JYVÄSKYLÄ STUDIES IN COMPUTING
112

Gábor Fekete

Network Interface Management in Mobile and Multihomed Nodes

JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ STUDIES IN COMPUTING 112

Gábor Fekete

Network Interface Management in Mobile and Multihomed Nodes

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi Agora-rakennuksen Alfa-salissa
kesäkuun 11. päivänä 2010 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Agora, Alfa-hall, on June 11, 2010 at 12 o'clock noon.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2010

Network Interface Management in Mobile and Multihomed Nodes

JYVÄSKYLÄ STUDIES IN COMPUTING 112

Gábor Fekete

Network Interface Management in Mobile and Multihomed Nodes

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2010

Editor
Timo Männikkö
Department of Mathematical Information Technology, University of Jyväskylä
Pekka Olsbo
Publishing Unit, University Library of Jyväskylä

URN:ISBN:978-951-39-3923-6
ISBN 978-951-39-3923-6 (PDF)

ISBN 978-951-39-3888-8 (nid.)
ISSN 1456-5390

Jyväskylä University Printing House, Jyväskylä 2010

ABSTRACT

Fekete, Gábor
Network interface management in mobile and multihomed nodes
Jyväskylä: University of Jyväskylä, 2010, 94 p.(+included articles)
(Jyväskylä Studies in Computing
ISSN 1456-5390; 112)
ISBN 978-951-39-3923-6 (PDF), 978-951-39-3888-8 (nid.)
Finnish summary
Diss.

Mobile computing and mobile networking allow people to roam freely and yet still be reachable and connected to others, giving the feeling of safety. Therefore, we rely more and more on mobile networks and corresponding mobile nodes (handsets). This reliance calls for a robust infrastructure that can provide continuous network connection to end nodes, covering the largest geographical area possible.

Due to the characteristics of the available network access technologies, this goal is achieved by deploying different access networks depending on the physical location and service quality requirements. For this reason, mobile nodes are equipped with multiple different network interfaces, each being able to connect to a specific access network. This configuration introduces challenges with regards to managing the available network interfaces in mobile nodes.

Multihomed nodes can be seen as a generalization of *multiple interfaced* mobile nodes. There may be multiple different paths between a multihomed node and any other node in the network. The management of these paths poses a similar challenge as the management of multiple network interfaces.

In this dissertation, we explore different approaches to network interface and path management in mobile and multihomed nodes. We analyze the behaviour of the Mobile IPv6 protocol and propose an interface management system that takes into account user preferences and policies, and combines it with various parameters of the available network interfaces in order to provide the best possible connection to the network.

The study of different multihoming capable protocols initiated our research for path management in multihomed nodes. This resulted in two different approaches for path selection for inbound network traffic. The first focuses on the separation of path management policies from the underlying multihoming protocols. The second approach leverages the inherent flow identification information in network datagrams to avoid the explicit transfer and deployment of the necessary policies for inbound path selection.

Keywords: IP mobility, handover, Mobile IPv6, multihoming, policy, All-IP network, cross-layer, adaptation

Author

Gábor Fekete
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Supervisor

Professor Timo Hämäläinen
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Reviewers

Prof. Gennady Yanovsky
Bonch-Bruevich Saint-Petersburg State University of Telecommunications
Russia

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

Dr. Kimmo Raivio
Aalto University
School of Science and Technology
Department of Information and Computer Science
Finland

Opponent

Dr. Pertti Raatikainen
VTT Technical Research Centre of Finland
Finland

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the invaluable help of the many people I had the privilege to encounter during the work.

Professor Timo Hämäläinen, my supervisor, provided the much needed guidance, patience, help, encouragement and valuable contributions. I would like to thank Professor Gennady Yanovsky, Dr. Kimmo Raivio and Professor Yevgeni Koucheryavy for their effort with reviewing the dissertation. I thank Dr. Pertti Raatikainen for being my opponent at the defense of this work.

I owe a lot to my fellow researchers and project managers with whom we have spent many months and years together with valuable debates, discussions, co-operation and a great company: Jani Puttonen, Petteri Weckström, Jukka Mäkelä, Tapio Väärämäki.

Most of the fruits of this work is the result of the great times I have spent with the students involved in the related projects: Ewa Wlodarczyk, Ismo Keränen, Lukas Lukovsky, Martin Bauma, Michal Chukwu and others. Special thank goes to Pawel Rybczyk for the cheerful times and his publication contributions. I also thank Jorma Narikka for his ideas and contributions in the co-authored publications.

I would like to thank the help and resources provided by the staff of the University of Jyväskylä, its Department of Mathematical Information Technology and the Jyväskylä University of Applied Sciences.

The COMAS graduate school and the CIMO Fellowships provided the necessary financial support for this work. Their help is much appreciated. I thank the support, encouragement and understanding of my employer, Tieto Oy.

I especially thank my parents, Károly Fekete and Váradi Ibolya, relatives, and my friends for their love, patience and trust.

Jyväskylä, May 2010
Gábor Fekete

ACRONYMS

3GPP	3rd Generation Partnership Project
ABC	Always Best Connected
AP	Access Point
API	Application Programming Interface
AR	Access Router
BGP	Border Gateway Protocol
BU	Binding Update
CARD	Candidate Access Router Discovery
CN	Correspondent Node
CoA	Care-of Address
DAD	Duplicate Address Detection
DCCP	Datagram Congestion Control Protocol
DFZ	Default Free Zone
DNA	Detecting Network Attachment
DNS	Domain Name System
EDGE	Enhanced Data rates for GSM Evolution
EGP	Exterior Gateway Protocol
ESP	Encapsulating Security Payload
FMIPv6	Fast Handover for Mobile IPv6
FFHMIPv6	Flow based Fast Handover for Mobile IPv6
FQDN	Fully Qualified Domain Name
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HA	Home Agent
HIP	Host Identity Protocol
HIT	Host Identity Tag
HoA	Home Address
IANA	Internet Assigned Numbers Authority
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
INAS	Implicit Inbound Address Selection
IP	Internet Protocol

IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISP	Internet Service Provider
ITU	International Telecommunication Union
LAN	Local Area Network
LAC	Link Access Controller
LID	Locator Identifier
LIP	Link Information Provider
LIR	Local Internet Registry
LISP	Locator/ID Separation Protocol
LTE	Long Term Evolution
MADM	Multiple Attribute Decision Making
MIH	Media Independent Handover
MIPv6	Mobility Support for IPv6
MMP	Mobility Management Protocol
MTCP	Multipath TCP
MN	Mobile Node
NGN	Next Generation Network
NLMP	Network Layer Multihoming Protocol
OSI	Open System Interconnection
QoS	Quality of Service
RIR	Regional Internet Registry
RO	Route Optimization
RTT	Round-Trip Time
SA	Security Association
SCTP	Stream Control Transport Protocol
SHIM6	Site Multihoming by IPv6 Intermediation
SIP	Session Initiation Protocol
SPI	Security Parameter Index
TCP	Transmission Control Protocol
TE	Traffic Engineering
UDP	User Datagram Protocol
ULID	Upper Layer Identifier
ULP	Upper Layer Protocol

VERHO	Vertical Handover in Heterogeneous Environment
WLAN	Wireless Local Area Network
WiMAX	Worldwide Interoperability for Microwave Access

LIST OF FIGURES

LIST OF TABLES

CONTENTS

LIST OF INCLUDED ARTICLES

PI Jani Puttonen, Gábor Fekete, Pawel Rybczyk and Jorma Narikka. Practical Experimentation of Mobile IPv6 in Heterogeneous Environment. *Acta Electrotechnica et Informatica*, 2006.

PII Jukka Mäkelä, Timo Hämäläinen, Gábor Fekete and Jorma Narikka. Intelligent Vertical Handover System for Mobile Clients. *Proceedings of the 3rd International Conference on Emerging Telecommunications Technologies and Applications (ICETA 2004), Košice, Slovakia*, 2004.

PIII Jani Puttonen, Gábor Fekete, Jukka Mäkelä, Timo Hämäläinen and Jorma Narikka. Using Link Layer Information for Improving Vertical Handovers. *Proceedings of the 16th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC2005), Barcelona, Spain*, 2005.

PIV Jani Puttonen and Gábor Fekete. Interface Selection for Multihomed Mobile Hosts. *Proceedings of the 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC2006), Helsinki, Finland*, 2006.

PV Tapio Väärämäki, Jani Puttonen and Gábor Fekete. Next Generation Network Related Standardization - Enablers for the Convergence. *Proceedings of the Fourth Advanced International Conference on Telecommunications (AICT'08), Athens, Greece*, 2008.

PVI Jani Puttonen, Gábor Fekete, Tapio Väärämäki and Timo Hämäläinen. Multiple Interface Management of Multihomed Mobile Hosts in Heterogeneous Wireless Environments. *Proceedings of the 8th International Conference on Networks (ICN 2009), Cancun, Mexico*, 2009.

PVII Gábor Fekete and Timo Hämäläinen. State of Host-Centric Multihoming in IP networks. *Proceedings of the Third International Conference on New Technologies, Mobility and Security (NTMS 2009), Cairo, Egypt*, 2009.

PVIII Gábor Fekete. Policy Based Flow Management with the Host Identity Protocol for Multihomed Hosts. *Proceedings of the The Fifth European Conference on Universal Multiservice Networks (ECUMN 2009), Sliema, Malta*, 2009.

PIX Gábor Fekete. State of Host-Centric Multihoming and Traffic Control in IP networks. *Reports of the Department of Mathematical Information Technology, Jyväskylä, Finland*, 2010.

CONTRIBUTIONS IN THE INCLUDED ARTICLES

During the work on the subject of this thesis, the author has produced and cooperated in several publications.

In [PI], the performance of the Linux implementation of Mobile IPv6 is investigated with regards to handover delays and packet loss. This study provided the motivation for finding ways of reducing delays and packet loss by using Link Layer information. The author contributed to the test case definitions (the used test topology) and provided guidance for test setup, execution and evaluation (delay measurement results with regards to the steps involved during handovers). The results of this article are presented in Section 2.4.1.

In [PII], the Vertical Handover in Heterogeneous Environment (VERHO) system is introduced. It describes the overall idea and motivation for a policy based network interface selection system. The author contributed to the idea behind the VERHO system and conducted the feasibility study and preliminary work for a possible implementation. This work involved the analyzation of an existing Mobile IPv6 implementation, which lead to a proof of concept implementation of the necessary extensions that made the VERHO system possible (controlling Mobile IPv6 handovers based on policies and network conditions). The results of this article are presented in Section 3.2.

In [PIII], a system, called Link Information Provider (LIP), is introduced for extracting Link Layer information and providing it to other layers. The method gathers information from different access technologies, converts them to a common format and representation and provides it for consumers (e.g. Upper Layer Protocols). This mechanism is utilized by the VERHO system to trigger Mobile IPv6 handovers. The author designed, implemented and tested the LIP module that forms one of the pillars of the VERHO system, and provided the necessary technical explanations and descriptions for the publication. The results of this article are presented in Section 3.3.

In [PIV], we introduce several methods for network interface selection based on multiple criterias. It proposes one of these that provides the best results and flexibility to achieve a policy based interface selection method. The author contributed to the study of the proposed methods and to the implementation and testing of the chosen one in the VERHO system. This study provided the measurements related to the combined algorithm in the publication. The results of this article are presented in Section 3.4.

In [PV], we survey the ongoing research activities and trends with regards to the convergence towards an all-IP world. The author contributed the parts of this article related to the Internet Engineering Task Force (IETF) activities where the roles, scope and achievements of different working groups are reviewed. The results of this article are presented in Section 3.1.

