[3] A. Surtees, R. Agüero Calvo, J. Tenhunen, M. Rossi, and D. Hollos, "Routing Group Formation in Ambient Networks," in *14th IST Mobile & Wireless Communications Summit*, June 19–22, Dresden, Germany, June 2005.

[4] J. Tenhunen, V. Typpö, and M. Jurvansuu, "Stability-Based Multi-Hop Clustering Protocol," in *The 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, Sept. 11–14, Berlin, Germany, Sept. 2005, to appear.

[5] M. Eyrich, M. Majanen, E. Perera, R. Toenjes, R. Boreli, and T. Leinmueller, "GSA: An Architecture for Optimising Gateway Selection in Dynamic Routing Groups," in *The 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, Sept. 11–14, Berlin, Germany, Sept. 2005, to appear.

[6] E. Tuulari and A. Ylisaukko-oja, "SoapBox: A Platform for Ubiquitous Computing Research and Applications," in *Lecture Notes in Computer Science 2414: Pervasive Computing*. August 26–28, Zürich, CH: Springer, Aug. 2002, pp. 125–138.

PV

TRIGGER MANAGEMENT MECHANISMS

by

J. Mäkelä and K. Pentikousis 2007

Proceedings of the Second International Symposium on Wireless Pervasive Computing (ISWPC 2007)

Reproduced with kind permission of IEEE.

Trigger Management Mechanisms

Jukka Mäkelä and Kostas Pentikousis
VTT Technical Research Centre of Finland,
Kaitoväylä 1, FI-90571 Oulu, FINLAND
Email:firstname.lastname@vtt.fi

***Abstract*— Current protocol stacks can handle only a handful of events that may trigger actions for mobility management, such as signal strength indicators and cell load. We argue for new mechanisms that can deal with a greater variety of triggering events, which may originate from any component of the node's protocol stack as well as mobility management entities within the network. We present the main concepts that govern our trigger management mechanisms (TMM), which aim at operating in a richer mobility management framework, and enable the deployment of new applications and services. After summarizing the architectural requirements with respect to trigger collection, processing, storage, and dissemination, we introduce a real implementation of TMM on commodity mobile devices. We briefly review our testbed environment and present experimental results using TMM to drive a lossless streaming video session handover between a laptop and a PDA. We position the current TMM design and implementation within the Ambient Networks architecture, centering in particular on the use of policies to steer TMM operation, and the role that TMM can play in an information service infrastructure. Finally, we outline current and future work items.**

I. INTRODUCTION

Mobile devices are now capable of running demanding network applications, may have multiple wireless network interfaces and, thus, connectivity options. For example, most advanced PDAs and smartphones include interfaces for wireless local, personal, and wide area networks (WLAN, WPAN, and WWAN, respectively). Nevertheless, state-of-the-art mobile protocol stacks can only handle a small set of event notifications, typically related to radio access network (RAN) connectivity, spatial mobility, and load balancing. For instance, signal strength deterioration generally leads to a base station handover (HO) in cellular voice; 2G/3G mobile phones typically opt for 3G connectivity when the user moves into a new area; and sustained high data traffic loads may force the Universal Terrestrial Radio Access (UTRA) transport function to reallocate resources (and even perform a HO) in WCDMA 3G/UMTS networks [1].

We argue for a framework to handle a much larger set of notifications caused by events that originate not only from the lower layers of the protocol stack (physical, data link, and network), but from the upper layers (session, transport, and application) as well. We present mechanisms that allow mobile devices to manage both conventional mobility events, such as the availability of a new network access, received signal strength indications (RSSI), network capacity load, and higher-level events, for example, security alerts, policy violations, end-to-end quality of service deterioration, and network access cost changes. The corresponding event sources need to be able to deliver notifications to interested applications and other system entities (for example, the mobility management protocol used, such as Mobile IP or Host Identity Protocol [2]), in a uniform, concise and standardized manner. We refer to these standardized notifications as *triggers* in the remainder.

We introduce the main concepts of trigger management in §II and our architecture in §III. Section IV reviews our implementation on commodity devices, describes our experimental study scenario, and reports results from our testbed. Section V positions our work within the Ambient Networks architecture. We discuss related work in §VI, and conclude the paper in §VII outlining plans for future development.

II. TRIGGER MANAGEMENT CONCEPTS

After surveying the literature (see, for example, [1], [3]–[7]) and based on our own expertise, we identified more than one hundred different types of network events related to mobility management. We cluster triggers, regardless of the underlying communication technology, based on groups of events related to changes in network topology and routing, available access media, radio link conditions, user actions and preferences, context information, operator policies, quality of service (QoS) parameters, network composition [8], and security alerts. Fig. 1 illustrates six different trigger groups as boxes. The "offshoots" on top point to example triggers belonging to each group. The rightmost group includes representative link layer "up/down" triggers (irrespective of the radio access technology). On the other hand, the leftmost group includes triggers originating from the application layer. In this example, certain triggers originate from the node ("System Resources") while others originate from the network ("Macro Mobility").

With respect to mobility management, triggers can be classified and filtered based on five criteria: type, origin, occurrence/frequency, event persistence, and temporal constraints. For example, we identified three trigger types based on whether an event may, will, or must force a HO. Origin corresponds to the entity that produces the trigger, for example, the radio access component. With respect to frequency of occurrence, an event may be either periodic (such as, network measurements) or asynchronous (such as, the availability of a new network access or a security alert). Finally, events can be either transient or persistent, and they may be associated with a real-time constraint. An advantage of our grouping and classification approach is that we can detect the relations

1-4244-0523-8/07/$20.00 ©2007 IEEE

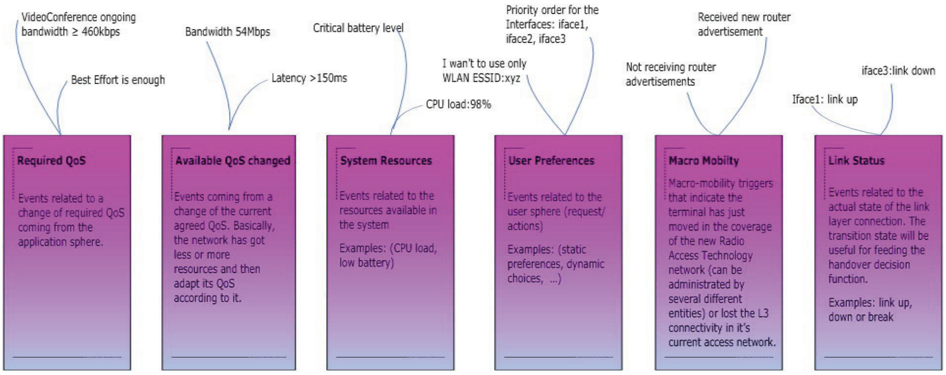


Fig. 1. An example of trigger groups originating from the node (and different layers of the protocol stack) and the network.

between otherwise disparate triggers. This prevents the generation of excessive transient triggers based on, for example, the same root-cause event, such as a link outage, and reduces the number of events that need to be processed. For instance, in Fig. 1, the Link Status trigger "link up", could be advantageous for a mobile node if it starts sending router/agent solicitation messages immediately after receiving the trigger, thus getting the router advertisements earlier than if it passively waited for them. This is especially the case for vertical handovers; see for example [9]. Conversely, the mobile node does not need to wait for router advertisements after a "link down" trigger.

Event sources need to be able to deliver notifications to interested applications and other system entities in a uniform, concise, and standardized manner. This approach simplifies notification handling considerably, while guaranteeing sufficient diversity for event separation and classification. In order to manage and efficiently propagate triggers originating from a variety of sources we developed trigger management mechanisms (TMM) that lay the foundation upon which sophisticated HO operations can be performed. We aim at establishing an extensible framework where new sources of triggers can be included as necessary. It is important to highlight that TMM provide the means to disseminate mobility-related information between one or more event sources and several trigger consumers and that HO decisions are still the responsibility of the mobility management protocol, say, Mobile IP. TMM can also provide hints about moving the communication endpoint from one device to another, as explained in §IV.

III. TMM ARCHITECTURE

The bird's eye-view of the TMM ecosystem is shown in Fig. 2 and comprises (a) sources (or producers) feeding relatively fast-changing information about events; (b) trigger consumers, which receive notifications in the form of standardized triggers about events they are interested in; and (c) the implementation of TMM, which includes data stores and internal logic. The TMM implementation processes the events received from the producers, generating triggers based on consumer-provided (filtering) rules and making sure that all system-wide policies are enforced.

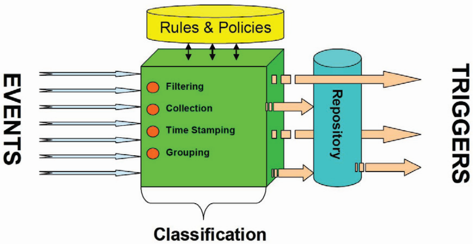


Fig. 2. Schematic of TMM.

A central part of the design is designating entities as producers and consumers of triggers. Consumers must state their need to receive triggers and can choose to stop receiving them anytime. The same entity may be simultaneously acting as both a consumer and a producer. For example, it can receive all triggers originating from RAN events, but opt to receive only the upper-layer triggers associated with security policy violations. In the former case, the consumer takes advantage of the trigger grouping and classification functionality; in the latter, it additionally requests trigger filtering. Consumers can, of course, use these triggers to generate their own and serve as a producer for other entities. We expect that TMM will be used to guide HO decision and execution. In particular, consumers can use triggers to derive whether the mobile device is moving within a single network or it is crossing different access technology boundaries, and whether the addressing scheme, trust and provider domains should be changed accordingly.

The TMM architectural requirements address functional, performance, and security issues. As shown in Fig. 3, the core TMM implementation is partitioned in three major components, namely, trigger event collection (§III-A), processing (§III-B), and storage (§III-C), described next. The figure also depicts examples for TMM event sources (access technology, HIP, MIP, and TCP) and TMM consumers (applications, TCP, MIP, and HIP). As mentioned earlier, the same component may act as a trigger source and consumer at the same time, such as the case of MIP in the figure. In short, events are collected from the corresponding sources and are handed over to the trigger processing engine which is responsible for time-stamping and reformatting triggers (if necessary), and assigning them to the appropriate group. Consumers-specific rules for filtering and the filtering itself are handled during trigger processing. Processed triggers typically have an expiration time, after which they are automatically removed from the active triggers repository. We support the application of different triggering policies (§III-D), defined as a set of classification, filtering, trust, and authorization criteria/rules. This allows our implementation to enforce a different policy at different times or when the node operates in different contexts.

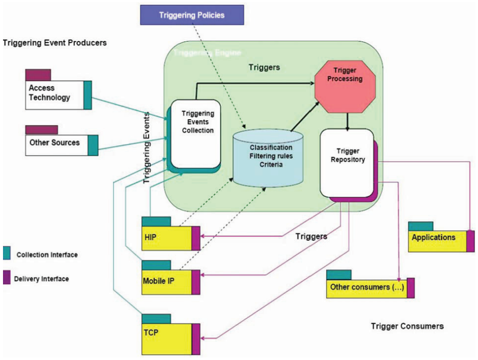


Fig. 3. TMM Components.

A. Triggering Events Collection

Triggering events collection is a function in TMM, which receives events from various sources in the network system via the trigger collection interface. A TMM implementation may contain several event collectors, which may be distributed and which may be responsible for collecting different types of events. The need for different event collectors arises from the fact the origin of an event source can be a hardware device, a system component implemented in kernel space, or an application implemented in user space. For example, each device driver could implement its own event collection functionality, which would be capable of handling triggering events produced by the specific device only. Moreover, sources can also be located in the network such as at active network elements or at the user's home network. Finally, a particular TMM implementation can act as a consumer to another one located in a different node. Thus, orchestrating the collaboration of, perhaps, several collection entities is needed in order to efficiently gather a larger amount of events.

B. Trigger Processing

We want to handle triggers using a common format and reformat any "legacy" ones into the chosen standardized format once they arrive via the triggering event collection. This allows us to tap into existing event sources, which are not yet compatible with our TMM and will be instrumental in migrating "legacy" systems to the new framework. New sources should implement the trigger event collection functionality and use the trigger collection interface in order register their triggers and to make them available to consumers. Consumers can subscribe by specifying a set of triggers (and, optionally, filtering rules) and are expected to unsubscribe when they do not wish to receive them any longer. For each consumer subscription, the TMM processing component makes sure that filters are formatted correctly, may supply default filters for certain consumers, and performs the actual filtering. Basic rules can also be used as building blocks for crafting more sophisticated rules.

C. Trigger Repository

The repository is designed to meet the stringent requirements placed by mobility management, but can be used to store non-mobility triggers. The basic primitives include adding, removing, updating, and disseminating triggers in a standardized format. Each stored trigger has an associated lifetime and is removed automatically once it expires.

D. Policies and Rules in TMM

The availability, on the one hand, of a system-wide policy and, on the other, consumer-supplied filters lies at the center of our TMM design. These two are orthogonal providing flexibility and adaptability. System policies ensure that only designated consumers can receive certain groups or types of triggers. For example, a node may operate under different policies regarding network attachment depending on whether the user is on a business or a leisure trip. Policies can also establish different trigger classification and groupings in different contexts and are typically stored in a separate repository, accessible to the TMM implementation. Filters allow a consumer to focus a trigger subscription. For example, a monitoring application may be interested in receiving all network utilization measurements, while a VoIP application may be interested in a receiving a trigger when utilization exceeds a threshold and the user is in a call. The VoIP application can even be an intermittent trigger consumer, subscribing and unsubscribing to receive triggers as needed.

IV. CONCEPT VALIDATION

We implemented TMM and experimented with it in a real environment. We tested our user-space C++ implementation on

Fig. 4. Triggering in practice. The video is streamed to the laptop for the first three minutes (left), followed by a session handover to the PDA (center), and a handover back to the laptop (right).

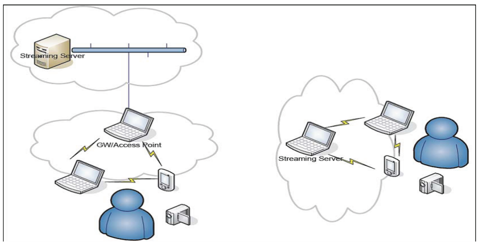


Fig. 5. Topology of the test cases.

laptops running FreeBSD, Linux, and Windows, and on a PDA running Linux. This particular implementation is called TRG and is currently being integrated in a larger prototype for the Ambient Networks project (see §V, below). Our demonstration testbed, detailed in [10], employs TRG to enable streaming video session handovers (VSHO) between two mobile devices. In the scenario of Fig. 4, the user starts watching a video streamed to his laptop. His PDA is nearby and the user decides to move to another room but would like to keep watching the video on the way. The PDA is fitted with a multi-sensor device (detailed in [11]), which was extended to provide standardized orientation triggers (see §III-A). Sensors are becoming more common in everyday life and new smartphones come with several sensors measuring illumination, for example; we expect that these sensors will become more common and are well suited to act as trigger sources. When the user picks up the PDA, a "vertical orientation" trigger initiates a VSHO from the laptop to the PDA; the two have to coordinate and arrange for the transfer of the video streaming session (Fig. 4). A successful VSHO allows the user to receive the streaming video on the PDA over the WLAN seamlessly. The user can also explicitly initiate VSHO by pressing a PDA button. In this example, TRG handles triggers associated with mobility, orientation, and user preferences, keeping the video flowing smoothly while changing the communication end-point.

Fig. 5 illustrates the logical topologies of two different test cases, which were evaluated in our lab, based on the scenario shown in Fig. 4. On the right-hand side of Fig. 5, all devices are connected using a WLAN in ad-hoc mode, as if the user streamed a video from his digital collection at home. On the left hand side, the server is located in a different network, as would be the case when watching a video from a service provider over the Internet. In both setups, we stream a ten-minute video encoded at 576 kb/s over UDP. At $t = 3$ min, a session HO from the PDA to the laptop is triggered, and at $t = 7$ min, the session is "moved" back to the laptop. We qualitatively examined the quality of the video stream and we did not notice any lost frames. Tcpdump traces captured during the experiments confirmed that no packet losses occurred, and that the effect on packet delay is negligible, compared to packet delays before and after the VSHO. It is important to note that the focus of these experiments is not on showcasing the VSHO. This is simply a particular application of triggers leading to a HO. Instead, we emphasize that these experiments aim at assessing the feasibility of introducing a TMM implementation in small handheld devices. We do not attempt to formally draw peak signal to noise ratio (PSNR) curves, but point out that the absence of packet loss in our evaluation scenarios is a good indication about the ability of the current TMM implementation to expeditiously deliver triggers and enable video session HO.

Fig. 6 illustrates the packet interarrival times as recorded by tcpdump using a boxplot [12]. The box width is proportional to the number of packets recorded in each trace. The box represents the middle 50% of the data, the line in the middle represents the median packet interarrival time, and the "hinges" the $Q1$ and $Q3$ quartiles. The width of the notch in the middle corresponds to the 95% confidence interval (CI) of the distribution median. Values outside the whiskers, shown as circles, are considered outliers. The first box on the left shows the server tcpdump trace when the entire video is streamed from beginning to end without a VSHO directly from the server to the laptop (ad-hoc scenario); the second box is the corresponding laptop trace. The third box from the left shows the server in the case when the ten-min video was streamed to the laptop during $t = 0 \ldots 3$ and $t = 7 \ldots 10$ min. During $t = 3 \ldots 7$ min the video was streamed to the PDA. In the handover case, 9 packets captured at the server side (the outliers in the figure) recorded intearrival times exceeding 0.1 s. By inspection of the packet numbers we found out that all 9 packets with intearrival times exceeding 0.1 s were sent when the stream was sent to the PDA and 5 consecutive packets when the HO occurred. These results are very promising, despite the fact that this is a prototype implementation, especially when taking into consideration that the PDA was running the video client and captured packets using tcpdump throughout the experiment living few spare system resources available.

Back of the envelope calculations for the ad-hoc scenario yield packet transmission times, under excellent conditions, in the ms-range. On the other hand, packets are transmitted from the video server with interpacket distances in the order of several dozens of ms. Since we conducted the above experiments, we continued the development and evaluation of TRG. In fact, we have now added a web services trigger collection interface, as we proceed integrating more event sources. Using this new version of TRG, we ran experiments where we submit 100 000 back-to-back triggers to TRG and

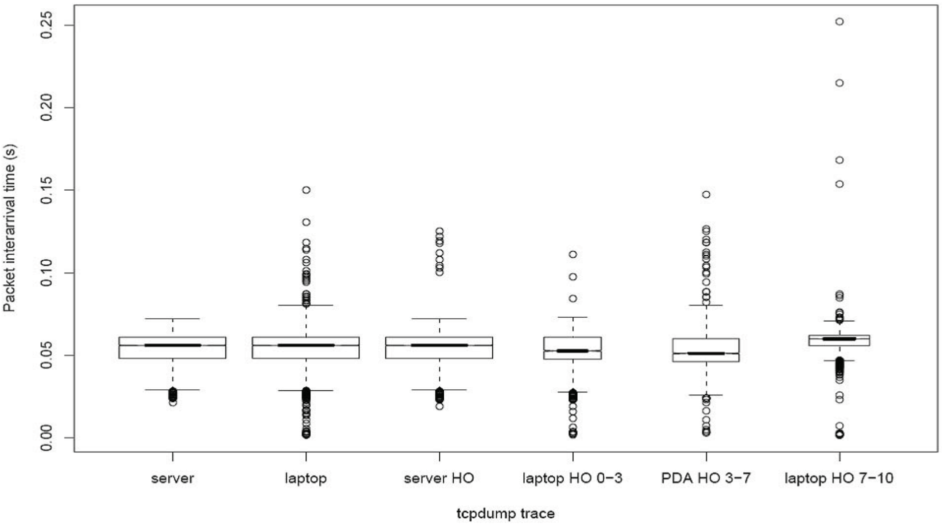


Fig. 6. Experimental results when triggering a VSHO.

found that it takes about 1 ms to collect, process, store, and disseminate a trigger, on the average, which is a good indication that TRG is already "good enough" for several scenarios. More evaluations are on the way, and are part of our ongoing effort.