In [PVI], the VERHO system is presented. It describes the architecture and shows possible use cases by demonstrating some applications that take advantage of the system. The applications include video and audio applications that

adapt to the quality changes due to handovers between different access technologies. It also demonstrates a UI application to modify the behavior of the system via user defined policies. The author was responsible for the design (except for the preference value calculation) and implementation of the core VERHO system (excluding the prototype demo applications) and contributed most to the parts of this article related to system architecture, link information provider, link module, access point module, link access controller, and preference value calculation. The results of this article are presented in Section 3.2 and 3.5.

In [PVII], we survey the current trends in host-centric multihoming. Multihoming provides multiple paths to reach the same host. In host-centric approaches, the selection of paths can be influenced by the end-hosts themselves. In this article, we review the most relevant host-centric proposals and focus on what methods do they provide for path selection. The author contributed the majority of the covered topics. The results of this article are presented in Section 4.1.

In [PVIII], the author introduces a method to exchange flow management policies between end-nodes for the Host Identity Protocol (HIP). The method uses a reduced variant of the Lua programming language to represent policies. The advantage of this method is its extensibility and re-usability with other multihoming capable Network Layer protocols (e.g., Mobile IPv6). The results of this article are presented in Section 4.2.

In [PIX], the author reviewed and surveyed the state of host-centric multihoming and host-side traffic control. The article reviews and compares the most relevant multihoming protocols at the transport and network layers. Furthermore, the traffic control capabilities of the reviewed protocols are also discussed. The article serves as a general overview of the various multihoming approaches. The similarities are highlighted as to provide a basis for further research on common frameworks for host-centric multihoming and related traffic control. The results of this article are presented in Section 4.1.

1 INTRODUCTION

Mobile phones are an indispensable tool in our everyday communications. Mobile phone networks are unstoppably moving towards an Internet Protocol (IP) infrastructure with the goal of having almost all the services provided by today's cellular networks available through an all-IP network ([69], [68]).

With the advent of cellular networks and mobile phones, mobility during communication became a reality for common people. This need for mobility also appeared in other types of communication, the Internet. In contrast to cellular mobile networks, the Internet is an IP network that was designed with stationary nodes in mind.

Wireless access technologies are the base of mobility in IP networks. These technologies vary with regards to their physical area coverage and performance. For this reason, there is a trend in the manufacturing of current mobile end nodes to include multiple different wireless network interfaces. This is to provide the nodes with the most chance of being able to connect to the Internet through at least one available access technology.

There are two essential components for efficient mobile communication over IP networks:

- Host Mobility: the ability of network nodes to change their point of attachment to the network.

- Quality of Service: the ability to guarantee a well defined user perceived quality for services.

This dissertation is concerned with the management of multiple network interfaces and multiple IP addresses in mobile and multihomed network nodes. It involves areas mostly from host mobility and, to a limited extent, quality of service management.

1.1 Assumptions and definitions

In this dissertation, we deal with IP networks. For the sake of clarification, we present the assumptions that define an *IP network* in our context.

- We consider only network nodes with an OSI compatible network stack (e.g., Transmission Control Protocol/Internet Protocol (TCP/IP)).
- The network consists of IP subnetworks, each represented by LANs.
- Network nodes connect to the LANs via Access Points. The underlying access technologies provide these Access Points. In this regard, a switch or hub can be considered as an IEEE 802.3 Ethernet Access Point. Similarly, an IEEE 802.11 Wireless local Area Network (WLAN) Base Station is considered an Access Point.
- We assume that nodes communicate with Internet Protocol version 4 (IPv4) or version 6 (IPv6) at the Network Layer.

We use the terms *IP network* and *IP subnetwork* interchangeably. Ad-hoc IP networks are not considered.

About network nodes, we have the following general assumptions:

- A network node is equipped with one or more network interfaces.
- Each network interface corresponds to one access technology.
- IP addresses are assigned to network interfaces.
- One network interface can be assigned multiple IP addresses.

This means that a network node can have multiple IP addresses. When we discuss mobile nodes in Chapter 2 and Chapter 3, we assume that the mobile node has only one IP address per network interface. General node multihoming, the case when a network interface may have multiple IP addresses, is handled in Chapter 4.

1.2 Mobility

IP networks were designed for stationary nodes. Every node is identified by an IP address. The IP address has a double role. It serves both as a node identifier and as a locator. The locator aspect is used by routers to deliver the datagrams to their destination.

A mobile node is a network node that changes its topological position or point of attachment in the network. This change in the topology is called a *handover* ([35]).

When a network node changes its point of attachment, its IP address may also change. This means that other nodes in the network need to be updated with this new IP address in case they want to reach the moved node at its new location.

The change in the IP address of the moving node causes established Upper Layer Protocol (ULP, above the Network Layer) connections to break. This is problematic for long-time connections, such as voice communication, that can easily last for several minutes.

Another related issue is packet-loss and handover latency that can easily cause user visible service degradation during mobility events. Therefore, minimizing these two parameters is essential for providing enjoyable mobile experience for users.

It is a general trend that network nodes (e.g., notebooks, phones, etc.) are equipped with multiple network interfaces. The reason for this is the fact that there are multiple different network technologies in existence (e.g., Ethernet, WLAN, Worldwide Interoperability for Microwave Access (WiMax)). These technologies provide different coverage and performance. Also, the networks behind the available network interfaces may be operated by different Internet Service Providers (ISPs).

A network node can use the available network interfaces either one at a time or simultaneously. In the former case, it is an obvious assumption that the user of a mobile network node would like to be connected, with minimal effort, to the network that is available at her current location. If multiple different networks are available, then the user would expect to get connected to the one that is considered the *best* according to some well defined preferences. We call this feature Always Best Connected (ABC) [23].

1.3 Multihoming

The case when the network node uses its multiple network interfaces simultaneously, is a subset of a more general case. This general case is called *multihoming*. A multihomed node has multiple IP addresses that it can be reached at. These IP addresses may be assigned to the same or to different network interfaces, and may be of the same or different IP subnetworks. Usually, the path of IP datagrams is determined by the source and destination IP addresses. Therefore, the address selection in multihomed nodes provides a tool for local traffic engineering that is likely to affect the quality of service.

1.4 Quality of Service

IP networks are packet-switched, end-to-end networks where the logic is pushed as much towards the end-nodes as possible. The core of the network consists

of *dumb* routers that merely forward individual datagrams based on destination addresses (and sometimes other parts of the datagram).

In a pure IP network, data is transferred with a *best-effort* delivery guarantee. It means that routers try hard to deliver data packets to destinations but the communicating peers have to deal with possible packet-losses. *Best-effort* delivery guarantee is not sufficient for voice communication over IP (VoIP). This can be alleviated by either building faster networks so, packet-loss becomes unnoticeable due to fast retransmissions, or by introducing Quality of Service (QoS) measures.

Currently there are two kinds of major QoS methods in IP networks:

- Integrated Services: IntServ: A fine-grained QoS method [10]. Resources are reserved for a specific session. A session is a set of packets belonging together according to source IP address, destination IP address and other parameters. Resources for the session are reserved on every router on the path from the sender to the receiver with the Resource Reservation Protocol (RSVP [11]).

- Differentiated Services: DiffServ: A coarse-grained QoS method [8]. Individual IP datagrams are assigned to different classes at edge routers by packet marking. Core routers handle IP datagrams according to their class (i.e., packet marks).

An IP datagram in a packet-switched network travels via a path from the source node through intermediate nodes (e.g., routers) until it reaches its destination node. IntServ and DiffServ are ways to alter the path the datagram takes in the intermediate nodes.

When a node is multihomed, there exist multiple paths between the node and any other node in the network. Current approaches for multihoming in the IPv4 Internet put either too much pressure on the core routers (e.g., too large BGP routing table sizes) or put too much logic towards the core (e.g., middle boxes). For this reason, multihoming needs to be solved and re-engineered in a different way for the IPv6 Internet. The current proposal for it is host-centric multihoming where multihomed nodes (hosts) have multiple IP addresses.

In the host-centric multihoming approach, the path of network datagrams can be affected by merely altering the source or destination IP addresses. A change in the source IP address of an outgoing datagram is likely to cause a change also in its path by taking a different outbound ISP or network interface. Similarly, a change in the destination IP address of an outgoing datagram is likely to cause a change also in its path by taking a different inbound ISP or network interface at the destination. Therefore, path manipulation based on IP address selection at end-nodes can affect the quality of service by sending datagrams over paths with different characteristics.

1.5 Policies

It is desirable that nodes make their decisions automatically about what network interface or network to connect to, or what path an outgoing datagram should take. On the other hand, it is unlikely that any one algorithm would prove to be satisfactory to every user. This means that even though the decision should be as automatic as possible, there should be a way for users (humans or applications) to alter the behavior of the algorithm in a dynamic way.

Therefore a policy based approach is beneficial. With the help of policies, the user can tell the system how it should behave in certain situations.

1.6 Problem statement

The aim of the research efforts of this dissertation is to find ways and methods that can help addressing the problems highlighted in the previous few sections.

We have addressed these issues in two separate research work. One was the development of a handover management system called Vertical Handover in Heterogeneous Environment (VERHO) in Chapter 3. The other was the development of inbound path selection protocols for multihomed nodes in Chapter 4.

The following list shows the problems that are addressed in this dissertation and the proposed solutions.

- Handover latency and packet loss: The time during a mobility event of a mobile node when it is unable to receive or send data packets is the handover latency. This time must be reduced as much as possible so data communication is the least interrupted. The operations performed during handovers can affect established communication sessions by causing packet-loss. The amount of lost packets therefore must be minimized. In the VERHO system, a solution is given by a handover controller that disseminates information to ULPs about network characteristics before handovers occur. This allows ULPs to prepare for gaps in the communication and adjust themselves for the changed environment.

- Policy based interface selection: When a mobile node has multiple network interfaces, there must be a flexible way to influence the selection of the one that is used for data communication. The VERHO system provides a policy based interface selection mechanism that can take into account multiple attributes in its decision making process.

- Multihoming path selection: When a network node is multihomed, there are multiple communication paths between the node and its peers. These paths may have different characteristics (e.g., price, bandwidth, latency, jitter). Therefore, it would be desirable if there was a flexible way to influence

the selection of the available paths. A policy based path selection mechanism is designed for multihomed nodes for the Host Identity Protocol (HIP). Furthermore, an alternative approach to policy exchange between multihomed nodes is introduced.

1.7 Outline

Chapter 2 introduces mobility in IP networks. It provides an analysis of the handover performance of Mobile IPv6. This analysis lead to the development of the VERHO system. Chapter 3 describes the VERHO system. The research work for VERHO involved three areas. Link Layer information acquisition in different wireless access technologies, Link Layer information utilization in Upper Layers and policy based network interface selection. Chapter 4 introduces the research work done in the area of multihoming and network address (network interface) selection. It describes two different approaches to inbound address selection that support multiple Network Layer based multihoming protocols. Chapter 5 concludes the dissertation.

2 MOBILITY

2.1 Layers of mobility

A mobile node is a network node that changes its topological position or point of attachment in the network. This change in the topology is called a *handover* and it may involve different layers in the network stack.

Figure 1 shows the cases of handovers we deal with in this dissertation. The mobile node moves from IP Network A to IP Network B. The Access Routers (AR) are the default routers in their respective networks. The Access Points (AP) resemble the Layer 2 attachment points of either the same or different access technologies (e.g., IEEE 802.3 Ethernet switch, IEEE 802.11 WLAN Base Station, etc.).

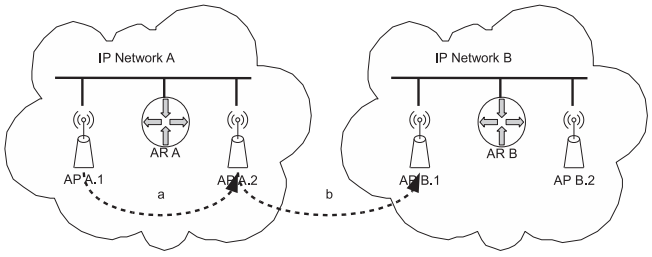

FIGURE 1 Layer 2 and Layer 3 handovers

In Figure 1, node movement from AP A.1 to AP A.2 (marked as *a*) is performed at Layer 2 (or the Link Layer). During this movement, the mobile node stays in IP Network A, which means that the IP address of the mobile node does not change. This kind of handover is usually handled automatically at the Link Layer in an access technology specific way and we do not consider its mechanism here.

The movement between AP A.2 and AP B.1 (movement *b*) involves both a Link Layer movement and a Network Layer movement. Since after the movement, the mobile node ends up connected to IP Network B, its IP setting also changes. By IP setting, we mean

- Default Router,
- IP subnetwork and
- IP address.

Out of these, the change in the IP address is the most important. This change causes the need for an IP mobility management protocol. The new IP address of the mobile node has to be signaled to the appropriate peers for the following reasons:

- New connections: Allow peers to establish new data communication towards the mobile node.
- Established connections: Allow the continuation of already established data communications between the mobile node and its peers.

The preservation of established communications is necessary only for long lived sessions. For example, when a user browses a website, the web client establishes many short lived Transmission Control Protocol (TCP) connections to the web server. Certainly, such a use case does not require preservation of these connections since the user can just refresh (reload) the site in case a handover occurs. On the other hand, with the proliferation of multimedia over IP networks, such as streaming audio and video, voice and video communication, peer-to-peer file-sharing, there are more and more communication sessions that last for multiple minutes. If these long-lived sessions break during mobile node movements user satisfaction is likely to drop.

2.2 Types of handover

Handovers can be classified in various different ways [45]. In this section, we introduce the ones that are relevant to this work.

Figure 2 shows the two kinds of Link Layer handovers. When the handover is performed between Access Points belonging to the same access technology, we talk about *horizontal handovers*. On the other hand, when the handover is performed between different access technologies, we talk about *vertical handovers*.

In Figure 2, the ellipses represent the coverage area of Access Points. The horizontal axis represent also the physical horizon. In this sense, at a certain physical position the coverage area of multiple, both the same and different, Access Points may overlap. For example, at the dashed vertical line, a mobile node is in the coverage area of an Access Point of access technology Tech 3 and two Access Points of access technology Tech 1.

In current consumer network nodes, usually one network interface corresponds to one access technology. This means that a mobile node equipped with a single network interface can only perform horizontal handovers.

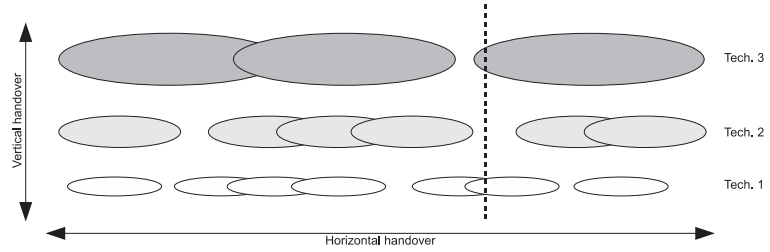

FIGURE 2 Vertical and horizontal handovers

During the handover, there may be a time interval during which the mobile node is offline, that is, it is not connected to the network at all. Based on this, we distinguish between two kinds of handovers.

- Hard handover: The mobile node disconnects temporarily from the network during the handover process.
- Soft handover: The mobile node always maintains at least one connection to the network during the handover process.

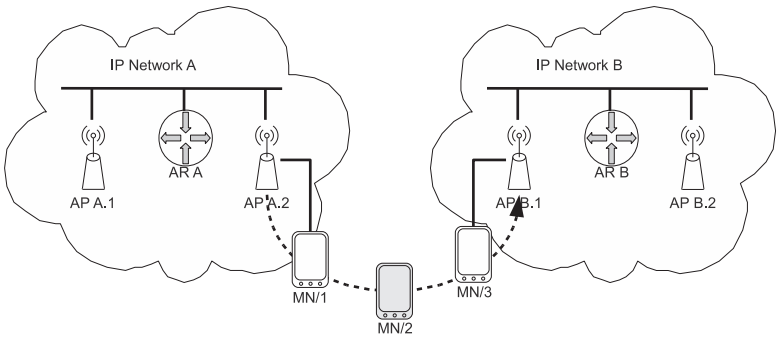

FIGURE 3 Hard handover

Figure 3 shows a Mobile Node (MN) performing a hard handover. At phase 1 (MN/1), the mobile node is connected to AP A.2, then it disconnects from the network at phase 2 (MN/2) only to reconnect to AP B.1 at phase 3 (MN/3). If a mobile node is equipped with only a single network interface, it can perform only hard handovers. An obvious side effect of hard handovers is that data packets sent during the offline phase are likely to get lost and need to be retransmitted by the sender, unless they are buffered at some point.

Figure 4 shows a mobile node performing a soft handover. At phase 1 (MN/1), the mobile node is connected to AP A.2, then it connects to AP B.1 without disconnecting from AP A.2. Then, in phase 3, the mobile node may disconnect from AP A.2. Soft handover can be performed only if the mobile node is physically capable of connecting to multiple access points at the same time,

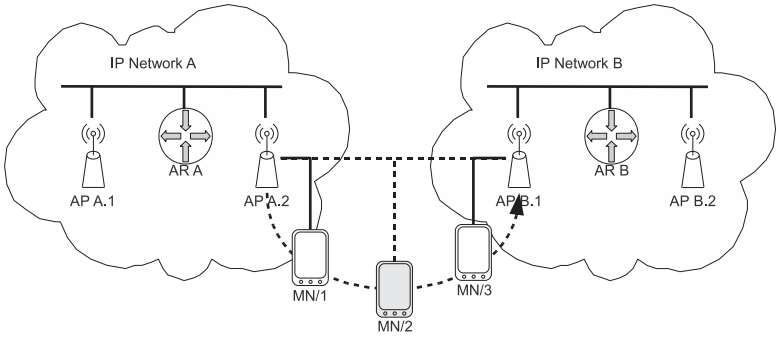

FIGURE 4 Soft handover

for example, it has multiple network interfaces. With this kind of handovers the amount of lost packets during the handover process can be reduced.

Another classification of handovers is based on the time *when* the handover process is initiated by the Mobility Management Protocol (MMP).

- Proactive: A handover is proactive when the MMP initiates its handover process before the MN leaves the current Access Point (i.e., before the Layer 2 handover).
- Reactive: A handover is reactive when the MMP initiates its handover process after the MN has already connected (or associated) to the new Access Point (i.e., after the Layer 2 handover).

Handovers can be distinguished also upon *which* node starts the MMP process. In this regards, we talk about the following types:

- Mobile initiated: The MN initiates the handover process based on its own information about the network environment.
- Network initiated: Some node in the subnetwork of the MN, for example the Access Router or some other agent, initiates the process. This initiation is usually in a form of a hint to the MN that it should start the handover.

2.3 Mobility Management Protocols

The location and sometimes the identity of network nodes in IP networks are represented by IP addresses. Different IP networks mean different IP subnetworks and IP addresses in one network are not likely to be usable in different networks.

When a mobile node changes its point of attachment to the network, it may end up in a different IP network (Network Layer handover). This means that the mobile node is no longer reachable at the IP addresses belonging to its previous network. The main reasons for this un-reachability are the following:

- Routing: Packets sent towards the mobile node containing a destination IP address that does not belong to any of the mobile node's visited networks will be routed to the wrong location and be dropped.
- Ingress filtering: Even if a packet with an invalid destination address reaches the currently visited network of the mobile node, the router of the visited network is likely to drop it. This is due to the practice of ingress filtering that drops every incoming packet with a destination address that is officially not routed through the visited network.
- Egress filtering: If the mobile node sends a packet with a source address that does not belong to the visited network, the router of the visited network will drop it.

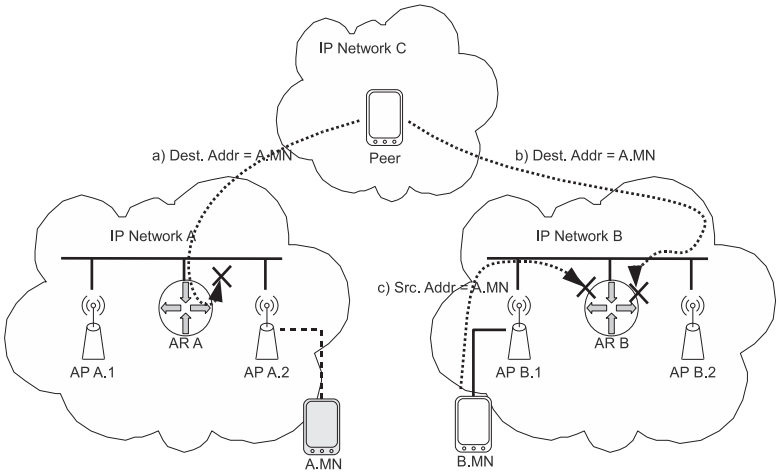

FIGURE 5 Un-reachability after Network Layer handover

Figure 5 shows these three scenarios. The mobile node moved from access point AP A.2 to access point AP B.1. Arrow *a* is a failed delivery of packets by the routing system to the mobile node's old location (IP address A.MN). Arrow *b* is dropped inbound packets due to ingress filtering at AR B, and arrow *c* is dropped outbound packets due to egress filtering by AR B.

In case a mobile node performs a vertical or a horizontal Link Layer handover inside the same IP subnetwork, it can continue data communication with other peers that reside in that network using either Link Layer communication or Network Layer communication. On the other hand, when the mobile node performs a Network Layer handover, communication with peers outside the newly visited IP subnetwork (peers in the old subnetwork or peers in other subnetworks) can be accomplished only by the Network Layer using IP. Since IP is based on routers and IP addresses to deliver the datagrams, the management of IP addresses of a mobile node is crucial. For these reasons, we discuss Mobility Management Protocols (MMP) only at the Network Layer and above.

2.3.1 Task of Mobility Management

An MMP has to perform the following main tasks:

- IP address registration and update,
- IP address agnostic packet delivery between ULPs.

The IP address of a mobile node changes every time it moves into a different IP subnetwork. Therefore its current IP address has to be stored at all times in a location directory. This location directory is used whenever a peer wants to send IP datagrams to the mobile node. It is not enough to only store the IP address once but it also has to be updated every time it changes.

Such a location directory can be, for example, a Dynamic Domain Name System (DNS) server ([5]). The mobile node may register its current IP address with its Fully Qualified Domain Name (FQDN) in the DNS server. When the peer wants to contact the mobile node, it looks up the IP address by a simple name lookup from the DNS. Although this works for newly established connections after a handover, it cannot be used to inform peers with ongoing connections.

Another solution is to use a dedicated public network node. This network node serves as a proxy for the mobile node. Peers communicate with the mobile node always through this proxy. The proxy knows about the location of the mobile node and relays the data traffic between the peer and the mobile nodes current location. The mobile node registers and updates its IP address at the proxy. For new connection establishment, the mobile node registers with its FQDN either the IP address of the proxy or the some other IP address that is routed through the proxy.