V. TMM IN AMBIENT NETWORKS

The Ambient Networks project (www.ambient-networks.org) is developing innovative mobile network solutions for increased competition and cooperation in an environment with a multitude of access technologies, network operators, and business actors. The project aims at offering a complete wireless network solution based on dynamic composition of networks that provide access to any network through the instant establishment of inter-network agreements. The concept offers common control functions to a wide range of different applications and access technologies, enabling integrated, scalable, and transparent control of network capabilities. It is difficult to imagine the entire Ambient Networks project succeeding in meeting all its ambitious goals without an ahead-of-the-curve, coherent, and forward-looking mobility solution, given its central role.

The TRG implementation of TMM introduced in the previous section is an important building block of the Ambient Networks Information Service Infrastructure (ANISI) [13]. ANISI is an information service infrastructure designed to provide services and applications at different layers of the protocol stack with support for network information gathering, correlation and intelligent decision-making in support of enhanced mobility management and context-aware communications. The Ambient Networks architecture [14], [15] capitalizes on the availability of a TMM implementation (i.e. TRG) to collect triggering events from sources within an Ambient Network Node (ANN) and the Ambient Network (AN) as well, classifying and filtering them according to dynamically defined classification rules. Within the Ambient Control Space (ACS), TRG disseminates classified triggers to all subscribed consumers in a fashion similar to that presented in §III.

VI. RELATED WORK AND DISCUSSION

Previously published work [5]–[7] shows the benefits of using event information, for example, to proactively perform a handover in order to maintain QoS levels. Our goal is to define a framework that supports the event collection and processing, and trigger distribution possibly from hundreds of different sources. We concur with Vidales et al. [4] that in heterogeneous network environments several sources of events and context information should be consulted in order to achieve seamless connectivity and develop swift mobility management mechanisms. Furthermore, earlier work in other event/notification systems [16]–[19], which introduces mechanisms on how to implement such systems, along with the evaluated event generation cases is very encouraging and complimentary to our effort in defining TMM as a specialized notification system for mobility-related events which originate from the entire protocol stack.

The IEEE 802.21 working group is standardizing an information service that will facilitate media independent handovers. The proposed standard includes an event service, which has common characteristics with our TMM design but does not prescribe a particular implementation and stops short of allowing upper-layer entities to provide events that can drive a HO. Our approach emphasized standardized ways for consumers to receive trigger from a variety of sources. Easy application registration to TRG permits them to get the information they want from different sources. Event generation, on the other hand, is by its very nature a distributed process and, without a central agent, all sources and consumers are forced to create a fully meshed topology. By introducing TRG, event collection becomes straightforward and trigger distribution standardized. Moreover, as part of ANISI, TRG allows system components to manage context and triggers comprehensively.

VII. Conclusion and Future Work

We presented TMM, a framework that can handle a greater variety of triggering events aimed for a richer mobility management network environment, and evaluated a particular implementation. The overall TMM architecture is still under development, and several issues need addressing. We already have a prototype implementation (TRG) and have tested the framework with handhelds in a real network. Our current work involves the refinement of the code in the prototype, and a more detailed empirical study, which will quantify the scalability and performance of the event collection and trigger distribution mechanisms, as well as assessing the value of using rules and policies for event and trigger classification.

Moving forward, we are integrating TRG with HIP and Mobile IP, taking advantage of the fact that it can be located both at the terminal and network side. For example, sources at the network side can monitor the network capacity load and other QoS metrics in overlapping WLAN and WWAN areas. Based on this information, TRG can send triggers to the terminal predicting or even forcing a vertical HO. Our experience so far has shown that implementing this kind of functionality in small devices is feasible. Addressing performance issues (scalability; network, computational and power consumption; and improved user experience, when compared to contemporary solutions), and dealing with security risks (due to malicious trigger consumers and producers), are part of our ongoing work.

Acknowledgement

We thank the anonymous reviewers for their comprehensive comments and thoughtful suggestions. They greatly enhanced this paper. This work has been carried out in the framework of the Ambient Networks project (IST 507134), which is partially funded by the Commission of the European Union. The views expressed in this paper are solely those of the authors and do not necessarily represent the views of their employer, the Ambient Networks project, or the Commission of the European Union. The comments and ideas from people involved in the projects mobility research are gratefully acknowledged.

References

[1] P. Prasad, W. Mohr, and W. Konhuser, *Third Generation Mobile Communication Systems*. Boston: MA: Artech House Publishers, 2005.

[2] T. Henderson, “Host mobility for IP networks: a comparison,” in *IEEE Network*, vol. 17, no. 6, 2003, pp. 18–26.

[3] Jochen Eisl et al, “Mobility Architecture & Framework - D4.2 Core Report,” 2005, IST-2002-507134-AN/WP4/D4.2.

[4] P. Vidales, J. Baliosian, J.Serrat, G. Mapp, F.Stajano, and A.Hopper, “Autonomic system for mobility support in 4g networks,” *IEEE JSAC*, vol. 423, no. 12, pp. 2288–2304.

[5] S. Ishihara, K.Koyama, G.Miyamoto, and M.Kuroda, “Predictive rate control for realtime video streaming with network triggered handover,” in *IEEE WCNC*, vol. 3, no. 13-17, Las Vegas, Nevada, USA, 2005, pp. 1335–1340.

[6] H. Chaouchi and P. Antunes, “Pre-handover signalling for QoS aware mobility management,” in *International Journal of Network Management*, 2005.

[7] E. Casalicchio, V. Cardellini, and S. Tucci, “A layer-2 trigger to improve QoS in content and session-oriented mobile services,” in *Proceedings of the 8th ACM international Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Montral, Quebec, Canada, OCT 2005, pp. 95–102.

[8] C. Kappler, P. Mendes, C. Prehofer, P. Pöyhönen, and D. Zhou, *Autonomic Communication*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005, ch. A Framework for Self-organized Network Composition.

[9] N. Montavont, E. Njedjou, F. Lebeugle, and T. Noel, “Link triggers assisted optimizations for mobile lPv4/v6 vertical handovers,” in *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC 2005)*, 2005, pp. 289 – 294.

[10] J. Makela, R. Aguero, J. Tenhunen, V. Kyllnen, J. Choque, and L. Munoz, “Paving the Way for Future Mobility Mechanisms: A Testbed for Mobility Triggering & Moving Network Support,” in *Proceedings of 2nd International IEEE/Create-Net Tridentcom*, Barcelona, Spain, March 2006.

[11] E. Tuulari and A. Ylisaukko-oja, “SoapBox: A Platform for Ubiquitous Computing Research and Applications,” in *Lecture Notes in Computer Science 2414: Pervasive Computing*. August 26–28, Zürich, CH: Springer, Aug. 2002, pp. 125–138.

[12] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2006, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org

[13] R. Giaffreda, K. Pentikousis, E. Hepworth, R. Agüero, and A. Galis, “An information service infrastructure for Ambient Networks,” in *IASTED 25th International Conference on Parallel and Distributed Computing and Networks (PDCN 2007)*, Innsbruck, Austria, 13-15 February 2007.

[14] N. Niebert, A. Schieder, H. Abramowicz, G. Malmgren, J. Sachs, U. Horn, C. Prehofer, and H. Karl, “Ambient Networks - An Architecture for Communication Networks Beyond 3G,” in *IEEEWireless Communications*, vol. 11, no. 2, Montral, Quebec, Canada, APRIL 2004, pp. 14–22.

[15] N. Niebert, M. Prytz, A. Schieder, N. Papadoglou, L. Eggert, F. Pittmann, and C. Prehofer, “Ambient Networks: A Framework for Future Wireless Internetworking,” *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, vol. 5.

[16] C. H. Lwi, H. Mohanty, and R. K. Ghosh, “Causal Ordering in Event Notification Service Systems for Mobile Users,” in *Int. Conf. Information Technology: Coding and Computing (ITCC)*, 2004.

[17] E. Yoneki and J. Bacon, “Determination of Time and Order for Event-based Middleware in Mobile Peer-to-peer Environments,” in *IEEE Int. Conf. Pervasive Computing and Communications (PerCom) Workshops*, 2005.

[18] R. Meier and V. Cahill, “STEAM: Event-based Middleware for Wireless Ad Hoc Networks,” in *IEEE Int. Conf. Distributed Computing Systems (ICDCS) Workshops*, 2002.

[19] H. A. Duran-Limon, G. S. Blair, A. Friday, T. Sivaharan, and G. Samartzidis, “A Resource and QoS Management Framework for a Real-time Event System in Mobile Ad Hoc Environments,” in *Int.Workshop Object-Oriented Real-Time Dependable Systems (WORDS)*, 2003.

PVI

TOWARDS SEAMLESS MOBILITY WITH CROSS-LAYER TRIGGERING

by

J. Mäkelä 2007

Proceeding of 18th Annual IEEE International symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007)

The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)

TOWARDS SEAMLESS MOBILITY SUPPORT WITH CROSS-LAYER TRIGGERING

Jukka Mäkelä
VTT Technical Research Centre of Finland,
Kaitoväylä 1, FI-90571 Oulu, FINLAND
Email:jukka.makela@vtt.fi

ABSTRACT

This paper presents mechanisms for collecting and processing cross-layer information from mobile devices and networks. That information is further used to feed the handover decisions as a part of the seamless mobility support. This paper presents a mechanism called Trigger Management, developed within the EU's Ambient Network project, for collecting and processing cross-layer trigger information. The trigger management system is bound to the VERHO vertical handover controller system, enabling the handover decisions based on several input parameters.

I. INTRODUCTION

The number of mobile devices is rising all the time. At the same time, they offer new wireless techniques new possibilities to create new services that users should be able to use whenever and wherever they choose, with optimized bandwidth and costs. The availability of wireless access networks with more and more bandwidth enables the possibility to use some streaming software or watch live video streams or TV broadcasts with mobile devices. The effective use of new services in IP networks also sets several requirements. One of the key issues is to support for the mobility in the infrastructure. Mobile devices also have several different access technologies available, like WLAN, 3G, GPRS. Supporting mobility will give people more freedom with regard to the time and place for using IP-based services.

The use of different access technologies needs vertical handover [1], a mechanism for handover between the technologies. Figure 1 illustrates a situation in a heterogeneous environment when a Mobile Node is moving through an area with GPRS, WiMAX and IEEE 802.11 hotspots available. There are already many protocols designed to support vertical handovers, such as those described in more detail in section II. These protocols also need support for making the decisions for the handovers. The optimal situation would be that the data flow could be on all the time during the handover with no interruptions in the services. To achieve seamless handover would be easier if the connection to the new access point could be built before the old connection breaks. Unfortunately, many layer 2 technologies require that the connection to the old access point must be cut before the new one can be established. However, the first step before the handover and updating the location of the Mobile Node is the decision about the handover. It may also lead to the decision about using a different access technology. Usually it is based on the availability of the network, which is, of course, a basic requirement for keeping the connection established. The decision much depends on the services and applications the user wants to use (required bandwidth, etc). There is also a significant amount of information, like traffic load, available in the network. Such facts as the actual cost of using a technique may be quite important for users as well. It is also shown by other studies that the use of cross-layer information provides significant benefits for making decisions about adaptations for multimedia stream [2,3]. There are hundreds of different types of events related to mobility management available - see, for example, [4–8] - and it can be argued that the use of cross-layer information and enhancements for mobility mechanisms (like handovers) is the way towards seamless mobility.

The rest of the paper is structured as follows. Section II visits the state of the art of mobility management and presents some of the protocols needed for supporting mobility. Section III describes the VERHO vertical handover controller that can be used for making the decisions about the handovers. The Trigger Management Framework, which takes care of the collecting and processing of events (cross-layer information) that are then used to trigger the handover decisions, is described in section IV. Prototyping and the results of experimentations are explained in section V and section VI concludes the paper.

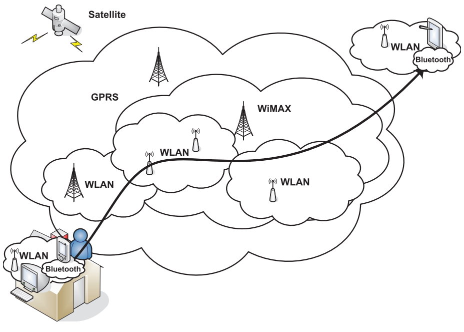


Figure 1: Movement of Mobile Node in heterogeneous environment

II. MOBILITY MANAGEMENT AND PROTOCOLS

Several mechanisms for changing the access between IP networks while moving are introduced. The Mobile IPv6 protocol (MIPv6) [9] handles mobility in IPv6 networks. There are still

1-4244-1144-0/07/$25.00©2007 IEEE

a number of drawbacks, such as high latency or packet loss, caused by the procedures that are needed during the handover execution phase [10]. Perhaps the biggest problem in handover is that the connection breaks during the handover, which causes some packet loss.

The Hierarchical Mobile IPv6 [11] introduces a Mobility Anchor Point that acts as a local Home Agent for Mobile IP. The benefit is that the signalling from the Mobile Node is needed only with the first Mobility Anchor Point, which is expected to be closer than the Home Agent; thus the signalling time can be reduced.

Fast handover for Mobile IPv6 [12] is another mechanism. In Fast Mobile IPv6 there is a router located in the access network that may work as a proxy for the mobile node. This proxy router uses tunnelling to route the packets to the new access router in the new access network. Fast Mobile IPv6 also uses link layer information that triggers the handover earlier than would otherwise be possible.

Mobile IP also has extensions to support network mobility (NEMO Protocol) [13]. The basic idea of NEMO is to use a gateway called Mobile Router to provide the Internet access for the moving network. Only Mobile Router will change its point of attachment to the Internet. NEMO is used to hide the mobility and other nodes see the moving network as a normal subnetwork and use Mobile Router as a default gateway.

Host Identity Protocol, HIP [14, 15] is also a mechanism for mobility support. HIP propose a new Host Identity layer between the network and transport layers. With this layer it is possible to map transport connections to host Identifiers instead of IP addresses. Contrary to IP addresses, Host Identity stays constant if a host is changing the network. Host Identity is a unique public cryptographic key of a public/private key pair. Using these cryptographical keys, Host Identity is a security cornerstone in the HIP architecture.

The IETF Detecting Network Attachments (DNA) working group [16] has been working with mechanisms enabling the mobile node to detect its connectivity status. When the mobile node is changing its IP network it needs to re-configure the IPlayer parameters, such as IP address and default gateway information, and DNS server address. Detecting the network change can usually be done by network layer indications like a change in the prefixes for IPv6, if this kind of information is available. The DNA working group has proposed link-layer notifications like a notification indicating that the node has established a new link-layer connection. That notification can be used to trigger probing the network for a possible configuration change [16].

The IEEE 802.21 working group [17] is working with the proposal for Media Independent Handover. This working group has described mechanisms supporting upper layer protocols and applications to participate in controlling the handover. This also supports more intelligent handover decisions by providing a means to collect information related to the capability of neighbouring networks. The IEEE 802.21 work is also partly trying to solve problems related to handovers and decisions about them.

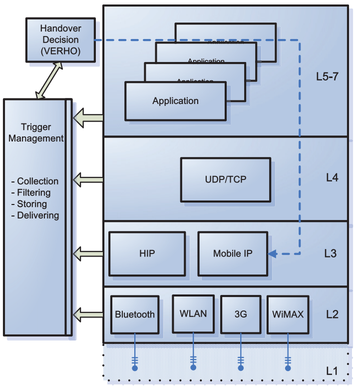


Figure 2: Cross-layer information

III. HANDOVER DECISIONS

The VERHO Vertical handover controller [18, 19] is designed to make the decision about which available access technology should be used. The system uses the information deduced from the decision process and controls the MIPv6 handover. The decision about when the handover should occur depends on the information collected from each available technology, which can originate from different protocol layers, as shown in Figure 2. The collecting and processing of this information, called trigger information, is described in more detail in section IV.. The most significant information is the information received from the link layer, such as link up/down. The link layer offers a different set of information depending on the technologies available. The use of link layer information also improves the mobility protocol handovers. The VERHO system is able to use several other sources of information, like user-specified information and the Quality of Service requirements of applications. The main goal for the system is to realize the intelligent decision based on the available information, and control the handovers and achieve a seamless vertical handover by performing a proactive handover. Papers like [20] have introduced the means for using several information sources as well. This paper supports those views and fully agrees that the solution for a decision is needed and it is not too complex to create. It may also be beneficial, not only for controlling the handovers but also for triggering the search for new possible access points.

The VERHO system is working completely on the Mobile Node side and the system does not need support from the network side, although network-assisted handovers are possible. The main idea is to use the information that is available for the Mobile Node. The design of the system is such that it can support possible future systems. The future systems can be systems on the network side that offer the handling of the authentication, security or the mechanisms for a better quality of service. Making the decision in the mobile node allows the users to give their own demands for the connection as well. Cost information, which is obviously very important for the users, can be one parameter for the decision. Users can, for example, predefine the cost of the use of different access technologies in terms of time. When there are several different access technologies available, such as WLAN and 3G, the efficient use of different interfaces with minimized power consumption while staying connected are very important. The applications may have their own demands for the required bandwidth or maximum delay of the link or the supported media codecs and adaptation methods. The availability of the link is of course the most important factor. This availability information can be gathered and derived from the link layer. Other sources of information, like location information from the GPS, could also be a beneficial source of information for the decision.

The decision in the VERHO system is based on the defined rules compared to the properties of the interface to the available techniques. The rules are divided into critical and non-critical rules. The critical rules are those rules that have to match before the interface is enabled for the decision process. The necessary critical rules are based on the state of the interface and link (Link Up and Link Down). The system also provides a possibility for the user to give some critical rules concerning, e.g., the type of the interface or even the maximum cost of the interface to be used. If the requirements of the critical rules match, the interface is available for the decision. A rule can be like this: Interface state = UP AND Interface type = IEEE 802.11b. This rule means that the user wants to use only WLAN.

The decision among those interfaces that match the critical rules is based on the preference number. Each interface will get a calculated preference value based on the values of different parameters. The parameters have weighted numerical values so that the most significant parameters will have greater weight. Each interface has several parameters like cost, power consumption, support for wireless QoS or bandwidth. If a parameter is lacking in a specific interface due to its different technology, that parameter can have a default constant value. The value of the preference number is calculated from each interface's parameter values. One group of parameters can be the measured QoS parameter's value. The QoS parameter value can be IP-level End-to-End quality measured with a separate software tool like QoSMeT [21]. Each measured parameter - e.g., delay, RTT and jitter - can then have the weighted value for the calculation of the preference value of the interface. There are also parameters like the type of interface without any numerical value. Those parameters can be assigned with a default numerical value. Deriving the value can be based on the user demands or, e.g., on the speed of the connection. By following the current state of the specific interface's properties, the system is able to choose whether there is other interface available that better matches the specified rules. The presented way of using the calculated preference value is just one possible way of deriving the final decisions and does not exclude the use of other suitable algorithms. All the information can be gathered using the Trigger Management Framework presented in the next section.

IV. CROSS LAYER TRIGGERING

One of the key components supporting the use of cross-layer information is the mechanism that supports the collection and processing of cross-layer information. One supporting mechanism is the Trigger Management Mechanism [22]. This mechanism has been developed in the EU's Ambient Networks Project [23] and is one of the key concepts of mobility management in Ambient Networks.

The event sources providing the usable information for handover decisions need to be able to deliver notifications to interested applications or other system entities like the mobility management protocols or the software tools controlling the handover like VERHO. It would be necessary to have the processing of the notifications in a common standardized manner. These standardized notifications are called triggers in this paper. It is also beneficial to group similar and related triggers as shown in [22]. The Trigger Management mechanism enables efficient collection and propagation of triggers originating from a variety of sources. The Trigger Management provides a framework where new sources of triggers and the possible users of the source information, called consumers (like VERHO), can be included as necessary. The mechanism aims at providing the means to propagate mobility-related information (triggers) between one or more event sources and several trigger consumers. The mechanism does not make handover decisions itself. This is done with systems like VERHO. Triggers can be classified and filtered based on five criteria: type, origin, occurrence/frequency, event persistence, and temporal constraints. Trigger types can be based on whether an event may, will, or must force a handover. The final decision on the handover is made by a different entity but this entity can already make some separation of different types of triggers with the help of Trigger Management. The origin of the event corresponds to the source that is producing the trigger, for example the radio access component. The source can be a system that listens to the events in layer 2 and uses the polling for a source that does not create any event to listen to. An example of this kind of source used in an ambient networks architecture [23] is the general link layer approach [24]. An event may also be periodic, like a network utilization measurement, or asynchronous, like the availability of a new network access. Finally, events can be either transient or persistent, and they may be associated with a real-time constraint. An advantage of our grouping and classification approach is that correlations between otherwise disparate triggers can be detected. This prevents the generation of transient triggers and reduces the number of events that need to be processed.