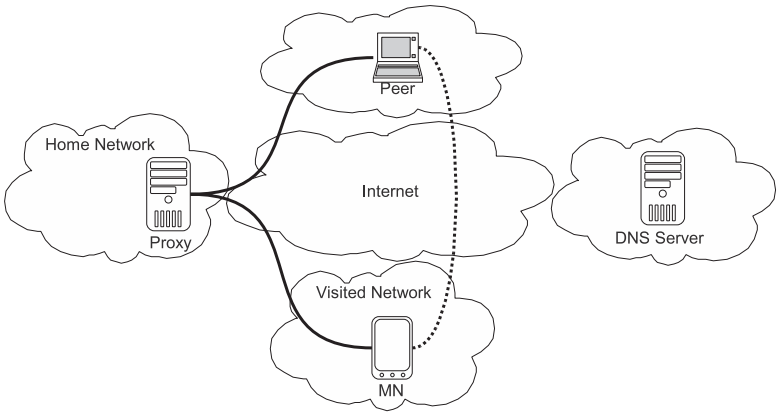

FIGURE 6 Triangular routing and route optimization

The problem with a dedicated proxy node is that the data traffic always goes through it. This may be suboptimal if the proxy is topologically out of the way of the path between the peer and the mobile node. This causes the *triangular routing* effect shown in Figure 6 as the continuous line between Peer-Proxy-MN.

For this reason, the mobile node can register its current IP address directly with its peers, so that data packets travel directly between them and the mobile node (the dashed line in the figure). This is usually called *route optimization* and can only be done for already established communications.

Layers above the Network Layer tend to depend on the stability of IP addresses provided by static IP networks without mobile nodes. Most of the Transport Layer protocols use IP addresses as part of their session identifiers, for example, TCP connections are represented by a (source IP address, destination IP address, source port, destination port) tuple. Only IP datagrams with the parameters defined in the connection tuple are handled (transmitted or received). If any of these components change during the lifetime of the connection, the TCP connection breaks.

2.3.2 Layers of Mobility Management

Mobility can be handled at different layers of the network stack. The main difference is, the lower the layer, the more general the solution becomes. By *general*, we mean the range of network traffic to which the solution is applicable. For example, a Network Layer solution usually applies to all the ULPs. This means that ULPs do not need to care about handling mobility because the Network Layer takes care of it. On the other hand, an Application Layer solution usually applies only to those applications that explicitly take it into use (e.g., via a shared library).

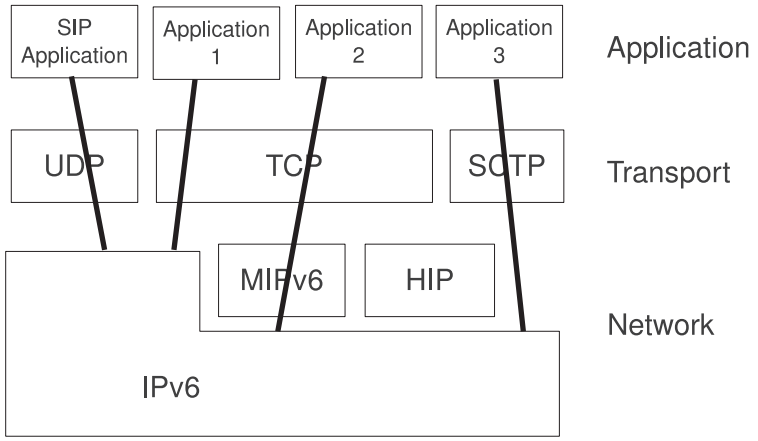

FIGURE 7 Layers of Mobility Management

The Session Initiation Protocol (SIP) provides Application Layer mobility [54]. At the Transport Layer, mobility can be achieved by extending TCP. Other solutions add mobility capabilities to Stream Control Transmission Protocol (SCTP), Datagram Congestion Control Protocol (DCCP). Network Layer solutions include Mobile IPv6 and the Host Identity Protocol (HIP).

Figure 7 shows the network stack of a node that supports mobility management at different layers. The *SIP Application* uses User Datagram Protocol (UDP)

with pure IPv6 and handles mobility with SIP. *Application 1* uses TCP with pure IPv6, therefore it cannot handle mobility events. For *Application 2*, mobility is handled at the Network Layer; it uses TCP on top of Mobile IPv6 (MIPv6). *Application 3* relies on the mobility management capabilities of SCTP in the Transport Layer.

In theory, mobility management protocols may be stacked on top of each other. For example, an application may use SCTP on top of MIPv6, with MIPv6 being on top of HIP with IPv6. Although such a scenario is possible, it is yet to be seen whether this is a useful use case.

2.3.3 Locator and identifier separation

In today's Internet, IP addresses are used both as identifiers of end-nodes, and as network layer locators used for routing from source to destination. The *identifier* aspect of addresses is required by ULPs, e.g., TCP. This ULP identifier cannot change during the lifetime of a ULP session, unless the ULP supports this. On the other hand, the *locator* aspect of the addresses is required only for the routing of packets. Therefore, it is a logical step to separate these two aspects. This makes it possible to provide a stable identifier to ULPs, while we do not need to worry what locators are used underneath to route their traffic [51]. In order to provide transport layer survivability, in the face of renumbering events, all the network layer mobility and multihoming solutions employ some kind of separation between locators and identifiers.

Although the separation of locators and identifiers provides transport layer survivability, the sudden change in the locators underneath ULPs can have negative effects on ULP behaviour, e.g., congestion control. This affects also traffic engineering where the packets of the same ULP session are simultaneously spread over multiple paths by varying the locators for outgoing traffic ([37], [28] and [25]). For this reason, this kind of load spreading is discouraged until an accepted solution is found in this area.

2.4 Mobile IPv6

Mobile IPv6 (MIPv6) [27] provides mobility extensions for IPv6. It defines a Network Layer solution for mobility. With these extensions, IP network nodes can roam between different subnetworks.

Figure 8 shows the actors involved in a basic MIPv6 scenario. The Mobile Node (MN) is associated with a Home Network. This is the default location of the MN. The Home Agent (HA) is a router in the Home Network. The Correspondent Node (CN) is a network node somewhere in the Internet. The MN is assigned an IP address from the Home Network, called the Home Address (HoA). The HoA serves as the stable Network Layer identifier and locator of the MN.

When the MN is at home, data traffic flows to and from the CN as in a

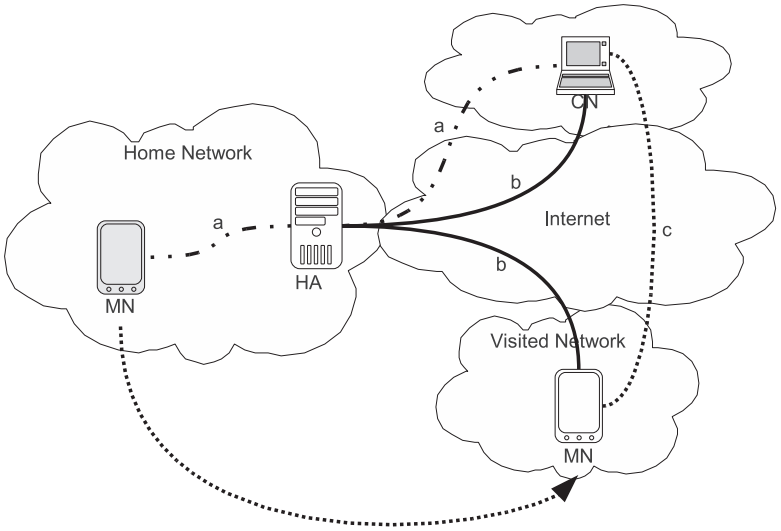

FIGURE 8 Basic MIPv6 topology

non-MIPv6 scenario. The CN sends IP datagrams to the HoA of the MN. For traffic from the MN, the MN uses its HoA as the source IP address in outbound datagrams. This situation is shown in Figure 8 by line *a*.

When the MN moves into a different IP subnetwork, it is assumed to be assigned a new IP address, the Care-of Address (CoA). The CoA is allocated either by stateless [63] or stateful [17] means. This CoA then has to be registered at the HA. This registration is called Binding Update (BU), and it binds the HoA to the CoA. This binding causes the HA to work as a proxy for datagrams flowing between the MN and the CN.

Figure 9 shows how the proxying is done. The arrows represent a transfer of an IP datagram. The labels above the arrows show which IP addresses are used for the transmission at each step. Two labels mean tunneling (outer IP addresses are shown in the first line, inner IP addresses are shown in the second line).

Datagrams from the CN to the MN, have the HoA of the MN in the destination address field of the IP header. These datagrams are then routed to the Home Network and eventually end up at the HA. The HA encapsulates the datagrams in a tunnel (either IPv6-in-IPv6 or IPsec). The tunneled datagrams are sent from the HA to the MN, therefore the outer IP header has the CoA set as the destination. The MN, upon receiving the tunneled datagrams, decapsulates them. This decapsulation process happens inside the Network Layer and the result is the original IP datagram that the CN has sent. This datagram is then passed to ULPs. Line *b* in Figure 8 shows the path of datagrams. Figure 9 shows how IP datagrams are transformed as they travel between the CN the HA and the MN. When datagrams are sent from the MN to the CN, a reverse tunneling is performed, that is, every datagram is sent via the HA towards the CN.

The tunneling makes sure that ULPs above the Network Layer are not af-

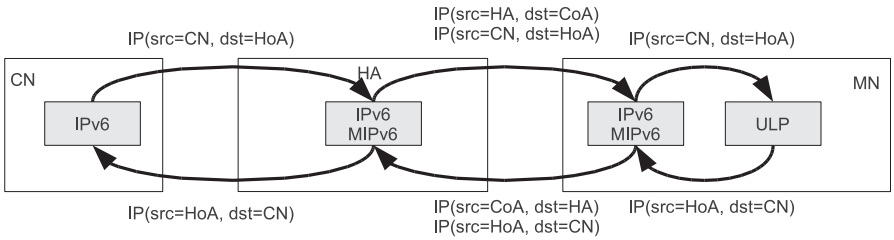

FIGURE 9 IP header transformation and tunneling with MIPv6

fected by the changes in the CoA. ULPs only see the stable HoA, which is translated to the proper CoA by the MIPv6 layer.

MIPv6 supports Route Optimization (RO). With RO, the MN can explicitly register its CoA with the CN. For this, the CN has to have support for MIPv6. RO is performed with a Binding Update between the MN and the CN. After the BU, the CN has a (HoA, CoA) associated with the MN. From then on, data traffic can bypass the HA and can flow directly between the CN and the MN.

Line *c* in Figure 8 shows the path of route optimized datagrams. Figure 10 shows how the IP header is changed when RO is used between the CN and the MN. These datagrams are also tunneled between the CN and the MN, just like in the un-optimized case, in order to hide the unstable (possibly changing) CoA from ULPs. The tunneling is performed by two different IPv6 extensions. One is a new Destination Option, the HoA Destination Option (HoADO) which is used for MN-to-CN traffic and it carries the HoA. The other is a new Routing Header with type 2 (RTHdr2) for CN-to-MN traffic, it also carries a HoA. With this two extensions, tunneling is performed without an extra IP header. The tunnel IP header is replaced by either the HoADO or the RTHdr2.

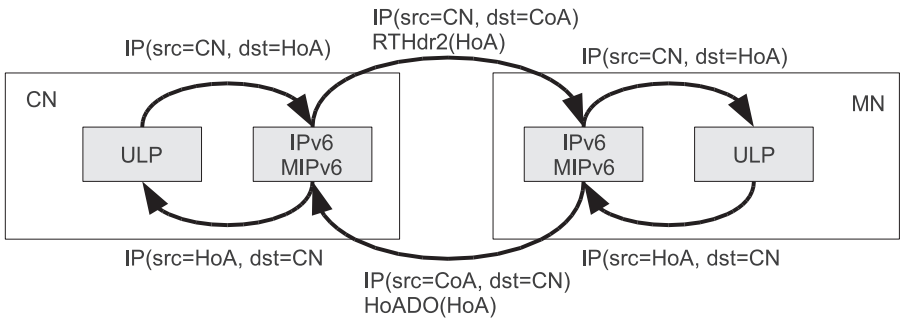

FIGURE 10 IP header transformation and tunneling with MIPv6 and RO

2.4.1 Handover performance of Mobile IPv6

In this section, we look at the handover performance of MIPv6 as analyzed in article [PI].