The Trigger Management Framework has four major components. First, events are collected from event sources via a collecting interface and are then further processed in the classification engine, which is also responsible for time-stamping and reformatting triggers, if necessary. Assigning triggers to the appropriate group is also done with the classification engine. The classified triggers are then stored in the repository.

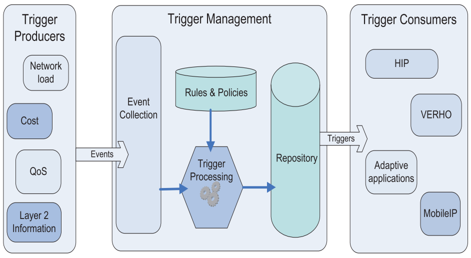


Figure 3: Trigger Management Framework

Each trigger typically has an expiration time, after which it is automatically removed from the trigger repository. The Trigger Management architecture supports the application of different triggering policies. The current implementation supports policies to restrict the accessibility of certain triggers to certain consumers. It is also possible to extend the policies to support the use of different rules for filtering and classification. This means that Trigger Management may use a different policy at different times or when in different contexts, without having to regenerate rules. The current implementation of the Trigger Management framework focuses on mobility management procedures and handles different triggers originating from radio access, network, upper layer events, and sensor data. Triggers can be formed depending on the system or application context. Events can originate from other devices and follow security restrictions and user requirements. The cornerstone part of the Trigger Management framework is entities as consumers and producers of triggers. Consumers such as VERHO must specify their needs to receive triggers. Consumers can also choose to stop receiving triggers at any time. The same entity may be simultaneously acting as both a consumer and a producer. Consumers can ask to receive all triggers originating from RAN event sources, but deny receiving the upper-layer triggers associated with security policy violations. This can be done by using the classification and filtering functionality. The consumer can use these triggers to generate its own context-related triggers and act as a producer for other functions (consumers) as well. It is expected that the Trigger Management framework will be used to feed and guide handover decisions and execution. It is possible for consumers to use the received triggers to decide whether the mobile device is moving within a single network or is crossing different access technology boundaries. It is also possible to monitor whether the addressing scheme, trust domain and provider domain should be changed accordingly.

V. PROTOTYPING AND EXPERIMENTATIONS

While the performance tests and scalability analysis is still ongoing work, the presented functionalities have been implemented and tested in several proof of concept prototypes. The current focus is to continue developing and testing the Triggering Management framework. The VERHO system, partly developed in previous research, has been tested using link layer information to assist the Mobile IPv6 handovers. These experiments are detailed in paper [19].

The Triggering Management framework is currently integrated into the Ambient Networks integrated prototype [25]. In that prototype the Triggering Framework is used to assist HIP protocol handovers having the Generic Link Layer GLL [24] acting as a triggering source for providing information about different access with triggers by monitoring the link status, routing table and IP address changes. The prototype also realizes the support for network mobility using the HIP mobile router. In addition to supporting vertical handovers, HIP also supports handovers between IPv4 and IPv6 networks.

To show and validate the functionality of the Trigger Management Framework, it has been used to assist entities other than mobility protocols in session handovers using sensor information. The functionality for supporting the movement of a session between different devices is here called a session handover. It is quite different from the traditional handover (i.e. changing the network access point). In a test bed, the currently available techniques with laptops and PDA was used for proving this concept. Using the developed mechanism it is possible to move the end point of the stream between the devices inside an ad hoc network in such a way that it is transparent to the streaming server. The mechanism is based on the idea of using a virtual interface for the data stream endpoint that can be moved and configured to different devices. The target devices should have support for continuing the same session, which, in our case, was a video stream. Enabling this type of session movement gives the user a possibility to use the most suitable device for watching the video stream at all times. For example, at home the user most probably has the possibility to watch the video on a bigger screen and while moving the user can continue watching using a handheld device like a PDA or a smartphone. More details about the scenario and prototype can be found in [26].

While the demonstration scenario and prototype implementation was used as a proof of concept demonstration, some preliminary tests were done as well. Tcpdump traces were used to monitor the udp video stream and showed that in all of the experiments there were no packet losses. In the test case, video was streamed to the laptop during $t = 0 \ldots 3$ and $t = 7 \ldots 10$ min. During $t = 3 \ldots 7$ min the video was streamed to the PDA. In the handover case, as few as 9 packets captured at the server trace had inter-arrival times exceeding 0.1 s. Closer evaluation of the packet numbers showed that all 9 packets with inter-arrival times exceeding 0.1 s were sent during at the time the stream was on the PDA and 5 consecutive packets when the handover was executed. It should be noted that this is a prototype implementation of Trigger Management and taking into consideration that the PDA was running the video client and captured packets using tcpdump, these preliminary test were very promising.

Early performance experimentations were also started where

100,000 triggers were submitted to Triggering Management. Measurements showed that it takes an average 1 ms to collect, process, store and deliver a trigger. This is a good indication that Trigger Management is already suitable for these kinds of scenarios. More evaluations and performance tests in real and simulation environments are part of currently ongoing work.

VI. CONCLUSIONS AND FUTURE WORK

This paper presented the Trigger Management Framework mechanism for collecting information from various event sources and processing the collected events to a standardized trigger format. These triggers can be further used to feed handover decisions to enable seamless handovers. A vertical handover controller (VERHO) system was also presented as an example for the decision mechanism that can benefit from trigger information. The paper also presented some preliminary results from the prototyping experiments. The current focus is to finalize the Trigger Management implementation, which has already been integrated to feed HIP, Mobile IP, and Session Handovers. While the implementation is intensively being tested and prototyped in a real heterogeneous environment (3G, WLAN, WiMAX), a simulation environment will be built to fulfil the testing needs for the performance and scalability analysis. The forthcoming studies will also show how the Trigger Management Framework maps with the new standards like IEEE 802.21.

VII. ACKNOWLEDGEMENTS

Author would like to thank colleagues who have been working with the presented topics in VERHO, MERCoNe and Ambient Networks Projects. Special thanks also to Dr. Marko Jurvansuu and Dr. Kostas Pentikousis.

REFERENCES

[1] J. Manner and M. Kojo, Mobility Related Terminology, RFC 3753 (Informational), June 2004.

[2] J. Huusko, J. Vehkapera, P. Amon, C. Lamy-Bergot, G. Panza, J. Peltola, and M. G. Martini, Cross-layer architecture for Scalable Video Transmission in Wireless Network, *Elsevier Signal Processing: Image Communication 22*, Apr. 2007.

[3] Maria G. Martini, Matteo Mazzotti, C. Lamy-Bergot, J. Huusko, and P. Amon, Content adaptive network aware joint optimization of wireless video transmission, *IEEE Communication Magazine, Special Issue on Advances in Visual Content Analysis and Adaptation for Multimedia Communication*, Jan. 2007.

[4] P. Prasad, W. Mohr, and W. Konhuser, ”Third Generation Mobile Communication Systems.” *Boston: MA: Artech House Publishers*, 2005.

[5] P. Vidales, J. Baliosian, J.Serrat, G. Mapp, F.Stajano, and A.Hopper, Autonomic system for mobility support in 4g networks, *IEEE JSAC*, vol. 423, no. 12.

[6] S. Ishihara, K.Koyama, G.Miyamoto, and M.Kuroda, Predictive rate control for realtime video streaming with network triggered handover, *in IEEE WCNC*, vol. 3, no. 13-17, Las Vegas, Nevada, USA, 2005.

[7] H. Chaouchi and P. Antunes, Pre-handover signalling for QoS aware mobility management, *in International Journal of Network Management*, 2005.

[8] E. Casalicchio, V. Cardellini, and S. Tucci, A layer-2 trigger to improve QoS in content and session-oriented mobile services, *in Proceedings of the 8th ACM international Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Montral, Quebec, Canada, Oct. 2005.

[9] D. Johnson, C. Perkins, and J. Arkko, Mobility Support in IPv6, ser. Request for Comments, no. 3775. IETF, June 2004.

[10] S. Fu and M. Atiquzzaman, TraSH: A Transport Layer Seamless Handover Scheme for Space Networks, *in Nasas earth science technology conference*, 2005.

[11] H. Soliman, C. Castelluccia, K. E. Malki, and L. Bellier, Hierarchical Mobile IPv6 Mobility Management (HMIPv6), ser. Request for Comments, no. 4140. IETF, Aug. 2005.

[12] R. Koodli, Fast Handovers for Mobile IPv6, RFC 4068 (Experimental), July 2005.

[13] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, Network Mobility (NEMO) Basic Support Protocol, RFC 3963 (Proposed Standard), Jan. 2005.

[14] R. Moskowitz and P. Nikander, Host Identity Protocol (HIP) Architecture, ser. Request for Comments, no. 4423. IETF, May 2006.

[15] P. Nikander, J. Ylitalo, and J. Wall, Integrating Security, Mobility, and Multi-homing in a HIP Way, *in Network and Distributed Systems Security Symposium (NDSS03)*, San Diego, CA, USA, Feb. 2003.

[16] Detecting Network Attachments *IETF working group, work in progress.*, 2006.

[17] Media Independent Handover, *IEEE Standard 802.21, work in progress*.

[18] J. Mäkelä and T. Hämäläinen and G. Fekete and J. Narikka, Intelligent Vertical Handover System for Mobile Clients, *in Proceedings of the 3rd International Conference on Emerging Telecommunications Technologies and Applications ICETA*, Sept. 2004.

[19] J. Puttonen, G. Fekete, J. Mäkelä, T. Hämäläinen, and J. Narikka, Using Link Layer Information to Improve Vertical Handovers, *in Proceeding of the IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*,PIMRC, Sept. 2005.

[20] P. Vidales, R. Chakravorty, and C. Policroniades, PROTON: a policy-based solution for future 4G devices, *in Proceedings of Fifth IEEE International Workshop on EPolicies for Distributed Systems and Networks*, June 2004.

[21] J. Prokkola, M. Hanski, M. Jurvansuu, and M. Immonen, Measuring WCDMA and HSDPA Delay Characteristics with QoSMeT, *in Proceedings of IEEE International Conference on Communications*, 2007.

[22] J. Mäkelä and K. Pentikousis, Trigger Management Mechanisms, *in International Symposium on Wireless Pervasive Computing*, San Juan, Puerto Rico, Feb. 2007.

[23] N. Niebert, A. Schieder, H. Abramowicz, G. Malmgren, J. Sachs, U. Horn, C. Prehofer, and H. Karl, Ambient Networks An Architecture for Communication Networks Beyond 3G, *IEEE Wireless Communications (Special Issue on 4G Mobile Communications Towards Open Wireless Architecture)*, vol. 11, no. 2, Apr. 2004.

[24] Ramón Agüero, Jens Gebert, Johnny Choque, Haradl Eckhardt, Towards a Multi-Access Prototype in Ambient Networks, *in 16th IST Mobile & Wireless Communications Summit*, July 2007.

[25] P. Pääkkönen, P. Salmela, R. Agüero, and J. Choque, An Integrated Ambient Networks Prototype, *International Conference on Software, Telecommunications and Computer Networks, SoftCOM2007*, Split - Dubrovnik, Croatia, September 2007.

[26] J. Mäkelä, R. Agüero, J. Tenhunen, V. Kyllönen, J. Choque, and L. Munoz, Paving the Way for Future Mobility Mechanisms: A Testbed for Mobility Triggering & Moving Network Support, *in Proceedings of 2nd International IEEE/Create-Net Tridentcom*, Barcelona, Spain, Mar. 2006.

PVII

MOBILITY TRIGGER MANAGEMENT: IMPLEMENTATION AND EVALUATION

by

J. Mäkelä, K. Pentikousis and V. Kyllönen 2009

International Journal of Communications, Network and System Sciences, Vol.2, No.3, (IJCNS 2009)

MOBILITY TRIGGER MANAGEMENT: IMPLEMENTATION AND EVALUATION

Jukka Mäkelä, Kostas Pentikousis and Vesa Kyllönen
VTT Technical Research Centre of Finland
Kaitoväylä 1, FI-90571 Oulu, FINLAND
Email:firstname.lastname@vtt.fi

Abstract— Modern mobile devices have several network interfaces and can run various network applications. In order to remain always best connected, events need to be communicated through the entire protocol stack in an efficient manner. Current implementations can handle only a handful of low level events that may trigger actions for mobility management, such as signal strength indicators and cell load. In this paper, we present a framework for managing mobility triggers that can deal with a greater variety of triggering events, which may originate from any component of the node's protocol stack as well as mobility management entities within the network. We explain the main concepts that govern our trigger management framework and discuss its architecture which aims at operating in a richer mobility management framework, enabling the deployment of new applications and services. We address several implementation issues, such as, event collection and processing, storage, and trigger dissemination, and introduce a real implementation for commodity mobile devices. We review our testbed environment and provide experimental results showcasing a lossless streaming video session handover between a laptop and a PDA using mobility and sensor-driven orientation triggers. Moreover, we empirically evaluate and analyze the performance of our prototype. We position our work and implementation within the Ambient Networks architecture and common prototype, centring in particular on the use of policies to steer operation. Finally, we outline current and future work items.

Key words: Triggering, Mobility Management, Mobile networks, Handover, Cross-layer Information Management

1. Introduction

Modern mobile devices, such as smartphones, internet tablets and PDAs, have several network interfaces and can run various network applications, like web browsers, email clients, and media players. Indeed, it is becoming common that said devices can take advantage of wireless LAN, PAN and cellular connectivity, and we expect that in the coming years mobile WiMAX will be supported as well. In such a multiaccess environment, mobility management support for both horizontal and vertical handovers should be one of the basic functionalities in future devices. Moreover, in order to allow a mobile device to remain always best connected, several events need to be communicated through the entire protocol stack, as we explain in the following section. Nevertheless, current implementations of state-of-the-art mobility management protocols, such as Mobile IP [1] or Host Identity Protocol [2]), can only handle a small set of event notifications that may lead to mobility management actions, including handover execution.

In this paper, we argue for a novel mobility trigger management framework that can handle a much larger set of notifications related to events originating not only from the lower layers of the protocol stack (physical, data link, and network), but also from the upper layers enabling the efficient use of cross-layer information for mobility management. This framework needs to be open, flexible, with low overhead, and incrementally deployable. After describing the main parts of the architecture, we present the implementation of such a framework, which allows mobile devices to manage, on the one hand, conventional mobility events, such as the availability of a new network access, received signal strength indications (RSSI), network capacity load and, on the other hand, higher level events, such as security alerts, policy violations, end-to-end quality of service deterioration, and network access cost changes. In our framework, event sources can deliver notifications to interested applications and other system entities used in a standardized manner. We will refer to these standardized notifications as triggers in the remainder.

The main elements of our trigger management framework are detailed in [3][4], and include the entities which generate the events (producers) and entities that use the trigger information (consumers). Our trigger management framework is capable of collecting event information from various producers through a specific collection interface. The collected events are then processed and converted into a unified trigger format, described in Section 5, and distributed to interested consumer entities. A trigger consumer can be any entity implementing the collection interface and can be located in the same or in different node in the network. It should be noted also that a same entity can act both as a producer and a consumer.

In this paper we concentrate on the evaluation of the implementation of our framework in the VTT Converging Networks Laboratory. Indeed this paper demonstrates the feasibility of our designed framework over a real testbed network. The concept and architecture behind our framework with some analysis to the similar existing concepts are also summarized below.

The rest of this paper is organized as follows. Section 2 introduces the fundamental elements of our framework for managing triggers, reviews the related work in this area and motivates our evaluation. Section 3 presents our implementation of the triggering framework and Section 4 discusses the role of policies and rules in the system design. Results from our experimental lab evaluation are presented in Section 5. Related work is discussed in Section 6, and Section 7 concludes the paper.

2. A Framework for Managing Mobility Triggers

After surveying the relevant literature (see, for example, [5], [6]–[10]), and based on our own expertise, we identified more than one hundred different types of network events related to mobility management. We cluster triggers, regardless of the underlying communication technology, based on groups of events related to changes in network topology and routing, available access media, radio link conditions, user actions and preferences, context information, operator policies, quality of service (QoS) parameters, network composition [11], and security alerts.

Fig. 1 illustrates six different trigger groups as boxes. The "offshoots" on top point to example triggers belonging to each group. The rightmost group includes representative link layer "up/down" triggers (irrespective of the radio access technology). The leftmost group includes triggers originating from the application layer. In this example, certain triggers originate from the node ("System Resources") while others originate from the network ("Macro Mobility"). The "origin" corresponds to the entity that produces the trigger, for example, the radio access component. An advantage of our grouping approach is that it allows us to detect relations between otherwise disparate triggers. This prevents the generation of excessive transient triggers based on, for example, the same root-cause event, such as a link outage, and reduces the number of events that need to be processed.

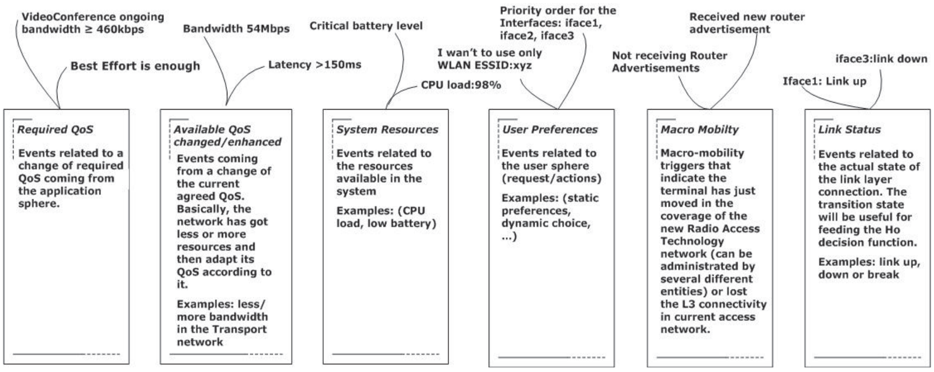


Figure 1 An example of trigger groups

Event sources need to be able to deliver notifications to interested applications and other system entities in a uniform, concise, and standardized manner. This approach simplifies notification handling considerably, while guaranteeing sufficient diversity for event separation and classification. In order to manage and efficiently propagate triggers originating from a variety of sources we developed a trigger management framework, which we call TRG. TRG lays the foundation upon which sophisticated handover (HO) operations can be performed. We aim at establishing an extensible framework where new sources of triggers can be defined and included in a system implementation as necessary. Note that this is quite different from other, in our opinion, more closed and specific approaches, such as the one followed in the IEEE 802.21 [12] working group. On the surface, both TRG and the IEEE 802.21 Media Independent Handover Services standard seem quite similar, aiming to improve mobility management performance. However as we argue in [13] the mechanisms and services introduced by the IEEE standard do not include dynamic information elements and any extensions will have to be introduced with lengthy standardization procedures in the future. Moreover, triggers cannot originate from the higher layers of the protocol stack, and system level events are simply out of scope of IEEE 802.21. Finally, 802.21 provides services to command and use the lower layer information to enable seamless handovers and multiaccess, which is not in the domain of TRG, but of the mobility management protocol. Last but not least, TRG is designed to handle much more event sources than MIHF. It is important to highlight that TRG provides the means to disseminate and filter mobility-related information between one or more event sources and several trigger consumers but that HO decisions are still the responsibility of the mobility management protocol, say, Mobile IP [1] or HIP [2]. TRG can also provide hints about moving the communication endpoint from one device to another, as explained in Section 5.

A central part of the design is designating different system entities as producers and consumers of triggers. Policies, described in Section 4, are handled by the Policy Manager. For communicating with different entities, TRG exposes three service access points (SAP). Event sources use the Producer SAP, to register events and emit notifications to TRG when changes occur. Consumers use another SAP, to subscribe with TRG and receive triggers in a single format when they become available. Finally, the Policy Manager uses another SAP to inform TRG about policies. Internally, TRG implements a local trigger repository and functional blocks for processing triggers.

Consumers must state their need to receive triggers and can choose to stop receiving them anytime. For example, the Mobile IP daemon can receive all triggers related to link layer events, but opt to receive only the upper-layer triggers associated with security or policy violations. In the former case, such a consumer takes advantage of the trigger grouping functionality; in the latter, it additionally requests trigger filtering. Consumers can use these triggers to generate their own and, thus, serve as an event producer for other entities. We expect that TRG will be used to guide HO decision making and execution. In particular, consumers can use triggers to derive whether the mobile device is moving within a single network or it is crossing different access technology boundaries, and whether the addressing scheme, trust and provider domains should be changed accordingly.

3. Architecture and Implementation

The core implementation of TRG has three major components: triggering event collection, trigger processing, and the trigger repository [3][4]. Triggering events collection receives events from various sources in the network system via the trigger collection interface. New triggers can be introduced in a straightforward manner by implementing the trigger event collection functionality and supporting the trigger collection interface. The latter allows sources to register their triggers and to make them available to consumers. A specific TRG implementation may contain several event collectors, which may be distributed, and are responsible for collecting different types of events. The trigger repository is designed to meet the stringent requirements placed by mobility management, but can be used to store non-mobility triggers as well. The basic primitives include adding, removing, updating, and disseminating triggers in a standardized format. Each stored trigger has an associated lifetime and is removed automatically once its time-to-live (TTL) expires.

The need for different event collectors arises from the fact that the origin of an event source can be a hardware device, a system component implemented in kernel space, or an application implemented in user space. For example, each device driver could implement its own event collection functionality, which would be capable of handling triggering events produced by the specific device only. Moreover, sources can also be located in the network such as at active network elements or at the user's home network. Finally, a particular TRG implementation can act as a consumer to another TRG located in a different node. Thus, orchestrating the collaboration of, perhaps, several collection entities is needed in order to efficiently gather a larger amount of events.

Having dedicated collectors for different event sources enables the use of TRG in different operating systems as well. The collector can format the events to the format that TRG understands and there is no need to modify the core of TRG functionality; instead the collector can be modified as necessary. This is also one of the key points in the architectural design of TRG that enables it to handle cross-layer information by having a collector at different layers as needed. For example TRG can get similar information considering the connectivity in FreeBSD through a collector that uses Route Socket and in Linux through a similar collector using RTnetlink socket but obviously these collectors need to have their own implementation. The core of TRG could be implemented in kernel space for performance reasons and allowing for direct access to lower layer information. On the other hand, TRG can be implemented in application space allowing for greater flexibility and easing implementation and code evolution. The prototype described in this paper follows the latter approach. Of course, certain event collectors will have to be implemented closer to the lower layers in the future.

The event sources are connected with TRG via producer SAP, as described also in section 2. The performance of the event collectors is obviously very important. They need to be fast enough to react to all different events, but the collector implementation itself is not part of the TRG framework architecture. TRG provides the interfaces to connect different event producers with the possible consumers by defining the SAP's between TRG and them. TRG core functionality per se provides the mechanisms for distributing, filtering and handling the policies for the whole system of the mobility event handling, but the collectors are out of scope of this paper. Fig. 2 illustrates the TRG framework with the different event producers and consumers.

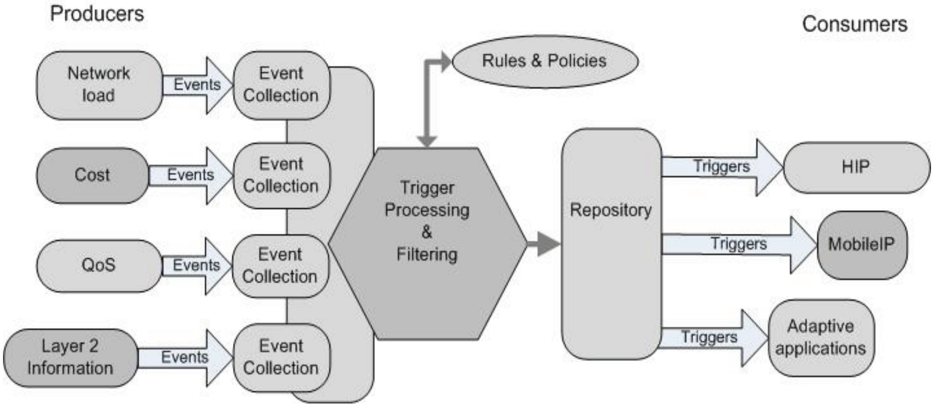


Figure 2 TRG Architecture

After events are collected from the producers, they are handed over to the trigger processing engine which is responsible for time-stamping and reformatting triggers (if necessary), and assigning them to the appropriate group. Consumers can subscribe by specifying a set of triggers (and, optionally, filtering rules) and are expected to unsubscribe when they do not wish to receive them any longer. For each consumer subscription, TRG makes sure that filters are grammatically and syntactically correct, and accepts or rejects the subscription. Basic rules can also be used as building blocks for crafting more sophisticated rules.

4. Policies and Rules in TRG

TRG supports the application of different triggering policies, defined as a set of classification, filtering, trust, and authorization criteria/rules. This allows our implementation to enforce a different policy at different times or when the node operates in different contexts. The availability of a system-wide policy and consumer-supplied filters lies at the centre of our TRG design. These two are orthogonal, providing flexibility and adaptability.

System policies ensure that only designated consumers can receive certain groups or types of triggers. For example, a node may operate under different policies regarding network attachment depending on whether the user is on a business or a leisure trip. Policies can also establish different trigger classification and groupings in different contexts and are typically stored in a separate repository, accessible to the TRG implementation. Filters allow a consumer to focus a trigger subscription. For example, a monitoring application may be interested in receiving all network utilization measurements, while a VoIP application may be interested in receiving a trigger only when utilization exceeds a certain threshold and the user is in a call. In fact, a VoIP application can even opt to be an intermittent trigger consumer, subscribing and unsubscribing to receive certain triggers solely when needed.

Our TRG implementation uses access control policies to define:

- Which producers are allowed to register and send triggers to TRG. Producers are identified by the trigger IDs they register, and can be chosen on a system basis. For example, a policy allows only specific producers to register with TRG.

- Which consumers are allowed to subscribe to triggers. Policies can be very specific, prescribing which consumers can receive certain triggers and from which producers. Consumers are identified by their locator (typically a host address). For example, in our proof of concept implementation described in Section 5, we can enforce a policy that dictates that triggers from producer with ID=50 are allowed to be subscribed only from "localhost" entities.

The Policy Manager applies access control using policies described in XACML (OASIS eXtensible Access Control Markup Language) [14]. Fig. 3 illustrates which Policy Manager functions are called when a producer registers or a consumer subscribes. Typically the decision on whether to allow producer registrations and consumer subscriptions is made immediately based on the system policies and the result is returned to the initiating entity. In the case where a consumer attempts to subscribe to all triggers, the decision may be deferred for when triggers become available. That is, the subscription for "all triggers" effectively becomes a subscription for "all triggers allowed", when system policies dictate so. In our current prototype implementation, policies are described using access control lists read from a configuration file. Policies also define which consumers are allowed to subscribe and for which trigger.

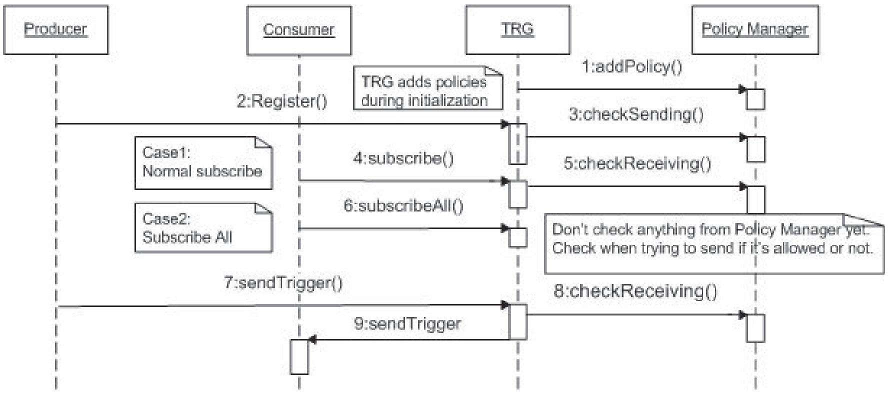


Figure 3 TRG-Policy Manager Message Sequence Diagram

5. Results

We tested our user-space C++ implementation of TRG on laptops running FreeBSD release 6.1, Linux Fedora Core 3 with kernel 2.6.12 and Windows XP, and on a PDA running Linux Familiar v.0.8.4 with kernel 2.4.19. Architecture design with the possibility to use separate event collectors in different environment, as discussed in Section 3, makes our TRG implementation portable and is currently being integrated in several prototypes, including the Ambient Networks [15] prototype [16, 17, 18]. For communication between producer, TRG and consumer a Web Service XML-based communication on top of HTTP was used. In this integrated prototype, TRG takes care of the delivery of all mobility-related events. Events were formatted according to the unified trigger format shown in Table 1.

Trigger data member	Type	Description
id	integer	Trigger identifier, same as producer identifier. Maps producer name to identifier.
type	integer	Specific to the trigger identifier. Mapping producer information to type.
value	std::string	Specific to trigger type.
timestamp	time_t	Time that a trigger enters the TRG repository.

Table 1 Trigger format

In previous work we presented a proof-of-concept testbed and demonstrated the feasibility of the concepts governing our TRG implementation. These preliminary validation results are summarized briefly in subsection 5.1; further details are available in [4]. Subsection 5.2 presents the first detailed results of our stress-test empirical evaluation of TRG in the lab.

5.1. Proof of Concept Validation

In [4, 19], TRG was employed to enable streaming video session handovers between different mobile devices. In the scenario, the user starts watching a video streamed to his laptop. His GNU/Linux PDA is nearby and the user decides to move to another room but would like to keep watching the video on the way. The commercial, off-the-shelf (COTS) PDA is augmented with a multi-sensor device (detailed in [20]), which was extended to provide "device orientation" triggers. For example, when the user picks up the PDA, a "vertical orientation" trigger is produced, initiating a session HO from the laptop to the PDA. The two devices have to coordinate and arrange for the transfer of the video streaming session. A successful session handover allows the user to receive the streaming video on the PDA over the WLAN seamlessly. The user can also explicitly initiate a session HO by pressing a PDA button. In this example, TRG handles triggers associated with mobility, orientation, and user preferences, keeping the video flowing smoothly while changing the communication end-point. Two logical topologies were evaluated in our lab. First, all devices are connected using IEEE 802.11 in ad-hoc mode, as if the user streamed a video from his digital collection at home. Second, the video streaming server is located in a different network, as would be the case when watching a video from a service provider over the Internet. For both setups in our lab proof-of-concept validation, we stream a 10- minute video encoded at 576 kb/s over UDP. At t = 3 min, a session HO from the laptop to the PDA is triggered, and at t = 7 min, the session is "moved" back to the laptop.

We captured all traffic traces during the experiments using tcpdump and cross-checked all packet IDs sent by the video server with the packet IDs received at the laptop and PDA video clients to confirm that no packet losses occurred. Moreover, the effect of TRG signalling and the actual session handover on packet delay is negligible, compared to packet delays before and after the session handover. Fig. 4 illustrates the packet inter-transmission times as recorded by tcpdump at the streaming server and the packet inter-arrival times at (a) the receiving laptop and (b) the receiving PDA. On the left-hand side, the packet inter-arrival time measured at the streaming server, laptop and the PDA during the delivery of the 10-min video stream are shown. The band around 50 ms indicates that the packets are sent and received in an orderly manner. We note a small number of inter-arrival times outside this band. The vast majority of inter-arrival times do not exceed 150 ms; only a handful of packets out of more than 12000 exceed this threshold. On the right-hand side of Fig. 4, we zoom in at around t=3 min when the first session HO is triggered from the laptop to the PDA. As the figure illustrates, only a few packets had >0.1 s inter-arrival time. These results are very promising, despite the fact that this is a prototype implementation, especially when taking into consideration that the PDA was running the video client and captured packets using tcpdump throughout the experiment leaving few spare system resources available.

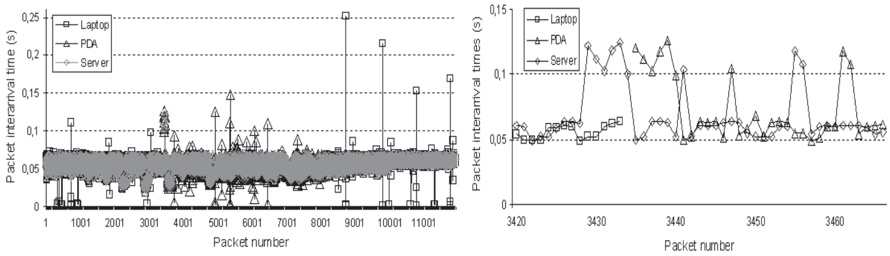


Figure 4 Experimental results when triggering a session HO

This paper focuses on the empirical validation and evaluation of TRG. The theoretical aspects (scalability, security, reliability) have been partly addressed elsewhere [21] and further analysis is also part of our future work agenda. It is important to note that these set of experiments go beyond showcasing the concept of TRG-assisted session HOs. This is simply a particular application of triggers leading to a HO. Instead, we emphasize that these experiments aim at assessing the feasibility of introducing a TRG implementation in small COTS handheld devices, a result which was not warranted when we embarked in developing TRG.

5.2. Experimental Evaluation

Since we conducted the experiments presented in the previous subsection, we continued the development and evaluation of TRG and used an updated implementation of TRG enhanced with web service interfaces and ran tests where we submitted 100000 triggers from several sources to TRG and delivered those to different consumers. We consider two test cases, with the aim of quantifying TRG performance under stress (and perhaps clearly unrealistic) conditions. Test Case 1 employs *n* producers connected with *m* consumers via TRG. During the test, each producer sends 100000 back-to-back triggers and all triggers are distributed to all *m* consumers. This means that TRG needs to process $n \times 10^5$ triggers and deliver $n \times m \times 10^5$ triggers. On the left-hand side of Fig. 5, we illustrate an example case where *n*=3 producers A, B and C each send 100000 triggers, with trigger IDs 51, 52 and 53, respectively, to *m*=4 consumers (labelled I, II, III, IV). That is, in this particular scenario, each of the four consumers will receive 300000 triggers from TRG.

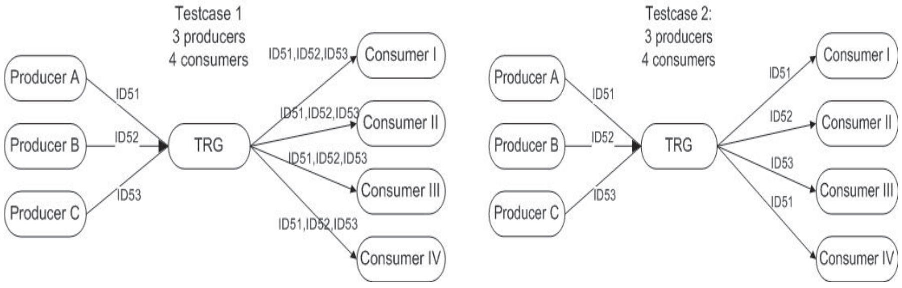


Figure 5 Triggers in Test Cases 1 (left) & 2 (right).

Table 2 shows the number of delivered triggers with average processing times in milliseconds for each trigger received by TRG from the producers in Test Case 1. In this case, only the number of consumers has a significant effect on the processing time of each trigger. This indicates that TRG can cope with several registered producers even when there is no subscribed consumer from certain producers. Moreover, the average trigger processing time is only few milliseconds per subscribed consumer in this stress test of the prototype implementation.

	Number of Producers				
Number of Consumers	1	2	3	4	5
1	**100k, 1.7 ms**	**200k, 1.7 ms**	**300k ,1.8 ms**	**400k, 1.7 ms**	**500k, 1.8 ms**
2	**200k, 2.3 ms**	**400k, 2.5 ms**	**600k, 2.4 ms**	**800k, 2.5 ms**	**1 000k, 2.4 ms**
3	**300k, 3.2 ms**	**600k, 3.2 ms**	**900k, 3.2 ms**	**1 200k, 3.1 ms**	**1 500k, 3.3 ms**
4	**400k, 3.7 ms**	**800k, 3.8 ms**	**1 200k, 3.8 ms**	**1 600k, 3.8 ms**	**2 000k, 3.8 ms**
5	**500k, 4.5 ms**	**1 000k,4.6 ms**	**1 500k, 4.7 ms**	**2 000k, 4.7 ms**	**2 500k, 4.5 ms**

Table 2 Total number of delivered triggers and average processing time (in ms) per trigger in test case 1.

Since there are several possible scenarios about how triggers are distributed between producers and consumers we made also a Test Case 2 setup, illustrated in the right-hand side of Fig. 5, where each consumer has only one dedicated producer. This means that TRG needs to process $n \times 10^5$ triggers and deliver $m \times 10^5$ triggers. If there are more producers than consumers, triggers will be distributed evenly between the available consumers. As mentioned above, all tests were made using a C++ implementation of TRG with a web service interface towards producers and consumers. We used a laptop with an Intel Pentium M 1.70 GHz PC with 1 GB RAM, running FreeBSD release 6.1 in the tests reported in this subsection.

Fig. 6 shows the total processing time of Test Case 1, with and without employing the TRG filtering mechanism. It can be seen that when the number of the consumers and producers increases, so does the total processing time. This is expected since the number of processed triggers is increasing when adding more consumers and producers. The costs of adding consumers and producers are both linear. But the cost of adding consumers is greater than the cost of adding producers. For example when comparing the calculated slope $k = \Delta y/\Delta x$ of the curves of total processing time, with and without filtering, we see that the processing time increases faster the more consumers are introduced (slope of the curve with one consumer $k = 177{,}6$ and with 5 consumer $k = 454{,}9$ in Test Case 1 without filtering), this can be explained as a cost of the duplication of triggers because the number of triggers that have to be duplicated and delivered to consumers increases when adding more consumers. Anyhow this does not increase the average processing time of one trigger. The number of producers has also effect to the total processing time, but not as much as the number of consumers.

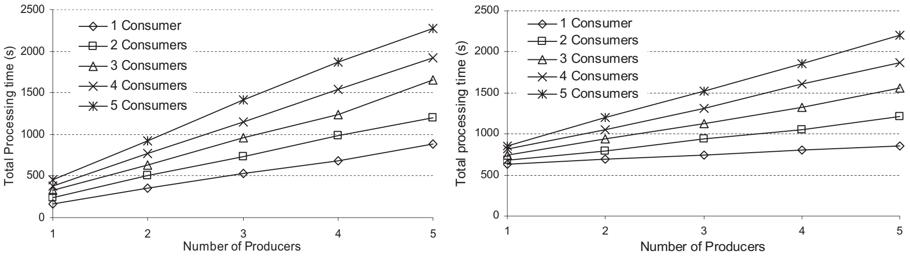


Figure 6 Total processing time in Test Case 1 without (left) and with (right) filter processing.

We also evaluated the cost of using the filtering function of TRG. With Test Case 1 we had all five producers registered and each one was sending 100000 triggers. It follows that TRG was receiving total of 500000 triggers during the test. The right-hand of Fig. 6 shows the total processing times when the filtering mechanism was used. When there is one producer, the triggers from the other four producers are filtered away, and the triggers from the sole producer are duplicated and delivered to all four consumers. In the case with two producers the triggers from three producers are filtered away, and so on. The results show that it takes more time to process all triggers but this is not caused by the filtering mechanism itself. When comparing the total processing times, in the case where triggers from 1 producer are delivered to consumers in Fig. 6, the total processing time is increased when the filtering mechanism is used, but this is because now there are five times more triggers received by TRG than in the case without the filtering mechanism, since all five producers are sending 100000 triggers all the time during the test. When the filtering mechanism is not used, the number of producers is controlled by making a new registration per producer.