Note that the base MIPv6 specification allows the MN to bind only a single CoA to its HoA. This means that even if the MN is equipped with multiple interfaces, it can use only one of them at a time for MIPv6 data communication with CNs. The interface that the currently bound CoA belongs to is called the *active interface*.

Handover events may introduce noticeable pauses in data communication. These delays are caused by the operations the MN has to perform during handovers. Figure 11 shows these operations.

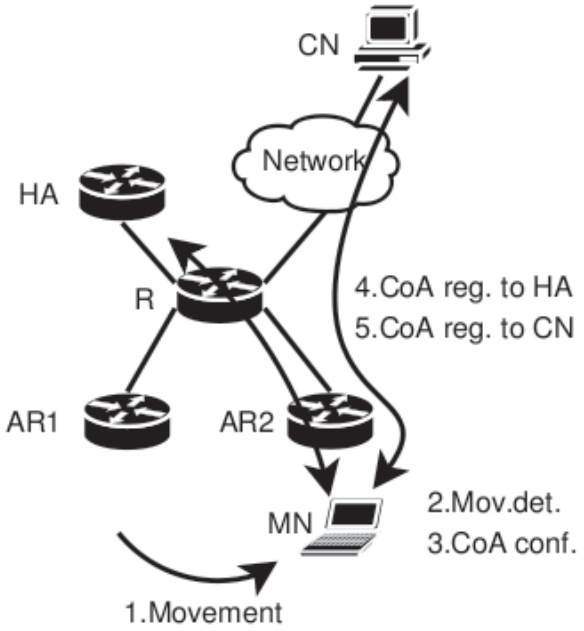

FIGURE 11 Steps of a MIPv6 handover

- Movement: This step involves the change of the Access Point. It can either be a horizontal handover, leaving the active interface unchanged; or it can be a vertical handover, with another interface becoming the active one.
- Movement detection: Also known as Detecting Network Attachment (DNA). The MN needs to detect whether it has ended up in a different IP subnetwork. If the subnetwork does not change then the CoA stays the same and no further operations are necessary. On the other hand, a subnetwork change involves a CoA change which necessitates further steps to regain IP connectivity. The movement detection is done by discovering the subnetwork information via Router Solicitation and Router Advertisement messages [44].
- CoA configuration: The MN configures a CoA on its active interface based on either DHCPv6 or stateless address autoconfiguration. This step also

involves usually the configuration of the default router in the routing table and the DNS servers.

- CoA registration to HA: The CoA registration involves a Binding Update and Binding Acknowledgement message exchange between the MN and the HA. It takes at least one round-trip time (RTT). There is an additional delay caused by the HA switching to proxy mode when the MN moves away from the Home Network to a visited network.

- CoA registration to CN: This is an optional step for Route Optimization. In a worst case scenario, when the CN has no prior binding for the MN, this step involves the Return Routability procedure and a Binding Update/Binding Acknowledgement exchange. The Return Routability consists of two simultaneous message exchanges, the Home Test and the Care-of Test exchange. The entire CoA registration to the CN takes around 1 RTT + max(MN-CN RTT, MN-HA-CN RTT).

In [PI], we conducted some experiments with MIPv6. The goal was to measure the performance of handovers and their effects on data traffic. The tests covered vertical and horizontal handovers, soft and hard handovers. We used three different access technologies, IEEE 802.3 Ethernet, IEEE 802.11 WLAN, and Bluetooth.

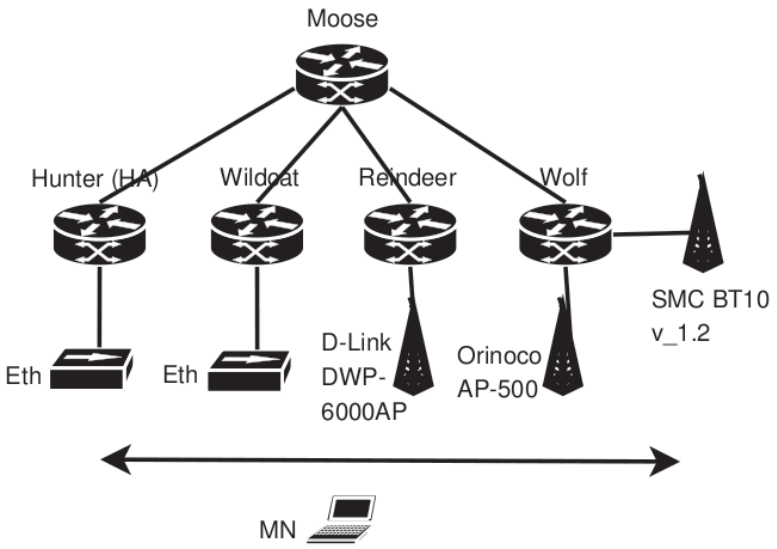

FIGURE 12 MIPv6 testbed

Figure 12 shows the test network topology. The MN was equipped with all three network interfaces (Ethernet, WLAN and Bluetooth). Each router had multiple Ethernet interfaces and every line in Figure 12 represents a separate IP subnetwork. One of these routers acted as the HA. All the network nodes were Linux based, the MN and the HA were using a freely available Linux implementation

TABLE 1 Time taken by MIPv6 handover operations

		Mov. det.	CoA conf.	CoA reg.	HO delay
Ethernet-Ethernet					
	HN->FN	0.96	1.61	1.01	3.58
	FN->HN	1.00	1.25	0.11	3.36
WLAN->WLAN					
	FN->FN	1.31	1.27	0.02	2.60
	FN->FN	1.71	1.32	0.02	3.05
Ethernet->Bluetooth					
	HN->FN	0.00	0.01	1.04	1.05
	FN->HN	1.07	2.00	0.12	3.19
Ethernet-WLAN					
	HN->FN	0.00	0.01	1.01	1.02
	FN->HN	0.66	1.74	0.11	2.51

of MIPv6 (Mobile IP for Linux (MIPL) version 2 from the Helsinki University of Technology, now maintained by the Nautilus 6 project [1]).

Even without Route Optimization, the handover delays were considerable, therefore we decided to disable route optimization during the tests. We performed two separate test sets. One was aimed at measuring the time that each step in the handover process took. The other was about measuring the time between the loss and regain of the IP connectivity from the applications point of view (the amount of time for which applications cannot send data due to handovers).

Table 1 shows the test results for the handover step measurements. The first two tests were emulating horizontal hard handovers between two Ethernet Access Points and then two WLAN Access Points. In these tests, the Movement Detection (the third column in the Table 1) took around one second, which involved the sending of a Router Solicitation message and waiting for the Access Router to reply with a Router Advertisement. According to [44], Access Routers can be configured administratively to react to Router Solicitations quickly, therefore Movement Detection delay may be reduced further. On the other hand, there are CoA Configuration and CoA Registration that need to be performed as well. In our tests, CoA Configuration was done by stateless address autoconfiguration [63] which involves the Duplicate Address Detection (DAD) procedure. DAD is used to make sure that no two nodes in the LAN use the same IP address and takes around another one second (the MN sends a Neighbor Solicitation and waits for a corresponding Neighbor Advertisement message). The CoA Registration involved the exchange of the Binding Update and Binding Acknowledgement messages between the MN and the HA.

Similar tests were performed between different access technologies, emulating vertical handovers. In these tests, movement detection time is zero or negligible when done from the home network to the visited network. This is because the new active interface (the one at the visited network or foreign network) was

TABLE 2 Gaps in application data communication during MIPv6 handovers

		Ping	TCP	UDP
Ethernet-Ethernet				
	HN->FN	10.30	6.25	6.33
	FN->HN	13.50	6.43	5.36
WLAN-WLAN				
	FN->FN	7.30	9.89	4.88
	FN->FN	4.80	6.56	4.28
Ethernet-Bluetooth				
	HN->FN	2.10	-	-
	FN->FN	1.28	-	-
Ethernet-WLAN				
	HN->FN	1.90	8.60	1.06
	FN->HN	1.00	0.74	0.03

already configured when the home interface was disabled (the Ethernet cable unplugged). The movement to the visited network was a hard handover. On the other hand, when moving back to the home network by enabling the home interface (soft handover), Movement Detection and CoA Configuration had to be performed.

Table 2 shows how application data communication is affected by the handovers. We measured the delay with ping, with a UDP data stream and with TCP data stream between the MN and the HA. Tests were done for both vertical and horizontal handovers. In all cases, the delay caused by handovers were multiple seconds.

Unfortunately, the used MIPv6 implementation was in experimental phase, therefore it may be likely that a more optimized version would perform better. Although, it is visible that artificial and configurable delays in the Movement Detection and CoA Configuration process can also help to reduce handover delays.

When a MN is equipped with multiple network interfaces another possibility arises for reducing handover delays, namely proactive handovers. The MIPL code we used in these tests has support for multiple interfaces. Every interface can be assigned a priority value. When an interface becomes available or unavailable, the active interface may be changed to one that has a higher priority. The handover process is reactive, it is initiated when the current active interface becomes unusable. On the other hand, if the MN acts pro-actively, some steps of the handover process can be initiated before the IP connectivity via the active interface is lost. This pro-activeness can be built into MIPL by actively monitoring link characteristics of the available access technologies. Also, a more sophisticated policy based interface selection can be implemented that takes into account fine grained user preferences. Chapter 3 explores these possibilities.

2.4.2 Performance related Mobile IPv6 extensions

There are several efforts for improving the scalability and performance of MIPv6 handovers. Especially, the IP connectivity loss caused by the MMP handover delay is a serious issue. Such a delay affects the usability and the user experience of networked real-time, voice (VoIP) and video (IPTV) communication, that seem to be crucial in current and future IP networks.

2.4.2.1 Hierarchical Mobile IPv6

One way to reduce the handover delay of MIPv6 is by shortening the path to which Binding Updates must be sent and by decreasing the signaling overhead caused by these Binding Updates. Hierarchical MIPv6 [55] is an extension for MIPv6 that has the above goals. It provides reduced handover signaling and handover delays. Figure 13 shows the topology of a basic HMIPv6 scenario. A new actor is introduced, the Mobility Anchor Point (MAP). The MAP is a router located in an administrative domain, the MAP domain. Inside the MAP domain, every AR advertises the MAP in extended Router Advertisement messages. The MN generates a Regional CoA (RCoA) with stateless address autoconfiguration based on these RAs. It also acquires a Local CoA, which is synonymous to the CoA of MIPv6, from the visited network. The LCoA is bound to the RCoA at the MAP. Similarly, the RCoA is bound to the HoA at the HA. The end result of this is a tunnel between the HA and the RCoA (dotted line *a*) and a tunnel between the MAP and the LCoA (dotted line *b*).

When no route optimization is used, traffic between the MN and the CN follows the path marked with *a* and *b*. Route optimization can be done in two different ways. The MN can bind either the RCoA (path *c* and *b*) or the LCoA (path *d*) with its HoA at the CN.

When the MN moves inside the MAP domain, it updates the (LCoA, RCoA) binding at the MAP. No update needs to be sent to the HA. If the MN has registered an (RCoA, HoA) binding at the CN then no update needs to be sent to the CN either.

2.4.2.2 Fast Mobile IPv6

Fast Mobile IPv6 [32] (FMIPv6) defines extensions for MIPv6 to reduce the time it takes for the MN to regain its IP connectivity after handovers. This is achieved by acquiring information about the New Access Router (NAR) where the MN intends to move. This information is requested by the current Access Router of the MN. The current Access Router (Previous Access Router, PAR) has information about nearby ARs. The provided information is used to

- generate a New CoA (NCoA) with the assistance of NAR eliminating or reducing the time taken by the Duplicate Address Detection procedure and
- request the PAR to forward packets destined to the MN to the NCoA.