To further quantify system behaviour when filtering is employed, we consider Test Case 2. When evaluating the filtering function in Test Case 2, each consumer had a filtering rule that was true for all triggers, allowing the distribution of all triggers to the subscribed consumers. By having this “receive all triggers” rule we were able to test the effect of the filtering mechanism, since every time a trigger is produced, TRG needs to run the filtering code before disseminating the trigger to consumers even though none of the triggers are in practice going to be filtered away. The purpose was to test the effect and cost of running the filtering function. The TRG filtering mechanism per se does not have a significant effect on the overall processing time, especially when compared to the effect of increasing the number of consumers. When comparing the processing times in Test Cases 1 (Table 2) and Test Case 2 (Fig. 7) we see that the duplication of each trigger to every consumer, needed in Test Case 1, increases processing times. In Test Case 2, when the number of producers and consumers are equal, the difference of the processing times can be measured in microseconds, since now there is no need to duplicate triggers.

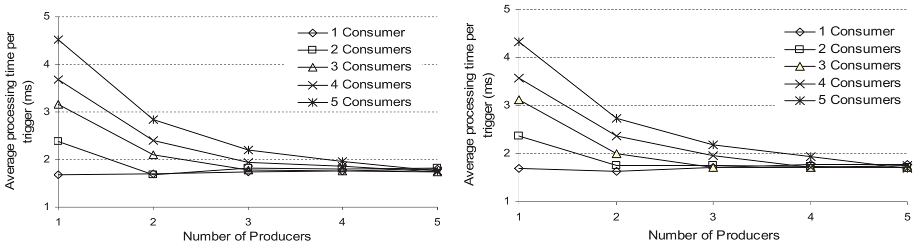


Figure 7 Average processing time in Test Case2 per trigger without (left) and with (right) filtering

The test and evaluation cases presented in this Section showed that it is in fact the duplication of triggers and number of messages that have a biggest effect on the processing time of triggers. It can be seen in Fig. 7, for *n*=5 producers and *m*=5 consumers, that the processing time does not depend on the number of consumers. There is no duplication of triggers in this case either.

The feasibility of using TRG to process, filter and disseminate a very large amount of triggering events was shown in practice. Each case showed that the processing time of one trigger does not increase, even when processing a huge amount of triggers. Although the stress-test cases are clearly unrealistic, they demonstrate that using TRG does not cause any major delays to handover times. On the contrary, TRG enables handover decision making mechanisms to react more rapidly and to larger set of events. It is also important to note that the TRG filtering mechanism does not have a major effect to processing times and this allows the handover decision making mechanisms to react faster to relevant events. Although the filtering mechanism can be used for the pre-decision about which events are to be collected, the handover decision per se is left to separate mechanisms with the decision algorithm. It was also shown that the cost of adding more consumers and producers increase processing times linearly and the cost of using filtering has only a marginal effect on the processing times. Of course the more triggers there are, the more total processing time is needed for processing and disseminating all triggers. However, by implementing grouping and classification of triggers [4] and having mechanism, e.g. in the TRG source for prioritizing trigger delivery which allows critical triggers to be processed and distributed faster, TRG is ready to process the triggering events.

6. Related Work and Discussion

Previously published work [7]–[9] shows the benefits of using event information, for example, to proactively perform a handover in order to maintain QoS levels. Our goal is to define a framework that supports the event collection and processing, and trigger distribution possibly from hundreds of different sources. We concur with Vidales et al. [7] that in heterogeneous network environments several sources of events and context information should be consulted in order to achieve seamless connectivity and develop swift mobility management mechanisms. Furthermore, earlier work in other event/notification systems [22, 23], which introduces mechanisms on how to implement such systems, along with the evaluated event generation cases is very encouraging and complimentary to our effort in defining TRG as a specialized notification system for mobility-related events which originate from the entire protocol stack.

The IEEE 802.21 Media Independent Handover (MIH) Services [12] working group is standardizing an information service that will facilitate media independent handovers. The scope of the IEEE 802.21 standard is to provide a mechanism that provides link layer intelligence and other related network information to upper layers to optimize handovers between heterogeneous IEEE 802 systems and facilitates HOs between IEEE 802 and cellular systems. IEEE 802.21 assists in HO Initiation, Network Selection and Interface Activation. The purpose is to enhance the experience of mobile device users. The standard supports HOs for both stationary and mobile users. For mobile users, HOs are usually needed when the wireless link conditions change. For stationary users, HOs are needed when the surrounding environment changes. Both mobile node and network may make decisions about connectivity. The HO may be conditioned by measurements and triggers supplied by the link layers on the mobile node. The IEEE 802.21 standard defines services that enhance HO between heterogeneous access links. Event service, Command service and Information service can be used to determine, manage and control the state of the underlying multiple interfaces. By using the services provided by MIH Function users, like Mobile IP, are able to better maintain service continuity, service adaptation, battery life conservation, networks discovery and link discovery. MIH Function also facilitates seamless handovers between heterogeneous networks.

The IEEE 802.21 Event Service has common characteristics with our TRG design but does not prescribe a particular implementation and stops short of allowing upper-layer entities to provide events that can drive a HO. It was also impossible to compare the performance of the implementations since no MIH implementation was available when these tests were performed. Our approach emphasized standardized ways for consumers to receive trigger from a variety of sources. TRG framework is also fully implemented and tested in a laboratory environment with several operating systems. Easy application registration to TRG permits them to get the information they want from different sources. Event generation, on the other hand, is by its very nature a distributed process and, without a central agent, all sources and consumers are forced to create a fully meshed topology. By introducing TRG, event collection becomes straightforward and trigger distribution standardized. That is why we propose that instead of using only the services provided by the IEEE 802.21 MIH functionality future mobile systems should use also TRG alongside 802.21 services. IEEE 802.21 can be, for example, the source entity that provides the lower layer information to TRG.

7. Concluding Remarks and Future Work

This paper presented a novel TRG framework for managing mobility-related triggers and its functionalities for collecting information from various event sources originating not only from the lower layers of the protocol stack (physical, data link, and network), but also from the upper layers and processing the collected events in a standardized trigger format. By using the defined mechanism, TRG framework enables easy and efficient use of cross-layer and cross-domain information. This framework was implemented and evaluated by performing tests in a real environment with several operating systems (Linux, FreeBSD, Windows, Linux Familiar for the PDA and Maemo Linux for the Nokia tablet) to prove its robustness and measure its performance.

The TRG framework experimentations with the performance test and evaluations showed that the implemented TRG functionalities are very promising. TRG can run efficiently in small device with very limited processing power and can enable lossless session handovers between devices. Stress tests showed that the TRG filtering mechanism does not cause delay for processing time and TRG can be used to filter and disseminate large numbers of triggers from several information sources.

Our Triggering management framework is currently integrated with Mobile IP [1] and HIP [2] protocols and is also a part of the Ambient Networks Architecture [15] and prototype as discussed in [16, 17, 18]. TRG and MIP integration with the use of network information a.k.a cascaded triggering presented in [24] showed the benefits of using TRG for the Mobile IP in the case when networks will be congested. HIP integration with TRG and test evaluations presented in [25] showed as well that TRG processing have only a small factor (less than 9%) to the total; trigger collection, processing and dissemination process.

Next steps will be to run a complete test with these integrated mobility protocols in real heterogeneous environments with the WLAN, 3G/HSDPA, WiMAX access technologies. Tests will benefit of the cascaded functionality of TRG when TRG can be located both at the terminal and network side as discussed in [24]. For example, TRG sources at the network side can monitor the network capacity load and other QoS metrics in overlapping networks and based on this information, the network side TRG can send triggers to the terminal initiating or even forcing a vertical handover.

While the tests and evaluations are made in a real testbed environment, a simulation environment will be built to fulfil the tests and analysis of the performance and scalability. In a forthcoming study we will also map the trigger management framework with the recently finalized standard by the IEEE 802.21 working group [12].

References

[1] D. Johnson, C. Perkins, and J. Arkko, Mobility Support in IPv6, ser. *Request for Comments, no. 3775. IETF*, June 2004.

[2] A. Gurtov, Host Identity Protocol (HIP): Towards the Secure Mobile Internet. (Wiley and Sons, June 2008, 328 p).

[3] J. Mäkelä, K. Pentikousis, M. Majanen and J. Huusko, Trigger Management and Mobile Node Cooperation, in M. Katz and F. H.P Fitzek (Eds), *Cognitive Wireless Networks; Concepts, Methodologies and Visions Inspiring the Age of Enlightenment of Wireless Communications*, (Springer-Verlag, 2007) 199-211.

[4] J. Mäkelä and K. Pentikousis, Trigger Management Mechanisms, *Proc. Second International Symposium on Wireless Pervasive Computing,*, San Juan, Puerto Rico, February 2007, 378-383.

[5] P. Prasad, W. Mohr, and W. Konhuser, *Third Generation Mobile Communication Systems*. (Boston: MA: Artech House Publishers, 2005).

[6] J. Eisl (Ed.), Ambient Networks D4.2: Mobility Architecture & Framework, (EU-project IST-2002-507134-AN/WP4/D4.2, 2005).

[7] P. Vidales, J. Baliosian, J.Serrat, G. Mapp, F.Stajano, and A.Hopper, "Autonomic system for mobility support in 4g networks," *IEEE JSAC, 423*(12), 2288–2304.

[8] S. Ishihara, K.Koyama, G.Miyamoto, and M.Kuroda, Predictive rate control for realtime video streaming with network triggered handover*, IEEE WCNC, 3*(13-17), Las Vegas, Nevada, USA, 2005, 1335–1340.

[9] H. Chaouchi and P. Antunes, Pre-handover signalling for QoS aware mobility management, *International Journal of Network Management*, *14*, 2005, 367-374.

[10] E. Casalicchio, V. Cardellini, and S. Tucci, A layer-2 trigger to improve QoS in content and session-oriented mobile services, *Proc. 8th ACM international Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Montral, Quebec, Canada, 2005, 95–102.

[11] C. Kappler, P. Mendes, C. Prehofer, P. Pöyhönen, and D. Zhou, A framework for self-organized network composition, *Lecture Notes in Computer Science, 3457, (*Springer 2005*)*, 139-151.

[12] IEEE Std 802.21™-2008, IEEE Standard for Local and Metropolitan Area Networks—Part 21: Media Independent Handover Services, IEEE, January 2009.

[13] R. Giaffreda, K. Pentikousis, E. Hepworth, R. Agüero, and A. Galis, "An information service infrastructure for Ambient Networks", *Proc. 25th International Conference on Parallel and Distributed Computing and Networks* (PDCN), Innsbruck, Austria, February 2007, pp. 21–27.

[14] OASIS eXtensible Access Control Markup Language, OASIS specification, Available: http://www.oasis-open.org/committees/xacml/

[15] N. Niebert, A. Schieder, J. Zander and R. Hancock (Eds.), *Ambient Networks; Co-operative Mobile Networking for the Wireless World* (Wiley & Sons, 2007)

[16] P. Pääkkönen, P. Salmela, R. Aguero, J. Choque, An Integrated Ambient Networks Prototype, *Proc. SoftCOM 2007*, September 2007, Split, Croatia, 27-29.

[17] C. Simon, R. Rembarz R., P. Pääkkönen, et al., Ambient Networks Integrated Prototype Design and Implementation*, Proc. 16th IST Mobile Summit,* Budapest, Hungary, July 2007, 1-5.

[18] K. Pentikousis, R. Agüero, J. Gebert, J. A. Galache, O. Blume, P Pääkkönen, The Ambient Networks heterogeneous access selection architecture, *Proc. First Ambient Networks Workshop on Mobility, Multiaccess, and Network Management (M2NM)*, Sydney, Australia, October 2007, pp. 49–54.

[19] J. Makela, R. Aguero, J. Tenhunen, V. Kyllönen, J. Choque, andL. Munoz, Paving the Way for Future Mobility Mechanisms: A Testbed for Mobility Triggering & Moving Network Support, *Proc. 2nd International IEEE/Create-Net Tridentcom*, Barcelona, Spain, March 2006.

[20] E. Tuulari and A. Ylisaukko-oja, SoapBox: A Platform for Ubiquitous Computing Research and Applications, *Lecture Notes in Computer Science 2414: Pervasive Computing*, (Zurich, CH: Springer, Aug. 2002) 125–138.

[21] C. Pinho, J. Ruela, K. Pentikousis, and C. Kappler, A protocol for event distribution in next-generation dynamic networks*, Proc. Fourth EURO-NGI Conference on Next Generation Internet Networks (NGI)*, Krakow, Poland, April 2008, pp. 123-130.

[22] C. H. Lwi, H. Mohanty, and R. K. Ghosh, Causal Ordering in Event Notification Service Systems for Mobile Users, *Pro. Int. Conf. Information Technology: Coding and Computing (ITCC)*, *2*, 2004, 735-740.

Conf. Distributed Computing Systems Workshops (ICDCS 02), 2002, 639-644.

[23] H. A. Duran-Limon, G. S. Blair, A. Friday, T. Sivaharan, and G. Samartzidis, A Resource and QoS Management Framework for a Real-time Event System in Mobile Ad Hoc Environments, *Proc. Int. Workshop Object-Oriented Real-Time Dependable Systems (WORDS)*, 2003, 217-224.

[24] M. Luoto and T. Sutinen, Cross-layer Enhanced Mobility Management in Heterogeneous Networks, *Proc. International Conference on Communications*, Beijing, China, May 2008.

[25] P. Pääkkönen, P. Salmela, R. Aguero, J. Choque, Performance Analysis of HIP-based Mobility and Triggering, *Proc. IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, 2008.

PVIII

DISTRIBUTED INFORMATION SERVICE IN OVERLAPPING MULTIACCESS NETWORKS

by

J. Mäkelä, M. Luoto, T. Sutinen and K. Pentikousis 2011

International Journal of Multimedia Tools and Applications, special issue on
mobile media delivery, volume 55, issue 2, 2011, Springer

Multimed Tools Appl
DOI 10.1007/s11042-010-0589-9

Distributed information service architecture for overlapping multiaccess networks

**Jukka Mäkelä · Markus Luoto · Tiia Sutinen ·
Kostas Pentikousis**

Abstract Multimedia delivery in mobile multiaccess network environments has emerged as a key area within the future Internet research domain. When network heterogeneity is coupled with the proliferation of multiaccess capabilities in mobile handheld devices, one can expect many new avenues for developing novel services and applications. New mechanisms for audio/video delivery over multiaccess networks will define the next generation of major distribution technologies, but will require significantly more information to operate according to their best potential. In this paper we present and evaluate a distributed information service, which can enhance media delivery over such multiaccess networks. We describe the proposed information service, which is built upon the new distributed control and management framework (DCMF) and the mobility management triggering functionality (TRG). We use a testbed which includes 3G/HSPA, WLAN and WiMAX network accesses to evaluate our proposed architecture and present results that demonstrate its value in enhancing video delivery and minimizing service disruption in an involved scenario.

Keywords Mobility management · Multiaccess · Event service · Information service

---

J. Mäkelä (✉) · M. Luoto · T. Sutinen
VTT Technical Research Centre of Finland,
Kaitovayla 1, PO Box 1100, 90571 Oulu, Finland
e-mail: jukka.makela@vtt.fi

M. Luoto
e-mail: markus.luoto@vtt.fi

T. Sutinen
e-mail: tiia.sutinen@vtt.fi

K. Pentikousis
Huawei Technologies European Research Center,
Carnotstrasse 4, 10587 Berlin, Germany
e-mail: k.pentikousis@huawei.com

1 Introduction

Heterogeneous network environments and multiaccess connectivity evidently create new opportunities, but also pose challenges for applications. Especially for Quality of Service (QoS) sensitive multimedia applications, it is difficult to fully utilize the potential of heterogeneous multiaccess networks as host mobility (and the necessary handover management that this leads to) undermine QoS. As the underlying network connection characteristics vary, major effects need to be handled in order to maintain a high-level user experience, for example, in video quality delivery. Currently, it is difficult for applications to take sufficient corrective measures (such as adaptation) and cope with the handover-induced QoS changes as the current layer 2 and 3 (L2/L3) mobility management solutions aim at hiding mobility effect completely from higher layers. Moreover, in many cases, the handover decision-making of current mobility protocols is suboptimal for multimedia applications as the decisions are solely based on lower layer information (e.g. RSS measurements) and do not take into account the applications' requirements for network access.

Mobility management in a heterogeneous network needs support from mobility management protocols such as Mobile IP [8] and Host Identity Protocol [6]. These protocols provide key mobility management mechanisms, i.e. handover management that enables changing the point of network attachment. The change of the point of attachment is usually based on a decision made by the mobility management protocols or by a separate entity guiding the protocols, as discussed in [12]. This decision should take into account the conditions prevailing in the terminal and/or the network side. More specifically, the decision is triggered by an event or events created within the system as a response to a change in the conditions. Examples of such changes include received signal strength, access network availability or user preferences; all these can define events which can lead to handover decisions.

Since conditions are changing both at the terminal and the networks side, distributed information exchange and decision-making is necessary for handover management. To enable these distributed actions, we propose a framework to exchange information between various entities that may be located at different layers of the protocol stack and in different network nodes. More specifically we use a distributed information service for collecting and distributing events created by condition changes in the different parts of the system. The Distributed Control and Management Framework (DCMF) with Trigger Management (TRG) proposed in this paper provide the required cross-layer signaling mechanisms to enable application-aware handover decision-making and enhanced application adaptation in the presence of heterogeneous handovers. First of all, a multimedia application is capable of informing the handover manager about its QoS requirements (bandwidth, delay, etc.). This information can be collected, for example, from application-layer session descriptors (Session Description Protocol) and communicated to the handover manager via TRG. The handover manager can then map this information to that received from lower layers (e.g. through Media Independent Handover (MIH) Service [15]) and from the network to select the best possible access for multimedia transmission.

In this paper, we put emphasis on evaluating and validating the proposed framework implementation in a state-of-the-art testbed, demonstrating the performance benefits achieved in a mobile multiaccess test case. We also present for the first time all main components that are mandatory for collecting and evaluating the information and what is more important to enable a distributed way of disseminating the information with our system. We use real heterogeneous multiaccess testbed environment with 3G/HSPA, WLAN and WiMAX access technologies. The benefits obtained from using our information service are validated in the context of changing the point of attachment between the different access networks of the testbed environment.

The rest of the paper is structured as follows. Section 2 describes the distributed information service architecture with the main concepts. Section 3 describes the experimental study scenario and heterogeneous testbed and shows the evaluation cases with results. Section 4 reviews relevant related work and, finally, Section 5 concludes the paper.

2 Distributed information service

Figure 1 illustrates the Distributed Control and Management Framework (DCMF). The illustration also shows examples of the software modules and functionalities that are typically located at mobile terminals (left-hand side of Fig. 1) and in core network nodes (right-hand side of Fig. 1). DCMF is designed to support distributed information exchange between operators, enable advanced access control, and allow for different optimization techniques such as balancing traffic load between access points in the network architecture. The basis of this framework is the TRG functionality that provides a unified signaling architecture between network nodes, capable of serving different entities, located either in the network or in the terminals and controlled by different players (terminals by the user, network by the service provider). The network side elements (right-hand side of Fig. 1) that will benefit from such a signaling framework include, for example, network/operator aided mobility management and resource management. On the terminal side (left-hand side of Fig. 1) mechanisms such as mobility management or transport protocol optimization solutions need access to extensive amount of information related to network access characteristics and roaming.

The information that is made available through DCMF, by using TRG for signaling between nodes, can be used for optimizing network and terminal side operations in terms of application QoS requirements, energy efficiency, security, and network load balancing. In order to obtain relevant information for the decision-making process, we need to rely on, for example, operator policies, QoS measurements, and various cross-layer events (e.g. those defined and grouped in [14]).

The main components of the DCMF framework are TRG, Network Expert System (NES), Mobile Expert System (MES), and network resource probes. The architectural view of these components is shown in Fig. 2.

The role of TRG in DCMF framework is to act as an interface between the information sources and the information consumers. TRG provides also an information dissemination mechanism between core network and mobile devices through the wireless access network. Information dissemination is can capitalize on the cascaded functionality of TRG. In this case, one TRG network entity collects information from the network and acts as an information source to the second TRG entity located in the mobile node. Other network side components are the Network Expert

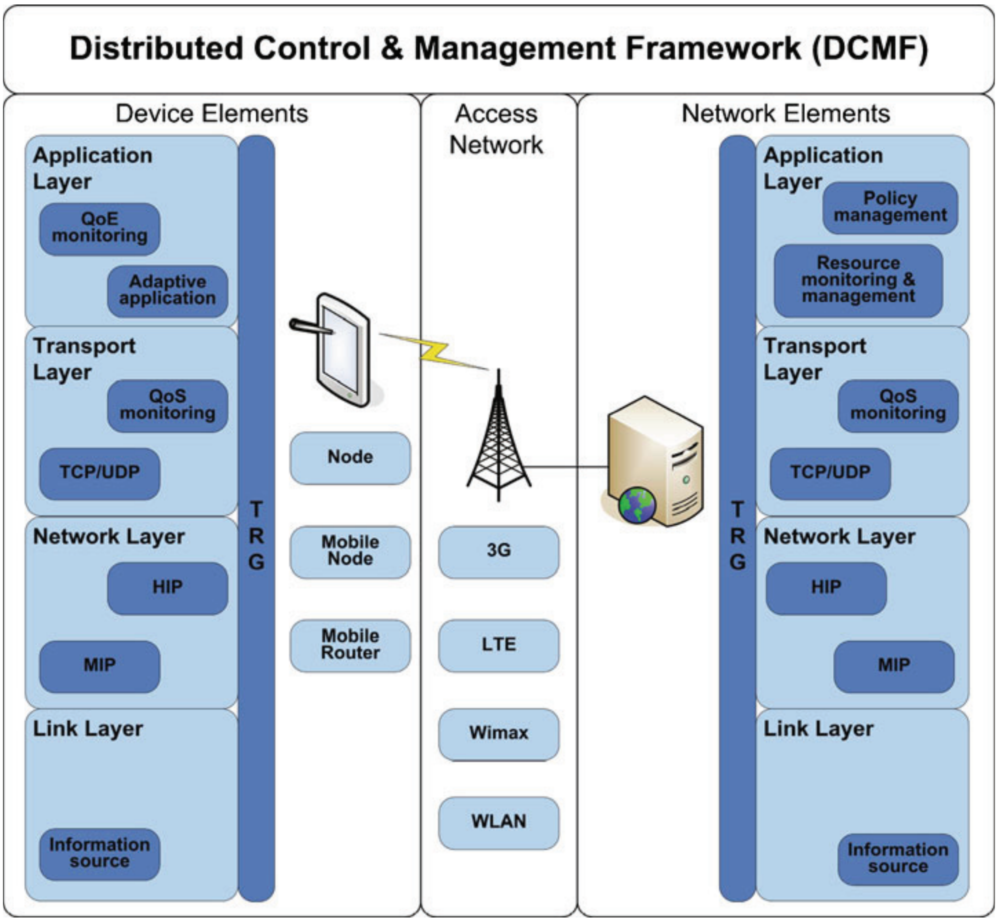


Fig. 1 The Distributed Control and Management Framework (DCMF)

System (NES) and network resource probe(s). Network resource probes are the components that are collecting available information from the network side. A network probe can, for example, collect valuable information from WLAN access points using the Simple Network Management Protocol (SNMP). This information is further processed by NES, which is a Self-organized Map (SOM) based decision-making component. NES is capable of analyzing the level of congestion in the access network. Human experts can guide the decisions of NES by using different training data and by defining different sets of rules and policies to guide decision-making. This processed information is further collected by the network-side TRG which provides the information (in the form of triggers) to the TRG residing on the mobile node. Communication with TRG uses reliable Web services or plain TCP sockets depending on the source or the consumer. The communication between TRGs is always Web service based. Figure 3 shows the messages in a simple example case where congestion information is gathered from WLAN access point to the mobile terminal.

By using this approach it is possible to minimize the access network traffic as well since only the necessary information is transferred to the mobile node over the

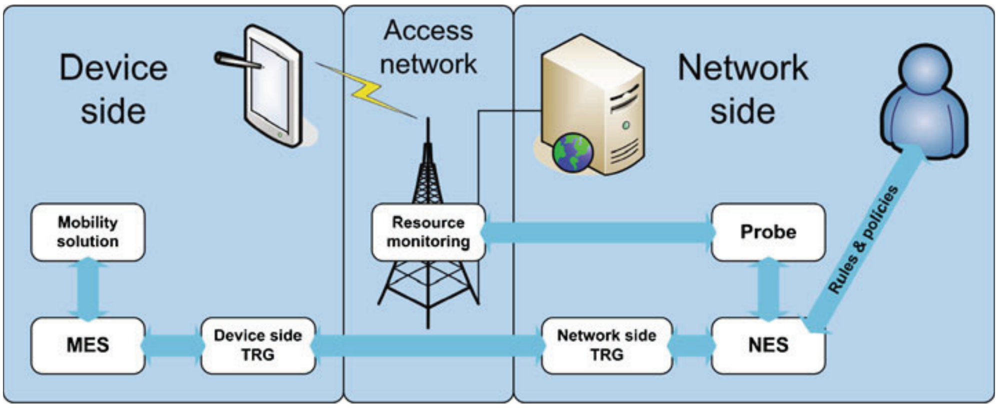


Fig. 2 Main components of Distributed Control and Management Framework (DCMF)

wireless network (message 5: in Fig. 3). The information can be further analyzed by the Mobile Expert System (MES) based on the conditions on mobile node. The main idea for this approach is that each node and network knows best its own conditions, characteristics, and resources and can help other entities to make their own decision as well.

We have implemented and tested the main components of DCMF and for this paper we built a heterogeneous network environment to test our implementation and show the feasibility of our proposed architecture and evaluate its performance in a heterogeneous network environment. Next we delve into more details about the design of DCMF components.

2.1 Trigger management

At the core of DCMF lies TRG (which is detailed in [14]). TRG is capable of collecting, storing temporarily, and delivering notifications, called triggers, within an

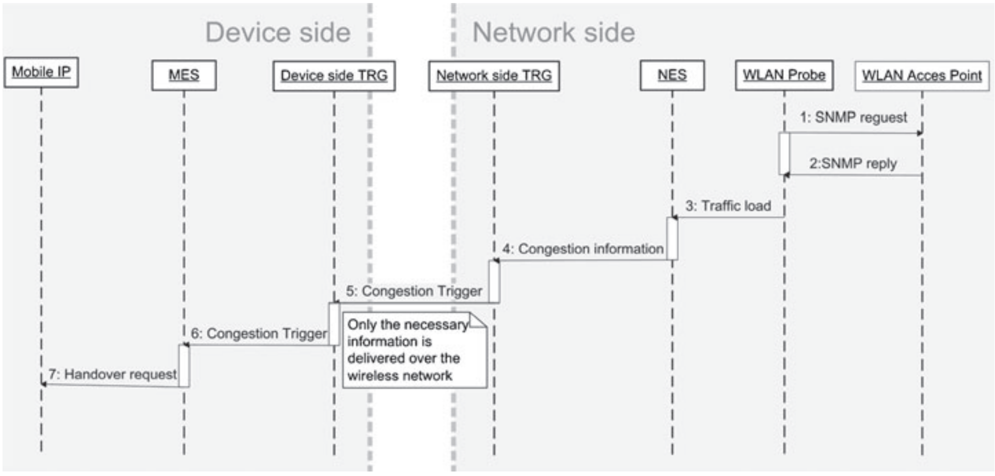


Fig. 3 Message flow between the components

IP-based communications system. Triggers signal a change in the system status and may originate from virtually any entity that influences node communication. This is in contrast with IEEE 802.21 [15] (and other proposals based on this emerging standard), in which events originate solely from the lower layers. On the other hand, similarly to IEEE 802.21, TRG leaves the interpretation of the information carried in the triggers to trigger consumers, which can use the delivered notifications as input to their own decision-making processes.

We are interested in developing a scalable and distributed system that uses TRG to deliver a distributed information service between different network nodes. We recently demonstrated the feasibility of our proposal using several TRG entities for delivering triggers from WLAN access points to a Mobile IP client [10]. TRG entities, located at various network locations and at the mobile nodes, can exchange information and collaborate in order to increase the performance of multiaccess mobile devices. TRG can capitalize on the availability of IEEE 802.21 Media Independent Handover Functions (MIHF) in the network, as both remote and local MIH events can be converted into triggers. As discussed in [5], the mechanisms and services introduced by the recently finalized IEEE standard are not sufficient for handover triggering and decision processes. IEEE 802.21 provides a way to command and use the lower layer information to enable seamless handovers and multiaccess, but there is no way to capitalize also on upper layer information such as user or application preferences. Our trigger management framework enables a two-way exchange of information allowing the mobility management entities to consider both changes at the upper layers and at the lower layers. That is why we propose that instead of using only the services provided by IEEE 802.21, future mobile devices should use the TRG framework together with 802.21 services. MIHFs can be, for example, source entities that provide lower layer information according to the IEEE 802.21 standard to TRG.

Most of the current application adaptation solutions rely on client feedback to make dynamic adaptation decisions (e.g. adjustments to bit rate, frame rate or resolution in the case of a video streaming application) in the server or some intermediate media-aware network node. The feedback for RTP/UDP-based multimedia applications is typically realized using RTCP [21]. Through TRG and DCMF, an adaptive application can gain access to more extensive information (e.g. MIH events, MIP or HIP mobility state changes) that it can use in adapting its operation more efficiently in a multiaccess environment. TRG entities can potentially be employed also in end-to-end feedback delivery, in cases where traditional feedback protocols do not suffice. For example [18] describes an architecture for such an application.

TRG can feed the decision-making entities, namely the Network Expert System (NES) and the Mobile Expert System (MES) in our system, or, for example, vertical handover control systems, which uses several inputs for the decisions as presented in [12]. The decision on which interface to use can be defined as a multiple attribute decision making problem and it can be based on different algorithms like Simple Additive Weighting: $argmax_{i \epsilon M} \sum_{j=1}^{n} w_j r_{ij}$ and Multiplicative Exponent Weighting: $\prod_{j=1}^{N} x_{ij}^{w_j}$ as detailed in [22, 25]. These are just a couple of examples that could be used. Note that decision making algorithms per se are out of the scope of this paper. Other adaptive procedures like genetic algorithms for solving the decision problems are also possible as the design of the decision mechanism allows the flexible use of different algorithms.

Table 1 Trigger format

Field	Type	Description
`id`	integer	Trigger identifier, same as producer identifier.Maps producer name to identifier
`type`	integer	Specific to the trigger identifier. Mapping producer information to type
`value`	std:string	Specific to trigger type
`timestamp`	time_t	Time that a trigger enters the TRG repository

2.2 Event collection and distribution

The TRG functional entity has three major modules: event collection, event processing, and the trigger repository. Event collection receives events from triggering event sources via collection interface. Triggers can be created by implementing the trigger event collection interface. The interface allows sources to register their triggers which makes them available to consumers. Implementations of TRG may have several event collectors which are responsible for collecting different types of events. The trigger repository is designed to temporarily store the triggers. It is designed to meet the requirements of mobility management, but can be used to store non-mobility triggers as well. The repository's basic primitives include adding, removing, replacing, and disseminating triggers in a predefined format (Table 1). The format consists of three fields, namely id, type and value. The id field is used identify the trigger also identifying the producer. The type field defines the type of the trigger if the same producers provides several types of triggers. The value field carries additional information provided by the trigger.

The origin of an event source can be a hardware device, a system component implemented in kernel space, or an application implemented in user space. For example, each device driver could implement its own event collection functionality, which would be capable of handling triggering events produced by the specific device only. Event sources can also be located in the network such as at active network elements or at the user's home network e.g. in the Mobile IP [8] home agent. TRG entity can act also as a consumer or a producer to another TRG (cascaded TRG functionality) located in a different node in the network. Thus, orchestrating the collaboration of several collection entities is needed in order to efficiently gather a larger amount of events.

Having dedicated collectors for different event sources enables the use of TRG in different operating systems as well. The collector can format the events to the format (Table 1) that TRG understands and there is no need to modify the core TRG implementation; instead the collector can be modified as necessary. This is also one of the key points in the architectural design of TRG that enables it to handle cross-layer information by having collectors at different layers as needed. For example TRG can get similar information regarding the connectivity in FreeBSD through a collector that uses Route Socket and in Linux through a similar collector using RTnetlink socket.

After events are collected from the producers, they are handed over to the trigger processing engine which is responsible for time-stamping and reformatting triggers (if necessary), and assigning them to the appropriate group [14]. Those entities called consumers which are interested in certain triggering events can subscribe by specifying a set of triggers (and, optionally, filtering rules) and are expected to unsubscribe when they do not wish to receive them any longer. For each consumer subscription, TRG ensures that filters are grammatically and syntactically correct, and accepts or rejects the subscription. Basic rules can also be used as building blocks for crafting more sophisticated rules. More details about the TRG architecture, functionality and the performance evaluation can be found in [14].

3 Evaluation

Figure 4 illustrates our testbed and a scenario for evaluating and validating DCMF operation . With the testbed we can also measure and demonstrate the performance benefits achieved by our architecture in a mobile multiaccess environment. The scenario, described next, and can be run on the testbed, includes a mobile router (MR) providing access to a mobile client. In the beginning of the scenario MR is connected to a WIMAX network. In the next step the MR switches its connection to WLAN1 to save energy. While connected to WLAN1, MR moves away from the WLAN1 access point and as the WLAN1 signal weakens below a threshold, MR makes a handover to WLAN2. While in WLAN2, MR moves out of coverage of both the WIMAX and WLAN1 and as WLAN2 gets suddenly congested the MR makes a handover to the 3G network. During the scenario the client connected to the MR continuously streams a video from an Internet service.

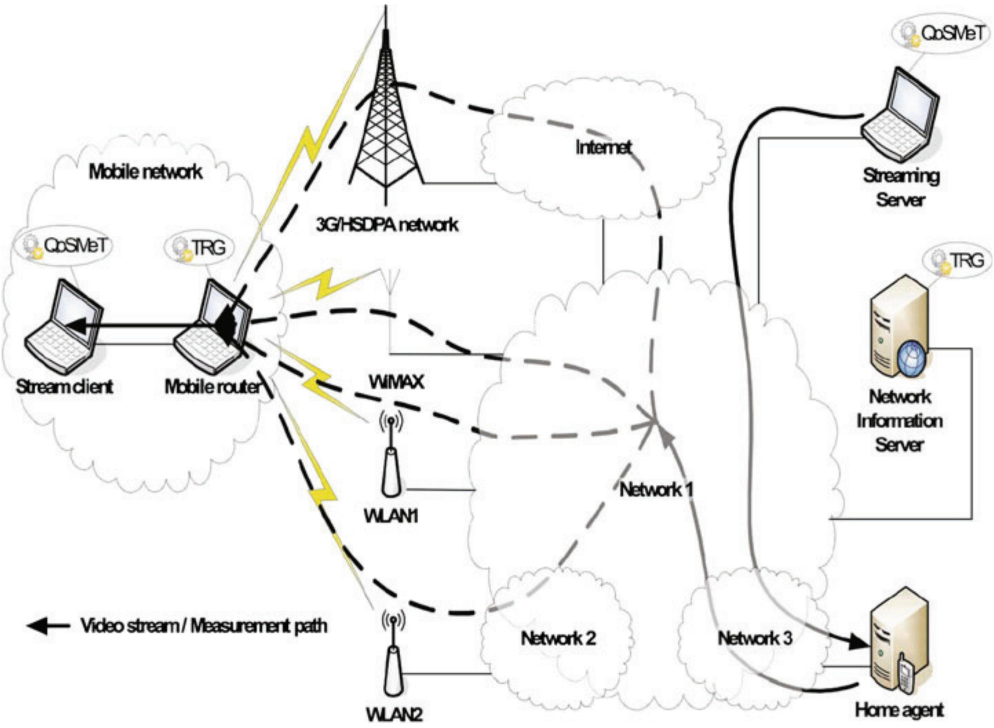


Fig. 4 Illustration of the heteregoneous network testbed and the DCMF evaluation scenario

Table 2 Triggers provided by DCMF

Name	Description
INTERFACE_ADD	Network interface added to the local machine
INTERFACE_REM	Network interface removed from the local machine
INTERFACE_CONN	Local network interface gained IP connectivity
INTERFACE_DISC	Local network interface lost IP connectivity
ROUTE_ADD	Route added to the local routing table
ROUTE_DEL	Route removed from the local routing table
WLAN_OPER_REM	WLAN access point operational status change
WLAN_CONG_REM	WLAN access point congestion status change
WLAN_QUAL_REM	WLAN radio interface quality status change

3.1 Testbed description

The testbed includes a 3G/UMTS cell, two WLAN access points, and one WiMAX cell (see Fig. 4). Using this testbed we show how DCMF can significantly improve the end-user experience. The configuration consists of three network servers, four access networks to which the mobile router is connected to, and a client connected to the mobile router (MR). For mobility management, MR uses Mobile IP (MIP) enhanced with DCMF and acts as router to the mobile network. DCMF provides the triggers detailed in Table 2 to the MIP client running on the MR. These triggers are provided locally on every network interface addition or removal, interface IP connectivity change, and routing table change. DCMF provides also remote triggers on WLAN access point operational or congestion status change and radio interface quality change.

The decision logic added to the MIP client uses the triggers listed in Table 2 to evaluate all available networks and steer the mobile router to choose the best possible network.

In our tests we employ the QoSMeT measurement tool [19], which is running on the streaming server and the client. QoSMeT can measure the one-way QoS characteristics experienced by the video stream from the server to the client. QoSMeT is a QoS measurement tool developed by VTT which uses GPS synchronization to accurately measure one-way QoS characteristics such as packet delay or jitter of an end-to-end link. The video used in the experiment is streamed over UDP (with an MTU of 1358 bytes) and an average bit rate of 485 kb/s.

3.2 Results

Figure 5 illustrates three aspects. First, the delay experienced by the video stream in our testbed with help of DCMF; second the network which is actively used for streaming the video to the client; and third the point at which a vertical handover occurs. MR is initially connected to the WiMAX network (A). The WiMAX network is then disconnected and the MR makes a handover to WLAN1 network (point B in Fig. 5). When the WLAN1 network is congested (by adding UDP traffic with a bit rate of 60 Mb/s to another node in WLAN1) the MR makes a handover to WLAN2 (point C). Finally, the WLAN2 network's transmission power is lowered gradually until the MR makes a handover to the 3G network (point D).

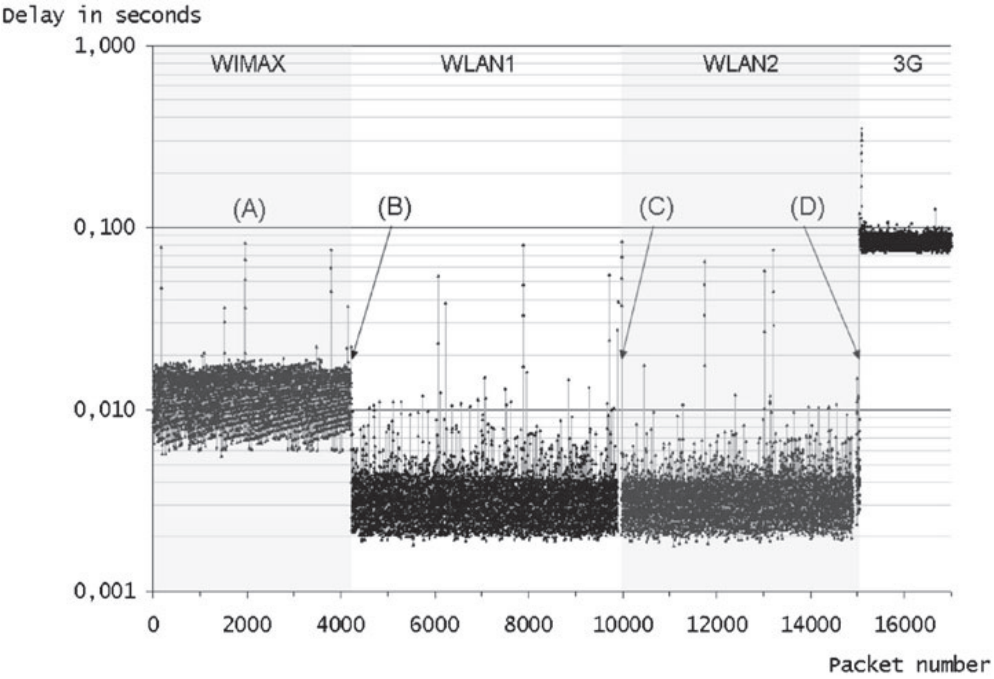


Fig. 5 One-way packet delay and handovers in our testbed evaluation

Fig. 6 Signal strength measurement

Although a network card is able to maintain connection with an access point, packets can still be lost due to low signal strength. To address this, DCMF provides triggers based on the wireless network adapter signal strength. Figure 6 illustrates a signal strength measurement by an information source in the DCMF. When the signal strength drops below the lower threshold set in the information source, the source sends a trigger informing interested parties that the interface in question is considered to be down due to low signal strength. Again when the signal strength rises above the upper threshold a trigger indicating the interface is up is sent. The interface down trigger based on the signal strength triggers the handover (D) in Fig. 5.