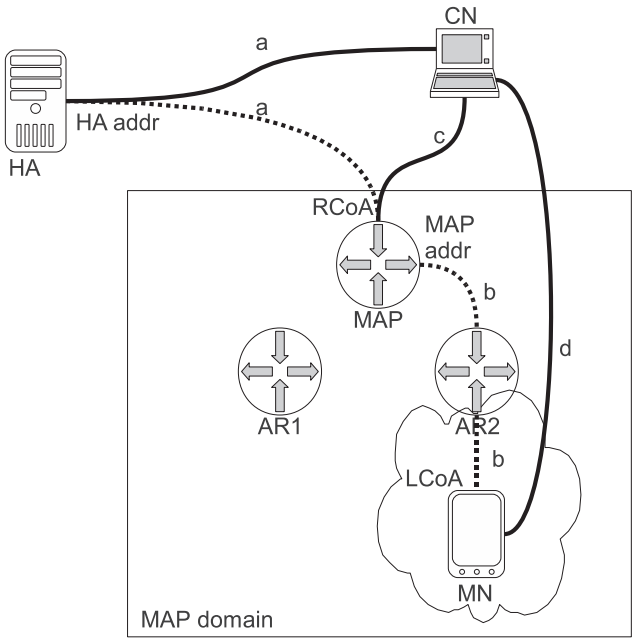

FIGURE 13 Topology and traffic paths with Hierarchical MIPv6

With this proactive approach, Movement Detection is not necessary since the MN already knows the subnetwork where it moves before the handover. Also, CoA Configuration time is either eliminated by generating the NCoA while the MN is still at PAR, or reduced with the assistance of NAR.

Figure 14 shows the actors and data paths involved in a simplified Fast MIPv6 scenario (the dotted paths represent tunnels). Originally, the MN is located at PAR and data packets travel the path a, $b + b'$ ($b + b'$ is the tunnel from HA addr to PCoA at the MN). Either before or after the MN performs a Layer 2 handover to NAR, a tunnel is established from PAR to NCoA (path c). When the FMIPv6 operation is finished, data flows the path a, b, and c. The PAR captures every datagram destined to PCoA and encapsulates them in the PAR-NCoA tunnel, therefore on path c, the datagram that was sent by CN is encapsulated twice (once by the HA on path b and once by the PAR on path c). Datagrams from the MN to the CN travel the same, but reversed, tunneled path.

At this point the MN regained its IP connectivity without updating its binding either at the HA or the CN, so the CoA Registration delay was eliminated.

The above mentioned scenario demonstrated Proactive, Mobile Initiated handovers. FMIPv6 also supports Reactive handovers, when the MN generates the NCoA and requests the PAR-NAR forwarding of datagrams after the handover. Network Initiated handovers are supported by allowing the PAR to send information about a NAR in an unsolicited way to the MN.

For its proactive behaviour, FMIPv6 relies on the assistance of the underlying Link Layer. The MN needs indications from the Link Layer to know when a

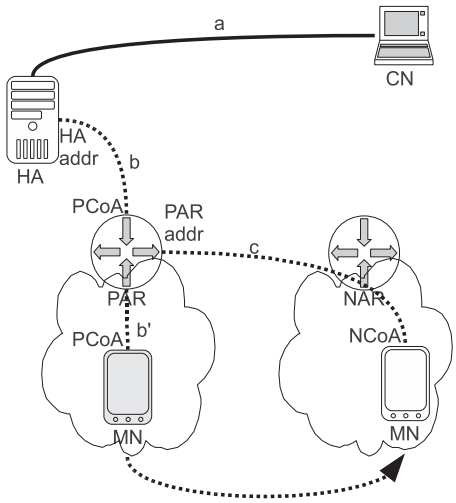

FIGURE 14 Topology and traffic paths with Fast MIPv6

handover may be imminent. This information is then used to acquire the information about neighboring Access Points and Access Routers in order to achieve a proactive handover. The type and timing of these Link Layer information largely depends on the underlying Access Technology.

For this reason there exist several IETF documents that describe the interaction of FMIPv6 and various Access Technologies. Different scenarios for handovers between WLAN APs are suggested in [38]. In [26], the authors discuss Link Layer indications and control for WiMAX (IEEE 802.16e). [71] explains how FMIPv6 may be integrated in 3G CDMA networks and proposes some necessary FMIPv6 extensions.

2.4.2.3 Flow based Fast Handover for Mobile IPv6

Flow based Fast Handover for Mobile IPv6 (FFHMIPv6, [61]) is an extension to Mobile IPv6 to reduce packet loss during handovers.

Figure 15 illustrates the basic idea of FFHMIPv6. Routers employ a routing cache or flow cache that hold information about sources and destinations between which they have recently forward datagrams. For example, routers along path *a* (e.g., RO and RX) contain information about datagrams sent between the HA and the MN in their flow cache.

When the MN changes its topological attachment point to router *RN*, after acquiring the new CoA and performing the DAD process, it sends a BU message to the HA. The FFHMIPv6 MN adds a new IPv6 Hop-by-Hop Options header (see [14] for an explanation of the Hop-by-hop Options Header) that contains, among other information, the new CoA (NCoA). This BU datagram passes the routers along path *b*. Routers along this path (i.e., at every *hop*) process the Hop-by-Hop Option. If path *a* and path *b* join then, at the junction point, there is a

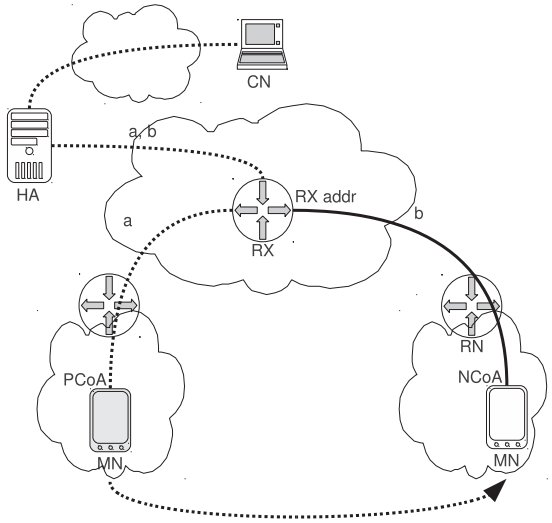

FIGURE 15 Topology and traffic paths with FFHMIPv6

router *RX* that has a flow cache entry that matches the information in the Option. A temporary IPv6-in-IPv6 tunnel is then established between RX (RX addr) and MN (NCoA). The Hob-by-Hop option is then modified by RX to indicate to other routers along path *b* that no further processing is necessary. Then the BU datagram is forwarded to HA as usual. While the tunnel between RX and MN is alive, all datagrams that are destined to the previous CoA (PCoA) of MN are encapsulated and sent to the new CoA over the tunnel. For datagrams sent by the MN, reverse tunneling is performed.

Therefore, FFHMIPv6 makes it possible for the MN to resume its data traffic with CNs and the HA even before the Binding Update procedure has been finished. After the BU procedure has been finished, the tunnel is no longer necessary and can be destroyed.

In case path *a* and path *b* do not join, FFHMIPv6 cannot be used and the handover procedure is done as per Mobile IPv6.

The authors in [61] show that, with regards to handover delay, FFHMIPv6 can outperform plain Mobile IPv6 by two orders of magnitude which is a remarkable gain. On the other hand, in order to reach this path a and b must join and all the involved routers should be able to understand the FFHMIPv6 protocol which significantly reduces its deployability.

2.4.2.4 Candidate Access Router Discovery

Candidate Access Router Discovery (CARD) [36] provides a way for the MN to discover information about neighboring Access Routers.

Figure 16 shows the basic operation of CARD. The MN can request AR information from its current AR about both the Current AR and Candidate ARs. For this, the MN supplies one or more AP identifiers. These AP identifiers (Layer

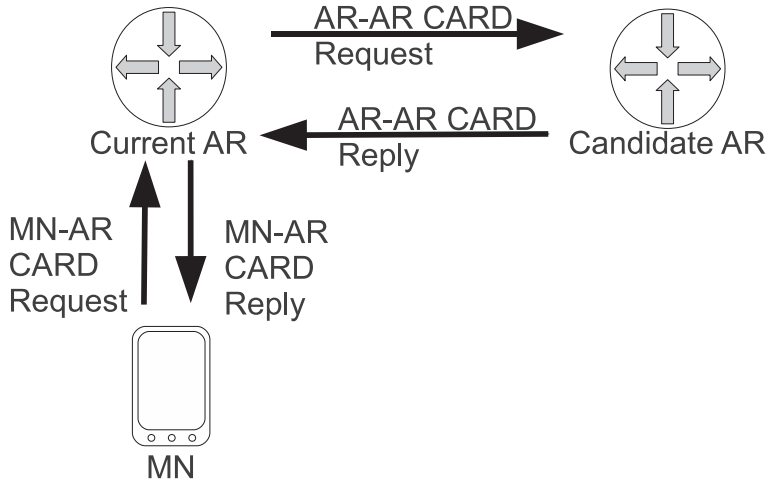

FIGURE 16 Candidate Access Router Discovery message exchanges

2 identifier) are mapped to AR IP addresses by the Current AR. For this purpose, ARs manage an (AP identifier, AR information) mapping table. This table can be either manually or automatically populated. Nevertheless, the maintenance of the contained information is done by the AR by exchanging AR-AR Request/Reply messages with Candidate Access Routers.

Besides IP addresses, AR capability information is also supported (e.g., QoS, bandwidth, etc.). The MN may request information about Candidate ARs that possess certain required capabilities. The capability information may contain dynamic and timely content that needs to be maintained by the Current AR. This is done by AR-AR Request/Reply messages.

The point when the MN requests Candidate AR information from its Current AR is not specified and is expected to be triggered by Link Layer indications.

2.5 Conclusion

In this chapter, we looked at mobility in IP networks. The Internet was designed with stationary nodes in mind therefore, the addition of support for mobile nodes (and mobile networks) introduces solutions that are inevitably disruptive. These solutions tend to involve various layers of the network stack in order to be effective. Also, an old issue with the design of IP networks surfaces again, namely the double role of IP addresses as both locators and identifiers [51]. Mobile IPv6 provides a solution for network node mobility but in its base form it is not efficient enough to provide seamless handovers that is demanded by increasing number of mobile devices. Therefore various extensions and enhancements exist that try to address the shortcomings of the base Mobile IPv6 protocol. The prototype

VERHO system, which is the result of the work described in Chapter 3, is based on Mobile IPv6.

3 MULTIPLE INTERFACE MANAGEMENT

Mobile devices (smartphones, notebooks, netbooks, etc.) are being equipped with multiple network interfaces, each corresponding to a different Access Technology. For example, in 2009, most high-end mobile phones already had Bluetooth, WLAN and GPRS access, while notebooks had Ethernet, Bluetooth and WLAN. Some manufacturers already started shipping devices with WiMAX interfaces. The assumption is that mobile devices will exist (often already do) in a heterogeneous wireless environment, sometimes called a *wireless overlay network*. Figure 17 shows a mobile device roaming in such a heterogeneous environment.

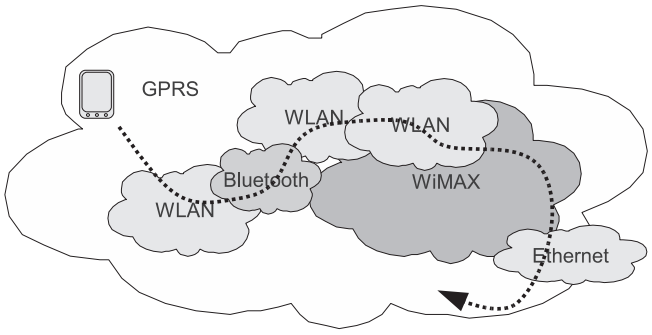

FIGURE 17 Mobile device in a heterogeneous network environment

People use the Internet for more and more tasks in their daily lives, checking bus schedules, navigating in the city with a networked map application, checking weather forecast, uploading pictures and messages to their favorite social networking site, using VoIP in wireless hotspots, etc. This is already a reality. With the proliferation of wireless hotspots, users get closer to the point when they can be connected to the network continuously.

In this chapter, we assume that a mobile device is

- equipped with multiple network interfaces of different technologies,

- uses only one of the available interfaces (the *active interface*) for data communication,

- may use multiple interfaces when performing MMP handovers.

With multiple choices in access technologies comes a management burden. Mobile devices need a way to intelligently decide which network they connect to at any given time when more than one is available, i.e., when the mobile device is in the coverage of various access technologies simultaneously. Different technologies have different characteristics with regards to bandwidth, QoS, price, etc. This necessitates that the network selection take into account user preferences.

Another issue is presented by vertical handovers. When the mobile device changes its active interface, the network characteristics of the new link (bandwidth, delay, etc.) may be substantially different from the previous one that it can visibly affect networked applications running in the device.

A system, therefore, would be beneficial that could allow the management of the available network interfaces and which allows applications to adapt to the changing network environment.

3.1 Trends and standardization efforts

There are various standardization bodies that deal with issues related to the handover and network interface management and utilization of link layer information in ULPs. In [PV], we surveyed the most relevant and active research in this area. It gives an overview of the relevant standardization work and their relation to each other.

In legacy cellular and wired networks, network nodes resided in a mostly homogeneous network environment with regards to Quality of Service, network security and bandwidth. This has started to change. Nowadays, the trend is to move towards an all-IP network environment. This new environment constitutes of IP networks managed by different service providers and/or operators. Users connect to the network via a multitude of possible access technologies (e.g., WLAN, WiMAX, GPRS, CDMA2000). From a mobile devices (i.e. user) point of view, the single important thing is the kind of service it gets when it connects to the network via a particular available technology. ITU (International Telecommunication Union) defines the Next Generation Network (NGN) as a packet switched network with end-to-end Quality of Service utilizing multiple access technologies [69]. This is a general definition and the tools (i.e., the standards and methods) to actually implement are chosen from among various alternative approaches.

The most relevant sources of possible standards are the IETF (Internet Engineering Task Force), the IEEE (Institute of Electrical and Electronics Engineers) and 3GPP (3rd Generation Partnership Project).

3.1.1 Internet Engineering Task Force

The Internet Engineering Task Force (IETF) is a non-profit, open organization of volunteers where anyone can contribute. The decisions are made upon merit and working implementations. The IETF consists of *Areas* and each Area contains *Working Groups*. NGN incorporates native support for mobile network nodes. With regards to this, the IETF has developed the Mobile IP protocol family, Mobile IPv4 [49] and Mobile IPv6 [27], that handles network node mobility at the Network Layer. NEMO (Network Mobility Basic Support) [15] is an extension of Mobile IPv6 to support not only mobile nodes but entire mobile networks. The expired MONAMI6 (Mobile Nodes and Multiple Interfaces) and the newer MEXT (Mobility EXtensions for IPv6) working groups are chartered to study and develop Mobile IPv6 extensions to support multihomed mobile nodes as [66] and [56]. As an alternative to Mobile IP, the HIP (Host Identity Protocol) charter works on the HIP protocol [43] and related extensions that solve the mobility and multihoming issues in an architecturally cleaner way as a side effect of introducing a Host Identity Layer between the Transport and Network Layers. The, now concluded, DNA (Detecting Network Attachment) working group was responsible for studying optimizations [33] for detecting subnetwork changes due to topological movements of the mobile node in the Network Layer, as discussed earlier.

3.1.2 Institute of Electrical and Electronics Engineers

The Institute of Electrical and Electronics Engineers (IEEE) is a non-profit standardization organization, mostly known for their work done on 802 standards such as 802.3 Ethernet, 802.11 WLAN, 802.16 WiMAX and 802.20 Mobile Broadband Wireless Access. Besides these, another relevant research area is the 802.21 Media Independent Handover (MIH) framework [2]. MIH provides the necessary Access Technology independent abstractions that are needed in order to perform optimized vertical handovers in multiple interfaced mobile devices across 802 and cellular networks.

The handover process may be divided to the following three phases:

- Handover Initiation: Search for a new link. Involves network discovery, network selection and handover negotiation.
- Handover Preparation: Setup new link. Involves Layer 2 and IP connectivity setup.
- Handover Execution: Connection transfer. Involves the tasks of the Mobility Management Protocol (e.g., Mobile IPv6).

MIH is concerned with the Handover Initiation and the Handover Preparation phases.

A Service Access Point (SAP) interface is added to the link layer implementation of every supported access technology This interface is used to gather and

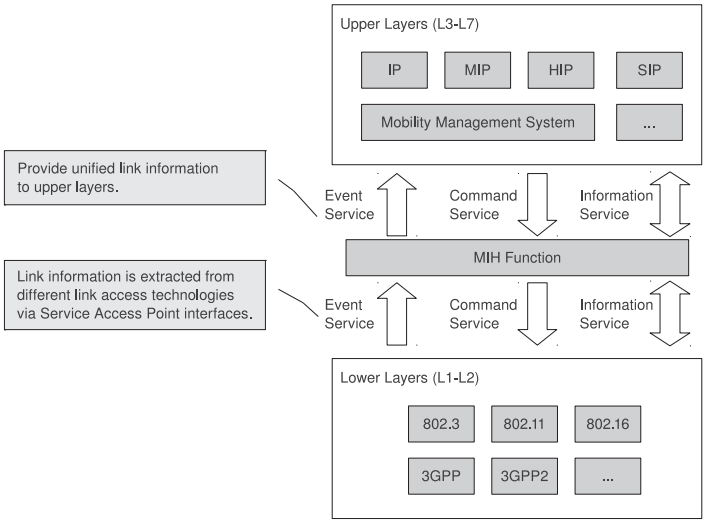

FIGURE 18 MIH function in a mobile node.

extract link information required for the operation of the MIH function. A simple illustration of a mobile node with MIH function is shown in Figure 18.

The MIH function provides the following services:

- MIES: Media Independent Event Service: Provides Layer 2 triggers and events. Some of the proposed events are shown in Table 3. The information is extracted from Link Layers via SAPs and are provided to ULPs via the MIES.

- MICS: Media Independent Command Service: Provides tools for sending commands from higher layers to lower layers. Some of the proposed commands are shown in Table 4.

- MIIS: Media Independent Information Service: Provides information about the available networks around the mobile node. Helps with network discovery and selection. With this service, information about networks belonging to different access technologies can be gathered via the current active link of the mobile node. That is, no need to switch to WiMAX to get information about WiMAX. The returned information is unified across different networks. Some of the information elements of MIIS is shown in Table 5.

For full support of MIH, also the Access Points are equipped with MIH functions. The MIH communication between mobile nodes and APs is done over Layer 2. APs exchange MIH related information between each other via Layer 3. A MIH Information Server node may also exist somewhere in the network that implements policies and decision making about mobile node handovers. MIH communication between APs and Information Servers, mobile nodes and Information Servers is done via Layer 3.

TABLE 3 Media Independent Event Service Layer 2 information

Event Type	Event Name	Description
State Change	Link Up	Layer 2 connection established
State Change	Link Down	Layer 2 connection broken
State Change	Link Detected	New link has been found
State Change	Link Parameters Change	Change in specific link parameters has crossed pre-specified thresholds (link Speed, Quality metrics).
Predictive	Link Going Down	Layer 2 connection break imminent
Link Synchronous	Link Handover Imminent	L2 intra-technology handover imminent (subnet change). Notify Handover information without change in link state.
Link Synchronous	Link Handover Complete	Notify handover state

With regards to who makes the handover decisions, there are three types of handovers:

- Terminal Controlled: The mobile node uses the MIH services to make its own handover decisions based on the gathered information (via MIES).
- Terminal Initiated, Network Assisted: The mobile node uses the MIH Information Service to get information about neighboring access points. Upon this information it makes it own handover decisions.
- Network Initiated, Network Controlled: The network uses the MIH Event and Command Services and Information Service and decides if a handover is necessary, choose the target access point/network and command the mobile node to handover.

3.1.3 3rd Generation Partnership Project

The 3rd Generation Partnership Project (3GPP) was established in 1998 as a group of telecommunications associations to produce technical specifications and reports for a 3G mobile system based on GSM (Global System for Mobile communication). Along the way it picked up the maintenance and development of the GSM, GPRS EDGE standards and reports. 3GPP publishes the produced specifications as *releases*.

Legacy and current 3GPP cellular networks are self contained in the sense that they solve the issues related to mobility, security, QoS and charging in their own specialized operator centric way. Mobile nodes have little say in handover decisions which is performed by nodes of the infrastructure (e.g., Base Stations).

TABLE 4 Media Independent Control Service commands

Command	Interacting nodes	Description
Handover Initiate	Client-Network	Initiates handover and sends a list of suggested networks and suggested Access Points.
Handover Prepare	Network-Network	Sent by MIHF on old network to MIHF on suggested new network. Allows the client to query for resources on new network and also allows to prepare the new network for handover.
Handover Commit	Client-Network	The client commits to do the handover based on selected choices for network and AP.
Handover Complete	Client-Network, Network-Network	A notification from new network AP to old network AP that the handover has been completed, new AP has been established and any pending packets may now be forwarded to the new AP.

TABLE 5 Media Independent Information Service information elements

Element	Description
List of available networks	List of available network types (e.g., 802.11, GSM, etc.)
Location of PoA	Geographical location of Point of Attachment (PoA), PoA identifier
Operator ID	Network provider name
Cost	Cost of network/Service usage
Security	Link layer security (e.g., cipher suites, authentication methods)
QoS	Link QoS parameters

3GPP Release 99 consists of a UMTS (Universal Mobile Telecommunications System) core network that is a mixture of circuit and packet switched technologies. This is about to change in the very near future. One of the main goals of further releases is the simplification of the core network. This means a move to an all-IP based solution. The current feature complete release is Release 8 (R8). R8 specifications were frozen in December 2008. One of the main components of R8 is LTE (Long Term Evolution). LTE is advertised as the 4G mobile network. It provides enhancements for UMTS and promises increased speed and capacity with an all-IP core architecture.

System Architecture Evolution (SAE) is a part of LTE. It is basically the IP-based core network architecture. In contrast with older releases, it allows mobility between 3GPP, non-3GPP and also legacy networks (e.g., GSM/GPRS, WiMAX, 3GPP2, etc.). IETF protocols, such as Mobile IPv4, Mobile IPv6, will be used for non-3GPP mobility management. Support for multihoming and simultaneous multiple access is likely to be brought by the efforts of the IETF MEXT working group.

3.1.4 Other related work

Multiple interface and handover management has been an active research area. Various research projects existed that looked at this issue. In this chapter, some of these are introduced.

The authors in [41] take a look at Mobile IPv6 and its two extensions, Hierarchical Mobile IPv6 ([55]) and Fast Mobile IPv6 ([32]). Enhancements are proposed in order to reduce handover latency and packet-loss in the form of anticipated handovers. For this purpose, Link Layer indications and triggers are used for IEEE 802.11 WLAN.

The Unified Link Layer API (ULLA) [67] implements an application programming interface for the creation of link-aware applications. It provides comparable, technology and platform independent information about Link Layer parameters. This information can be utilized in, for example, making handover decisions in a heterogeneous network environment. An implementation of MIH [2] is described based on ULLA.