The MR by itself can only assess the quality of a WLAN connection mainly based on received signal strength. In our architecture, DCMF can provide additional information from the network to the MR via TRG. The handover (C) in Fig. 5 is a result of such information. At that point in our scenario WLAN1 is heavily congested with additional traffic from another client in the network. The WLAN probe in the network information server (NIS) notices this and sends a trigger to TRG running in the NIS which forwards the trigger to the TRG in the MR which in turn sends it to the decision logic guiding the MIP client. The decision logic instructs the MIP client to switch to the WLAN2.

To better illustrate the advantage of DCMF, we now focus on the handover (C) in Fig. 5. The packet delay in this situation is graphed in more detail in Fig. 7 and as a reference without the help of DCMF in the same situation in Fig. 8. In Fig. 7 with the help of DCMF the packet delay remains constant and only 90 packets are lost while the MR makes a handover to the WLAN2. This packet loss is caused by the

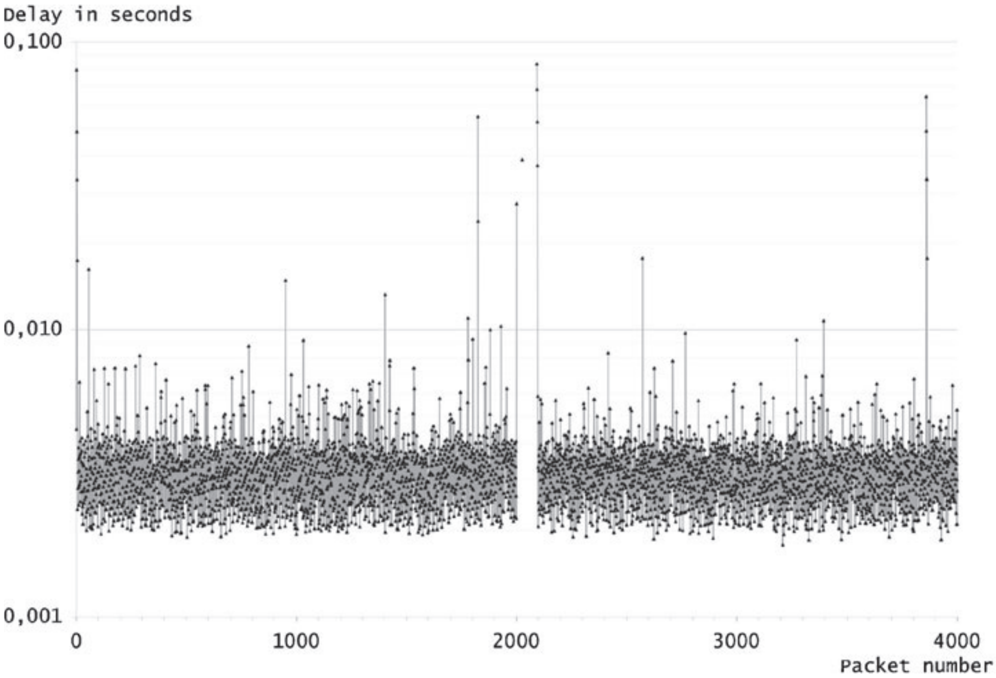


Fig. 7 Network congestion in WLAN1 in case of DCMF assistance

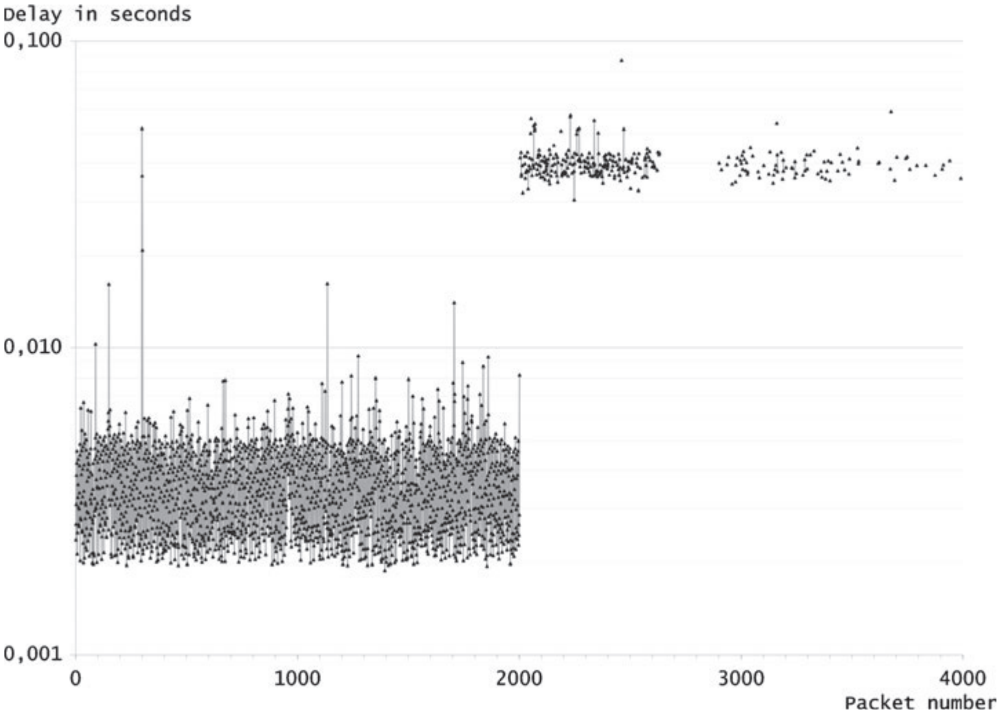


Fig. 8 Network congestion in WLAN1 without the assistance from DCMF

delay between the start of the congestion and the initiation of the handover. Visually a casual viewer experiences a small glitch in the video just before the handover is made. Figure 8 shows the loss of 1703 packets and no handover at all is made by the MR during this time. Eventually, after 6238 lost packets, the MR makes a handover which is caused by the MR not being able to update its binding with its HA because of the congestion. Visually this corresponds to 50 s of the video being unintelligible during which time only occasional broken frames of the video can be displayed to a person viewing the video.

The Table 3 shows a classification of packets in Figs. 7 and 8 based on the delay and also the number of packets lost. As can be seen from the table, DCMF enhanced system offers better QoS in terms of packet loss, the packet loss being 2.2% with DCMF compared to 42.6% without DCMF. Also during the congestion, the delivered packets without DCMF have significantly longer delay which can be seen in the number of packets with delay over 0,01 sec.

The measurements without the help of DCMF were made using the default settings of the MIP client. In this case the most relevant setting being the interval at which the MR renews its binding with the Mobile IP Home Agent (HA). The default

Table 3 Packet delay distribution (sec)

	Delay < 0,004	0,004 < Delay < 0,01	Delay > 0,01	Packets lost
DCMF	3,369 (84.2%)	518 (12.9%)	24 (0.6%)	90 (2.2%)
No DCMF	1,303 (32.6%)	690 (17.2%)	305 (7.6%)	1,703 (42.6%)

of this setting on the used MIP client is 60 s. This means that the MR's network connection will be down 30 s on average in case of a total connection loss or severe network congestion as in the example. This problem can be diminished by lowering the binding renewal interval but this also puts unnecessary strain on the network as the communication between the MR and the HA increases. Still, with less severe congestion it is possible for the MR to experience lowered QoS and packet losses indefinitely long periods of time as long as the MIP client is able to renew its binding with HA.

Test case evaluations, as seen from the figures, showed that there is a clear benefit when using DCMF to assist mobility management protocols. This evaluation case and the used implementations are also steps towards the media independent and seamless handovers in the mobile wireless world.

4 Related work

Using the event information service described in this paper allows one to proactively perform a handover in order to maintain QoS levels and maintain application/session continuity during handover processes. We also believe that in heterogeneous network environments several sources of events and context information should be consulted in order to achieve seamless connectivity and develop swift mobility management mechanisms like those presented in [23]. Event/notification systems presented in [4, 16], which introduce mechanisms on how to implement such systems, along with the evaluated event generation cases are very encouraging and complementary to our effort in defining TRG together with DCMF as a specialized notification system for events which originate from the entire protocol stack. Event and notification system like the one presented in this paper are needed also for the future Internet solutions as discussed in [17].

As mentioned earlier, the TRG functionality approach is related to the IEEE 802.21 [15] approach. The scope of the IEEE 802.21 standard is to develop a mechanism that provides link layer intelligence and other related network information to upper layers so that they can optimize handovers between heterogeneous IEEE 802 systems and facilitate handovers between IEEE 802 and cellular systems. IEEE 802.21 helps with Handover Initiation, Network Selection and Interface Activation. The purpose is to enhance the experience of mobile device users. The standard supports handovers for both stationary and mobile users. TRG, on the other hand, is designed to provide the way to command and use information from all layers, even from the physical radio interface. The main target is to define and implement a distributed framework as discussed in Section 2 that supports event collection and processing, and trigger distribution possibly from hundreds of different sources. This is also why we propose that instead of using only IEEE 802.21 services the system can integrate IEEE 802.21 services with TRG functionality as well e.g. TRG can be a distributor of MIH events to all application layer producers.

The generic nature of TRG allows its usage for a multitude of signaling purposes. Although the experimental evaluation of DCMF described in this paper focused on enhancing video delivery by the means of DCMF-aided handover management, the natural next step will be to utilize the proposed information service in enhancing application adaptation in heterogeneous multi-access environments. The usage of cross-layer information in adapting QoS-sensitive multimedia services especially to quickly-changing wireless network conditions has been an area of ongoing research for many years now. For example, adaptive video transmission can be enhanced with channel state information to be more responsive to changes in wireless link conditions as explained in [7].

In heterogeneous multiaccess environments, multimedia applications need to implement adaptive schemes beyond the traditional end-to-end RTCP feedback-based algorithms because mobility events such as vertical handovers can cause sudden and drastic changes in the characteristics of the network connection. Event information originating from mobility management protocols and MIH can be used in triggering application adaptation in this case, as discussed in [1]. As this kind of information is not included in traditional multimedia signaling protocols such as RTCP, the TRG-based notification system is viable also for this purpose, as proposed in [18]. Moreover, TRG and DCMF scale well for the needs of a full end-to-end multimedia adaptation solution that makes use of distributed adaptation in different parts of the end-to-end path: in the server, in the client, and in an intermediate media-aware network node (MANE) that can be implemented for example in a WLAN access router [9]. The difference is that with TRG signaling, MANE does not necessarily need to be an application-level proxy relying on RTP/RTSP signaling only but also more light-weight processing of the streams, such as link level scheduling [20], can be supported.

One important issue, which several papers, see for example [3, 11, 24] focus on, are temporal event ordering and other time-related issues. Since first of all older event should not overrun newer or current valid event information. Design and realization of the TRG functionality is flexible in a way that it supports dedicated mechanisms to handle the prioritization of events or ensuring the trust between different entities, namely consumers and producers. In addition to that, TRG has a mechanism for time stamping all events as well as handling the registration of producers. TRG also provides a mechanism to filter all unwanted events before disseminating them to the consumers as discussed in [14]. The current implementation provides also a way to have policies to prevent the access to certain event information.

5 Conclusion

We presented the distributed information service and detailed its main building blocks: Distributed Control and Management Framework (DCMF) and trigger management (TRG). Our information service makes for an efficient way of exchange and use cross-layer information in heterogeneous multiaccess environments. This paper concentrated on showing the performance benefits gained from using a network aided information service to assist handovers in a heterogeneous network. Of course, although our main target is to provide an efficient way to handle distributed decision making and information flows in heterogeneous networks and for multiaccess terminals, our framework can guide mobile terminals within a network consisting of only one access technology as well.

Triggering functionality and network monitoring components were used to build the presented distributed information service system. For testing and evaluation purposes we built a real heterogeneous testbed environment with 3G/HSDPA, 802.11/WLAN and 802.16/WiMAX access technologies. With this testbed we were able to validate that our implementation is already suitable for a real network environment. Test result showed that the there are real benefits gained with this system: in terms of packet loss and one-way delay a DCMF-enabled system can achieve an order of magnitude increase in performance during congestion-induced handovers. Our work continues by making more tests to validate and evaluate the components used for information gathering and decision making in different network nodes.

References

1. Cacae F, Vollero L (2006) Managing mobility and adaptation in upcoming 802.21-enabled devices. In: Proceedings of WMASH'06, 29 Sept 2006, Los Angeles, CA, USA
2. Daniel L, Kojo M (2008) Employing cross-layer assisted TCP algorithms to improve TCP performance with vertical handoffs. Int J Comm Network Distr Syst 1(4/5/6):433–465
3. Dousson C, Pentikousis K, Sutinen T, Makela J (2007) Chronicle recognition for mobility management triggers, computers and communications. In: Proceedings of IEEE symposium on computers and communications (ISCC'07), 1–4 July 2007, pp 305–310
4. Duran-Limon HA, Blair GS, Friday A, Sivaharan T, Samartzidis G (2003) A resource and QoS management framework for a real-time event system in mobile ad hoc environments. In: Int. workshop object-oriented real-time dependable systems (WORDS)
5. Giaffreda R, Pentikousis K, Hepworth E, Aguero R, Galis A (2007) An information service infrastructure for ambient networks. In: Proceedings of the 25th international conference on parallel and distributed computing and networks, Innsbruck, Austria, pp 21–27
6. Gurtov A (2008) Host identity protocol (HIP): towards the secure mobile internet. Wiley and Sons, Hardcover, 328 p. ISBN 978-0-470-99790-1
7. Huusko J, Vehkaperä J, Amon P, Lamy-Bergot C, Panza G, Peltola J, Martini MG (2007) Cross-layer architecture for scalable video transmission in wireless network. Signal Process Image Commun 22:317–330
8. Johnson D, Perkins C, Arkko J (2004) Mobility support in IPv6. Ser. Request for Comments, no. 3775. IETF
9. Kofler I, Prangl M, Kuschnig R, Hellwagner H (2008) An H.264/SVC-based adaptation proxy on a WiFi router. In: Proceedings of the 18th international workshop on network and operating systems support for digital audio and video (NOSSDAV'08), 28–30 May 2008. Braunschweig, Germany
10. Luoto M, Sutinen T (2008) Cross-layer enhanced mobility management in heterogeneous networks. In: Proceedings of the international conference on communications, Beijing, China
11. Lwi CH, Mohanty H, Ghosh RK (2004) Causal ordering in event notification service systems for mobile users. In: Int. conf. information technology: coding and computing (ITCC)
12. Mäkelä J (2007) Towards seamless mobility support with cross-layer triggering. In: Proceedings of personal, indoor and mobile radio communications, 2007. PIMRC
13. Mäkelä J, Luoto M, Sutinen T, Pentikousis K (2009) Evaluation of an information service for enhanced multiacces media delivery. In: Proceeding of 3rd European symposium on mobile media delivery (EUMOB)
14. Mäkelä J, Pentikousis K, Kyllönen V (2009) Mobility trigger management: implementation and evaluation. Int'l J of Communications, Network and System Sciences 2(3):211–221
15. Media Independent Handover. IEEE Standard 802.21
16. Meier R, Cahill V (2002) STEAM: event-based middleware for wireless ad hoc networks. In: IEEE int. conf. distributed computing systems (ICDCS) workshops
17. Mingardi C, Nunzi G, Dudkowski D, Brunner M (2009) Event handling in clean-slate future internet management. In: Proceedings of the 11th IFIP/IEEE international symposium on integrated network management, IM
18. Piri E, Sutinen T, Vehkaperä J (2009) Cross-layer architecture for adaptive realtime multimedia in heterogeneous network environment. In: European wireless

19. Prokkola J, Hanski M, Jurvansuu M, Immonen M (2007) Measuring WCDMA and HSDPA delay characteristics with QoSMeT. In: Proceedings of IEEE international conference on communications
20. Schierl T, Stockhammer T, Wiegand T (2007) Mobile video transmission using scalable video coding. IEEE Trans Circuits Syst Video Technol 17(9):1204–1217
21. Schulzrinne H, Casner S, Frederick R, Jacobson V (2003) RTP: a transport protocol for real-time applications. Ser. Request for Comments, no. 3550. IETF
22. Stevens-Navarro E, Wong VWS (2006) Comparison between vertical handoff decision algorithms for heterogeneous wireless networks. In: Vehicular technology conference, VTC 2006–Spring
23. Vidales P, Baliosian J, Serrat J, Mapp G, Stajano F, Hopper A (2005) Autonomic system for mobility support in 4g networks. IEEE JSAC 23(12):2288–2304
24. Yoneki E, Bacon J (2005) Determination of time and order for eventbased middleware in mobile peer-to-peer environments. In: IEEE int. conf. pervasive computing and communications (PerCom) workshops
25. Zhang W (2004) Handover decision using fuzzy MADM in heterogeneous networks. In: Wireless communications and networking conference, WCNC

JYVÄSKYLÄ STUDIES IN COMPUTING

1 Ropponen, Janne, Software risk management - foundations, principles and empirical findings. 273 p. Yhteenveto 1 p. 1999.

2 Kuzmin, Dmitri, Numerical simulation of reactive bubbly flows. 110 p. Yhteenveto 1 p. 1999.

3 Karsten, Helena, Weaving tapestry: collaborative information technology and organisational change. 266 p. Yhteenveto 3 p. 2000.

4 Koskinen, Jussi, Automated transient hypertext support for software maintenance. 98 p. (250 p.) Yhteenveto 1 p. 2000.

5 Ristaniemi, Tapani, Synchronization and blind signal processing in CDMA systems. - Synkronointi ja sokea signaalinkäsittely CDMA järjestelmässä. 112 p. Yhteenveto 1 p. 2000.

6 Laitinen, Mika, Mathematical modelling of conductive-radiative heat transfer. 20 p. (108 p.) Yhteenveto 1 p. 2000.

7 Koskinen, Minna, Process metamodelling. Conceptual foundations and application. 213 p. Yhteenveto 1 p. 2000.

8 Smolianski, Anton, Numerical modeling of two-fluid interfacial flows. 109 p. Yhteenveto 1 p. 2001.

9 Nahar, Nazmun, Information technology supported technology transfer process. A multi-site case study of high-tech enterprises. 377 p. Yhteenveto 3 p. 2001.

10 Fomin, Vladislav V., The process of standard making. The case of cellular mobile telephony. - Standardin kehittämisen prosessi. Tapaustutkimus solukkoverkkoon perustuvasta matkapuhelintekniikasta. 107 p. (208 p.) Yhteenveto 1 p. 2001.

11 Päivärinta, Tero, A genre-based approach to developing electronic document management in the organization. 190 p. Yhteenveto 1 p. 2001.

12 Häkkinen, Erkki, Design, implementation and evaluation of neural data analysis environment. 229 p. Yhteenveto 1 p. 2001.

13 Hirvonen, Kullervo, Towards better employment using adaptive control of labour costs of an enterprise. 118 p. Yhteenveto 4 p. 2001.

14 Majava, Kirsi, Optimization-based techniques for image restoration. 27 p. (142 p.) Yhteenveto 1 p. 2001.

15 Saarinen, Kari, Near infra-red measurement based control system for thermo-mechanical refiners. 84 p. (186 p.) Yhteenveto 1 p. 2001.

16 Forsell, Marko, Improving component reuse in software development. 169 p. Yhteenveto 1 p. 2002.

17 Virtanen, Pauli, Neuro-fuzzy expert systems in financial and control engineering. 245 p. Yhteenveto 1 p. 2002.

18 Kovalainen, Mikko, Computer mediated organizational memory for process control. Moving CSCW research from an idea to a product. 57 p. (146 p.) Yhteenveto 4 p. 2002.

19 Hämäläinen, Timo, Broadband network quality of service and pricing. 140 p. Yhteenveto 1 p. 2002.

20 Martikainen, Janne, Efficient solvers for discretized elliptic vector-valued problems. 25 p. (109 p.) Yhteenveto 1 p. 2002.

21 Mursu, Anja, Information systems development in developing countries. Risk management and sustainability analysis in Nigerian software companies. 296 p. Yhteenveto 3 p. 2002.

22 Seleznyov, Alexandr, An anomaly intrusion detection system based on intelligent user recognition. 186 p. Yhteenveto 3 p. 2002.

23 Lensu, Anssi, Computationally intelligent methods for qualitative data analysis. 57 p. (180 p.) Yhteenveto 1 p. 2002.

24 Ryabov, Vladimir, Handling imperfect temporal relations. 75 p. (145 p.) Yhteenveto 2 p. 2002.

25 Tsymbal, Alexey, Dynamic integration of data mining methods in knowledge discovery systems. 69 p. (170 p.) Yhteenveto 2 p. 2002.

26 Akimov, Vladimir, Domain decomposition methods for the problems with boundary layers. 30 p. (84 p.). Yhteenveto 1 p. 2002.

27 Seyukova-Rivkind, Ludmila, Mathematical and numerical analysis of boundary value problems for fluid flow. 30 p. (126 p.) Yhteenveto 1 p. 2002.

28 Hämäläinen, Seppo, WCDMA Radio network performance. 235 p. Yhteenveto 2 p. 2003.

29 Pekkola, Samuli, Multiple media in group work. Emphasising individual users in distributed and real-time CSCW systems. 210 p. Yhteenveto 2 p. 2003.

30 Markkula, Jouni, Geographic personal data, its privacy protection and prospects in a location-based service environment. 109 p. Yhteenveto 2 p. 2003.

31 Honkaranta, Anne, From genres to content analysis. Experiences from four case organizations. 90 p. (154 p.) Yhteenveto 1 p. 2003.

32 Raitamäki, Jouni, An approach to linguistic pattern recognition using fuzzy systems. 169 p. Yhteenveto 1 p. 2003.

33 Saalasti, Sami, Neural networks for heart rate time series analysis. 192 p. Yhteenveto 5 p. 2003.

34 Niemelä, Marketta, Visual search in graphical interfaces: a user psychological approach. 61 p. (148 p.) Yhteenveto 1 p. 2003.

35 You, Yu, Situation Awareness on the world wide web. 171 p. Yhteenveto 2 p. 2004.

36 Taatila, Vesa, The concept of organizational competence – A foundational analysis. - Perusteanalyysi organisaation kompetenssin käsitteestä. 111 p. Yhteenveto 2 p. 2004.

JYVÄSKYLÄ STUDIES IN COMPUTING

37 Lyytikäinen, Virpi, Contextual and structural metadata in enterprise document management. - Konteksti- ja rakennemetatieto organisaation dokumenttien hallinnassa. 73 p. (143 p.) Yhteenveto 1 p. 2004.

38 Kaario, Kimmo, Resource allocation and load balancing mechanisms for providing quality of service in the Internet. 171 p. Yhteenveto 1 p. 2004.

39 Zhang, Zheying, Model component reuse. Conceptual foundations and application in the metamodeling-based systems analysis and design environment. 76 p. (214 p.) Yhteenveto 1 p. 2004.

40 Haarala, Marjo, Large-scale nonsmooth optimization variable metric bundle method with limited memory. 107 p. Yhteenveto 1 p. 2004.

41 Kalvine, Viktor, Scattering and point spectra for elliptic systems in domains with cylindrical ends. 82 p. 2004.

42 Dementieva, Maria, Regularization in multistage cooperative games. 78 p. 2004.

43 Maaranen, Heikki, On heuristic hybrid methods and structured point sets in global continuous optimization. 42 p. (168 p.) Yhteenveto 1 p. 2004.

44 Frolov, Maxim, Reliable control over approximation errors by functional type a posteriori estimates. 39 p. (112 p.) 2004.

45 Zhang, Jian, Qos- and revenue-aware resource allocation mechanisms in multiclass IP networks. 85 p. (224 p.) 2004.

46 Kujala, janne, On computation in statistical models with a psychophysical application. 40 p. (104 p.) 2004.,

47 Solbakov, Viatcheslav, Application of mathematical modeling for water environment problems. 66 p. (118 p.) 2004.

48 Hirvonen, Ari P., Enterprise architecture planning in practice. The Perspectives of information and communication technology service provider and end-user. 44 p. (135 p.) Yhteenveto 2 p. 2005.

49 Vartiainen, Tero, Moral conflicts in a project course in information systems education. 320 p. Yhteenveto 1p. 2005.

50 Huotari, Jouni, Integrating graphical information system models with visualization techniques. - Graafisten tietojärjestelmäkuvausten integrointi visualisointitekniikoilla. 56 p. (157 p.) Yhteenveto 1p. 2005.

51 Wallenius, Eero R., Control and management of multi-access wireless networks. 91 p. (192 p.) Yhteenveto 3 p. 2005.

52 Leppänen, Mauri, An ontological framework and a methodical skeleton for method engineering – A contextual approach. 702 p. Yhteenveto 2 p. 2005.

53 Matyukevich, Sergey, The nonstationary Maxwell system in domains with edges and conical points. 131 p. Yhteenveto 1 p. 2005.

54 Sayenko, Alexander, Adaptive scheduling for the QoS supported networks. 120 p. (217 p.) 2005.

55 Kurjenniemi, Janne, A study of TD-CDMA and WCDMA radio network enhancements. 144 p. (230 p.) Yhteenveto 1 p. 2005.

56 Pechenizkiy, Mykola, Feature extraction for supervised learning in knowledge discovery systems. 86 p. (174 p.) Yhteenveto 2 p. 2005.

57 Ikonen, Samuli, Efficient numerical methods for pricing American options. 43 p. (155 p.) Yhteenveto 1 p. 2005.

58 Kärkkäinen, Kari, Shape sensitivity analysis for numerical solution of free boundary problems. 83 p. (119 p.) Yhteenveto 1 p. 2005.

59 Helfenstein, Sacha, Transfer. Review, reconstruction, and resolution. 114 p. (206 p.) Yhteenveto 2 p. 2005.

60 Nevala, Kalevi, Content-based design engineering thinking. In the search for approach. 64 p. (126 p.) Yhteenveto 1 p. 2005.

61 Katasonov, Artem, Dependability aspects in the development and provision of location-based services. 157 p. Yhteenveto 1 p. 2006.

62 Sarkkinen, Jarmo, Design as discourse: Representation, representational practice, and social practice. 86 p. (189 p.) Yhteenveto 1 p. 2006.

63 Äyrämö, Sami, Knowledge mining using robust clustering. 296 p. Yhteenveto 1 p. 2006.

64 Ifinedo, Princely Emili, Enterprise resource planning systems success assessment: An integrative framework. 133 p. (366 p.) Yhteenveto 3 p. 2006.

65 Viinikainen, Ari, Quality of service and pricingin future multiple service class networks. 61 p. (196 p.) Yhteenveto 1 p. 2006.

66 Wu, Rui, Methods for space-time parameter estimation in DS-CDMA arrays. 73 p. (121 p.) 2006.

67 Parkkola, Hanna, Designing ICT for mothers. User psychological approach. – Tieto- ja viestintätekniikoiden suunnittelu äideille. Käyttäjäpsykologinen näkökulma. 77 p. (173 p.) Yhteenveto 3 p. 2006.

68 Hakanen, Jussi, On potential of interactive multiobjective optimization in chemical process design. 75 p. (160 p.) Yhteenveto 2 p. 2006.

69 Puttonen, Jani, Mobility management in wireless networks. 112 p. (215 p.) Yhteenveto 1 p. 2006.

70 Luostarinen, Kari, Resource , management methods for QoS supported networks. 60 p. (131 p.) 2006.

71 Turchyn, Pavlo, Adaptive meshes in computer graphics and model-based simulation. 27 p. (79 p.) Yhteenveto 1 p.

72 Zhovtobryukh, Dmytro, Context-aware web service composition. 290 p. Yhteenveto 2 p. 2006.

JYVÄSKYLÄ STUDIES IN COMPUTING

73 Kohvakko, Nataliya, Context modeling and utilization in heterogeneous networks. 154 p. Yhteenveto 1 p. 2006.

74 Mazhelis, Oleksiy, Masquerader detection in mobile context based on behaviour and environment monitoring. 74 p. (179 p). Yhteenveto 1 p. 2007.

75 Siltanen, Jarmo, Quality of service and dynamic scheduling for traffic engineering in next generation networks. 88 p. (155 p.) 2007.

76 Kuuva, Sari, Content-based approach to experiencing visual art. - Sisältöperustainen lähestymistapa visuaalisen taiteen kokemiseen. 203 p. Yhteenveto 3 p. 2007.

77 Ruohonen, Toni, Improving the operation of an emergency department by using a simulation model. 164 p. 2007.

78 Naumenko, Anton, Semantics-based access control in business networks. 72 p. (215 p.) Yhteenveto 1 p. 2007.

79 Wahlstedt, Ari, Stakeholders' conceptions of learning in learning management systems development. - Osallistujien käsitykset oppimisesta oppimisympäristöjen kehittämisessä. 83 p. (130 p.) Yhteenveto 1 p. 2007.

80 Alanen, Olli, Quality of service for triple play services in heterogeneous networks. 88 p. (180 p.) Yhteenveto 1 p. 2007.

81 Neri, Ferrante, Fitness diversity adaptation in memetic algorithms. 80 p. (185 p.) Yhteenveto 1 p. 2007.

82 Kurhinen, Jani, Information delivery in mobile peer-to-peer networks. 46 p. (106 p.) Yhteenveto 1 p. 2007.

83 Kilpeläinen, Turo, Genre and ontology based business information architecture framework (GOBIAF). 74 p. (153 p.) Yhteenveto 1 p. 2007.

84 Yevseyeva, Iryna, Solving classification problems with multicriteria decision aiding approaches. 182 p. Yhteenveto 1 p. 2007.

85 Kannisto, Isto, Optimized pricing, QoS and segmentation of managed ICT services. 45 p. (111 p.) Yhteenveto 1 p. 2007.

86 Gorshkova, Elena, A posteriori error estimates and adaptive methods for incompressible viscous flow problems. 72 p. (129 p.) Yhteenveto 1 p. 2007.

87 Legrand, Steve, Use of background real-world knowledge in ontologies for word sense disambiguation in the semantic web. 73 p. (144 p.) Yhteenveto 1 p. 2008.

88 Hämäläinen, Niina, Evaluation and measurement in enterprise and software architecture management. - Arviointi ja mittaaminen kokonais- ja ohjelmistoarkkitehtuurien hallinnassa. 91 p. (175 p.) Yhteenveto 1 p. 2008.

89 Ojala, Arto, Internationalization of software firms: Finnish small and medium-sized software firms in Japan. 57 p. (180 p.) Yhteenveto 2 p. 2008.

90 Laitila, Erkki, Symbolic Analysis and Atomistic Model as a Basis for a Program Comprehension Methodology. 321 p. Yhteenveto 3 p. 2008.

91 Nihtilä, Timo, Performance of Advanced Transmission and Reception Algorithms for High Speed Downlink Packet Access. 93 p. (186 p.) Yhteenveto 1 p. 2008.

92 Setämaa-Kärkkäinen, Anne, Network connection selection-solving a new multiobjective optimization problem. 52 p. (111p.) Yhteenveto 1 p. 2008.

93 Pulkkinen, Mirja, Enterprise architecture as a collaboration tool. Discursive process for enterprise architecture management, planning and development. 130 p. (215 p.) Yhteenveto 2 p. 2008.

94 Pavlova, Yulia, Multistage coalition formation game of a self-enforcing international environmental agreement. 127 p. Yhteenveto 1 p. 2008.

95 Nousiainen, Tuula, Children's involvement in the design of game-based learning environments. 297 p. Yhteenveto 2 p. 2008.

96 Kuznetsov, Nikolay V., Stability and oscillations of dynamical systems. Theory and applications. 116 p. Yhteenveto 1 p. 2008.

97 Khriyenko, Oleksiy, Adaptive semantic Web based environment for web resources. 193 p. Yhteenveto 1 p. 2008.

98 Tirronen, Ville, Global optimization using memetic differential evolution with applications to low level machine vision. 98 p. (248 p.) Yhteenveto 1 p. 2008.

99 Valkonen, Tuomo, Diff-convex combinations of Euclidean distances: A search for optima. 148 p. Yhteenveto 1 p. 2008.

100 Sarafanov, Oleg, Asymptotic theory of resonant tunneling in quantum waveguides of variable cross-section. 69 p. Yhteenveto 1 p. 2008.

101 Pozharskiy, Alexey, On the electron and phonon transport in locally periodical waveguides. 81 p. Yhteenveto 1 p. 2008.

102 Aittokoski, Timo, On challenges of simulation-based globaland multiobjective optimization. 80 p. (204 p.) Yhteenveto 1 p. 2009.

103 Yalaho, Anicet, Managing offshore outsourcing of software development using the ICT-supported unified process model: A cross-case analysis. 91 p. (307 p.) Yhteenveto 4 p. 2009.

104 Kollanus, Sami, Tarkastuskäytänteiden kehittäminen ohjelmistoja tuottavissa organisaatioissa. - Improvement of inspection practices in software organizations. 179 p. Summary 4 p. 2009.

105 Leikas, Jaana, Life-Based Design. 'Form of life' as a foundation for ICT design for older adults. - Elämälähtöinen suunnittelu. Elämänmuoto ikääntyville tarkoitettujen ICT tuotteiden ja palvelujen suunnittelun lähtökohtana. 218 p. (318 p.) Yhteenveto 4 p. 2009.

JYVÄSKYLÄ STUDIES IN COMPUTING

106 Vasilyeva, Ekaterina, Tailoring of feedback in web-based learning systems: Certitude-based assessment with online multiple choice questions. 124 p. (184 p.) Yhteenveto 2 p. 2009.

107 Kudryashova, Elena V., Cycles in continuous and discrete dynamical systems. Computations, computer assisted proofs, and computer experiments. 79 p. (152 p.) Yhteenveto 1 p. 2009.

108 Blackledge, Jonathan, Electromagnetic scattering and inverse scattering solutions for the analysis and processing of digital signals and images. 297 p. Yhteenveto 1 p. 2009.

109 Ivannikov, Andriy, Extraction of event-related potentials from electroencephalography data. - Herätepotentiaalien laskennallinen eristäminen EEG-havaintoaineistosta. 108 p. (150 p.) Yhteenveto 1 p. 2009.

110 kalyakin, Igor, Extraction of mismatch negativity from electroencephalography data. - Poikkeavuusnegatiivisuuden erottaminen EEG-signaalista. 47 p. (156 p.) Yhteenveto 1 p. 2010.

111 Heikkilä, Marikka, Coordination of complex operations over organisational boundaries. 265 p. Yhteenveto 3 p. 2010.

112 Fekete, Gábor, Network interface management in mobile and multihomed nodes. 94 p. (175 p.) Yhteenveto 1 p. 2010.

113 Kujala, Tuomo, Capacity, workload and mental contents - Exploring the foundations of driver distraction. 146 p. (253 p.) Yhteenveto 2 p. 2010.

114 Lugano, Giuseppe, Digital community design - Exploring the role of mobile social software in the process of digital convergence. 253 p. (316 p.) Yhteenveto 4 p. 2010.

115 Kampylis, Panagiotis, Fostering creative thinking. The role of primary teachers. - Luovaa ajattelua kehittämässä. Alakoulun opettajien rooli. 136 p. (268 p.) Yhteenveto 2 p. 2010.

116 Toivanen, Jukka, Shape optimization utilizing consistent sensitivities. - Muodon optimointi käyttäen konsistentteja herkkyyksiä. 55 p. (130p.) Yhteenveto 1 p. 2010.

117 Mattila, Keijo, Implementation techniques for the lattice Boltzmann method. - Virtausdynamiikan tietokonesimulaatioita Hila-Boltzmann -menetelmällä: implementointi ja reunaehdot. 177 p. (233 p.) Yhteenveto 1 p. 2010.

118 Cong, Fengyu, Evaluation and extraction of mismatch negativity through exploiting temporal, spectral, time-frequency, and spatial features. - Poikkeavuusnegatiivisuuden (MMN) erottaminen aivosähkönauhoituksista käyttäen ajallisia, spektraalisia, aika-taajuus - ja tilapiirteitä. 57 p. (173 p.) Yhteenveto 1 p. 2010.

119 Liu, Shenghua, Interacting with intelligent agents. Key issues in agent-based decision support system design. 90 p. (143 p.) Yhteenveto 2 p. 2010.

120 Airaksinen, Tuomas, Numerical methods for acoustics and noise control. - Laskennallisia menetelmiä akustisiin ongelmiin ja melunvaimennukseen. 58 p. (133 p.) Yhteenveto 2 p. 2010.

121 Weber, Matthieu, Parallel global optimization Structuring populations in differential evolution. - Rinnakkainen globaalioptimointi. Populaation rakenteen määrittäminen differentiaalievoluutiossa. 70 p. (185 p.) Yhteenveto 2 p. 2010.

122 Väärämäki, Tapio, Next generation networks, mobility management and appliances in intelligent transport systems. - Seuraavan sukupolven tietoverkot, liikkuvuuden hallinta ja sovellutukset älykkäässä liikenteessä. 50 p. (111 p.) Yhteenveto 1 p. 2010.

123 Viukari, Leena, Tieto- ja viestintätekniikkavälitteisen palvelun kehittämisen kolme diskurssia. - Three discourses for an ICT-service development . 304 p. Summary 5 p. 2010.

124 Puurtinen, Tuomas, Numerical simulation of low temperature thermal conductance of corrugated nanofibers. - Poimutettujen nanokuitujen lämmönjohtavuuden numeerinen simulointi matalissa lämpötiloissa . 114 p. Yhteenveto 1 p. 2010.

125 Hiltunen, Leena, Enhancing web course design using action research . - Verkko-opetuksen suunnittelun kehittäminen toimintatutkimuksen keinoin . 192 p. Yhteenveto 2 p. 2010.

126 Aho, Kari, Enhancing system level performance of third generation cellular networks through VoIP and MBMS services. 121 p. (221 p.). Yhteenveto 2 p. 2010.

127 Häkkinen, Markku, Why alarms fail. A cognitive explanatory model. 102 p. (210 p.). Yhteenveto 1 p. 2010.

128 Pennanen, Anssi, A graph-based multigrid with applications. - Graafipohjainen monihilamenetelmä sovelluksineen. 52 p. (128 p.). Yhteenveto 2 p. 2010.

129 Ahlgren, Riikka, Software patterns, organizational learning and software process improvement. 70 p. (137 p.). Yhteenveto 1 p. 2011.

130 Nikitin, Sergiy, Dynamic aspects of industrial middleware architectures 52 p. (114 p.). Yhteenveto 1 p. 2011.

131 Sindhya, Karthik, Hybrid Evolutionary Multi-Objective Optimization with Enhanced Convergence and Diversity. 64 p. (160 p.). Yhteenveto 1 p. 2011.

JYVÄSKYLÄ STUDIES IN COMPUTING

132 Mali, Olli, Analysis of errors caused by incomplete knowledge of material data in mathematical models of elastic media. 111 p. Yhteenveto 2 p. 2011.

133 Mönkölä, Sanna, Numerical Simulation of Fluid-Structure Interaction Between Acoustic and Elastic Waves. 136 p. Yhteenveto 2 p. 2011.

134 Puranen, Tuukka, Metaheuristics Meet Metamodels. A Modeling Language and a Product Line Architecture for Route Optimization Systems. 270 p. Yhteenveto 1 p. 2011.

135 Mäkelä, Jukka, Mobility Management in Heterogeneous IP-networks. 86 p. (145 p.) Yhteenveto 1 p. 2011.