A mobility management system is introduced in [42]. The authors suggest the utilization of Link Layer information (e.g., triggers and hints) for anticipating Mobile IPv6 handovers. Network interface selection is based on user specified priority values.

A user-centric interface management solution is described in [46]. A power-saving multiple interface management policy is introduced. The network selection is based on user preferences. The authors in [12] extend the Media Independent Handover [2] with power management functionalities and present a method to take power consumption into account when making interface selection decisions.

3.2 Overview of the VERHO system

The VERHO system, as described in [PII] and [PVI], provides a policy driven, multiple interface management for mobile devices that allows networked applications to adapt to changing network conditions.

VERHO is based on a component architecture, shown in Figure 19. It consists of the following main components:

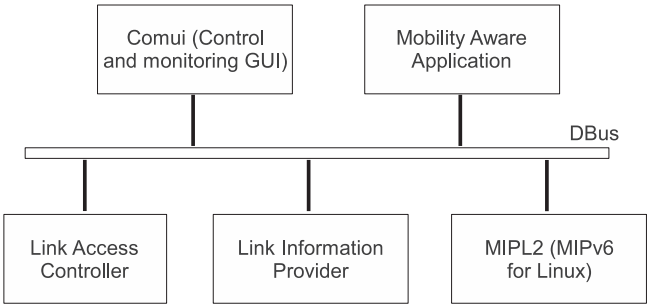

FIGURE 19 VERHO component oriented architecture

- Link Information Provider: LIP is responsible for gathering and monitoring information about the supported network interfaces. It provides a unified notification API over DBus to the other components. LIP is described in Section 3.3.
- Link Access Controller: LAC takes input from LIP and the user preferences. Based on its input it initiates MIPv6 handovers by controlling the MIPL2 component. A handover results in a change of the active interface. LAC is described in Section 3.4.
- MIPL2: MIPL2 [1] is an implementation of the MIPv6 protocol for Linux by the Helsinki University of Technology. It was extended with a DBus interface to be controlled by the LAC component.
- Comui: The control and monitoring user interface. Users can monitor and affect the behavior of the system with this application. For example, modify (add/rename/change) policies, enable or disable network interfaces, etc. Comui is described in Section 3.5.
- Mobility Aware Application: This can be any application that reacts to mobility events. The mobility events are broadcast over DBus by LAC and LIP. For demonstration purposes, several simple adaptive applications were developed as shown in Section 3.5.

Each of these components runs as a separate process. They communicate over an interprocess communication system, called DBus. This component based architecture makes it possible to use the same system to control handovers with other

mobility management protocol than MIPv6 (by possibly replacing the MIPL2 component).

The system allows end-users to express their preferences with regards to the available access technologies. For this purpose, a policy and profile based approach was designed. The user can create different profiles. Only one profile can be active at a time. Different profiles make the system to behave in different ways. A profile is described by a set of link properties (or characteristics) and associated weights. For example, a profile called "Max Speed" can be defined by specifying a high weight on the "Bandwidth" link property.

LAC takes the information from LIP and controls the MIPL2 component accordingly. This process requires the system to pass information between the network layers. MIPL2 works at the Network Layer, while LIP works at the Link Layer. Furthermore, user applications (Application Layer) can also benefit from the information provided by LIP and LAC. This makes the system to have a cross-layer design, as shown in Figure 20.

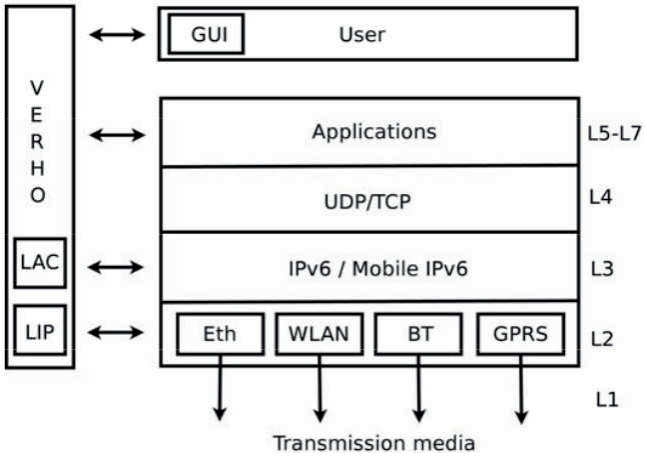

FIGURE 20 VERHO cross-layer design

With the VERHO system, a multi-interfaced mobile node can accomplish soft vertical handovers. This is done by having the MN attached to two different links on two different network interfaces while performing MIPv6 handovers.

The information provided by LIP is utilized to achieve proactive handovers. This is done by actively measuring the link properties (e.g., signal strength) and based on the current policy profile the LAC can initiate a handover prior to the active interface becoming unavailable for IP communication.

3.3 Gathering Link Information

One core component of the VERHO system is the Link Information Provider (LIP). LIP is described in greater details in [PIII] and [PVI].

The task of LIP is to extract information about the different available access technologies. Figure 21 shows the architecture of LIP. It consists of two main modules. The Link Module (LM) extracts information about the state of the network interfaces and the attached links. The Access Point Module (APM) periodically scans for neighboring Access Points on each wireless network interface. In the prototype implementation, the system supports Ethernet, WLAN and Bluetooth access technologies and is modular enough that support for others can be easily included.

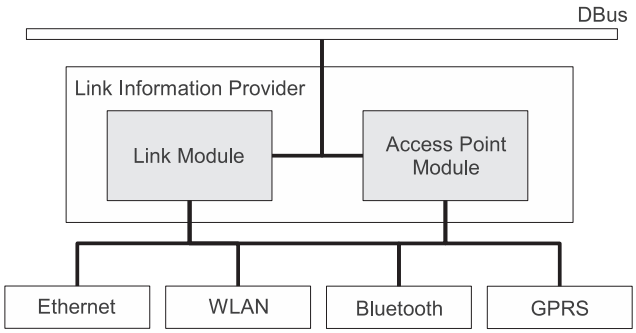

FIGURE 21 LIP architecture

The information gathered is distributed over DBus where other components, clients, (e.g., LAC or some mobility aware application) can utilize it. Link information consists of *events* and *properties*. Events may be classified in different ways. For example, based on the strength of its indication:

- Triggers: A message that informs about the occurrence of an event. At this point the event has already happened and clients may need to take immediate actions.
- Hints: A message that informs about events that may or may not need immediate action from clients.

Different access technologies provide different information, some of which is common to all. Usually, all the technology specific information is a Hint. Triggers are extracted and generated from Hints.

The most important triggers are the ones that signal the availability of a link:

- Link Up: Indicates that the link is capable of carrying IP datagrams. The link may not have a valid IP configuration yet (e.g., IP address). Usually it means that the Mobile Node has successfully associated with an Access Point on the given link (i.e., network interface).
- Link Down: Indicates that the link cannot be used to carry any network packet.

Link properties can be classified based on their dynamism:

TABLE 6 Unified Link Properties

Property Name	Dimension
Signal Strength	Integer value between 1 and 5
Tx Power Level	Converted to dBm
Bitrate	Converted to kbps

- Static: Properties that cannot or do not change during the lifetime of a network interface. For example, the interface name, the interface technology (e.g., WLAN), the MAC address.
- Configurable: Properties that can be modified either dynamically by the system or administratively. For example, interface status (up/down), IP configuration (IP address, routes, etc.), AP identifier, security, price, etc.
- Dynamic: Properties that dynamically change during runtime. For example, signal strength, power level, bit-rate, available bandwidth.

Some of the dynamic properties make up quality related information. For example, signal strength, bandwidth, price, security. These quality related properties are likely to be of interest of end users for defining their preferences for the active interface. For this purpose, LIP tries to provide a set of *unified* link information to clients. This unified information can be interpreted for all the supported access technologies. This allows the creation of technology agnostic mobility awareness.

Table 6, shows the unified information provided by the prototype LIP implementation. This information can be gathered from both Ethernet, Bluetooth and WLAN. In the case of Ethernet, however, the Signal Strength and Tx Power Level values are set to a constant value, the Signal Strength is maximized and the Tx Power Level is minimized.

The Signal Strength property hint is calculated from the technology specific values shown in Table 7. Its dimension reaches from 0 to 4 and is based on experimental measurements. The goal of these measurements were to deduce some minimum and maximum values for each access technology that are then divided to the five possible Signal Strength values. In order to avoid spurious Signal Strength property events when the technology specific value is *jumping* between two neighbor intervals, we use hysteresis. This means that the intervals are made to slightly overlap.

The Link Up and Link Down triggers are derived from the Signal Strength property. Based on experimentation, a Link Up indication is generated when the Signal Strength goes below value 2, and a Link Down indication is generated when it goes above value 1.

The Access Point Module provides information about neighboring Access Points. It may be used, for example, when the MN starts to loose its connection with its current AP (e.g., degrading Signal Strength). The APM manages and updates a list of APs for each wireless network interface. This is done by listening to AP related events from the operating system and performing periodical scans

TABLE 7 Unification of Signal Strength

	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

TABLE 8 Access Point Properties

Access Technology	Property
WLAN	ESSID, MAC addr, channel, bitrate, noise, signal quality
Bluetooth	AP name, MAC addr, link quality, TX power Level
GPRS	AP name, IP protocol

for potential APs. The AP information is made available to clients over DBus, who may use it to trigger layer 2 horizontal handovers on a network interface. Table 8 shows the information provided by the APM.

3.4 Interface Selection

In this section we describe the results of article [PIV] with regards to network interface selection algorithms.

The Link Access Controller (LAC) component of the VERHO system is responsible for selecting the active network interface during runtime. It bases its decision on the input provided by LIP and on user defined profiles.

The LIP information is represented as a vector of unified link properties, for example: $p = (p_1, p_2, p_3, ...)$. In the case of our experimental implementation, these properties are Signal Strength, Transmission Power Level and Bitrate. At any point in time, every available network interface is represented by such a link property vector.

In this section, *network interface*, *alternative* and *row of matrix R* represent the same thing. Furthermore, *preference* and *priority* are the same.

A user profile is represented by a weight vector, $w = (w_1, w_2, w_3, ...)$. LAC takes the weight vector (w) and a property vector (p) and produces a priority value for the corresponding network interface, $prio = LAC(p, w)$. At any point in time, the active interface is the one that has the highest priority value.

In general, every network interface is assigned several variables. With these variables, a fine grained control is provided over how the active interface is selected by the system:

- Priority value,

- Interface State: Enabled or Disabled. Only Enabled interfaces take part in the priority value calculation. That is, Disabled interfaces cannot be active,
- Priority Value Calculation: Automatic or Manual. In the Manual case, the user is responsible to define the priority value. In the Automatic case, the system dynamically calculates it based on the user profile and the LIP input,
- Interface State Management: Automatic or Manual. In the Manual case, the user is responsible to set the Interface State. In the Automatic case, the system dynamically sets the state based on LIP input.

When the priority value of the currently active interface becomes smaller than the value of another interface, LAC instructs the MIPL2 component to accomplish a vertical handover. The system is flexible enough to be configured to accomplish proactive soft vertical handovers. For example, in a wireless environment, the Signal Strength property may be given a high weight which triggers a handover when the MN is likely to leave the coverage area of its AP in the near future.

According to the flexibility or complexity of interface selection algorithms, we may differentiate between the following:

- Static Priority: Every interface is assigned a static priority value. The available interface with the highest priority is selected. We consider an interface available if it can carry IP datagrams.
- Dynamic Priority: Every interface is assigned a priority value at runtime. The priority values are dynamic and are likely to change over time.
  - Single Property: Only one link property is taken into account when calculating the priority order.
  - Multiple Property: Multiple properties are considered for the priority calculation.
- Conditional Dynamic Priority: The priority calculation can be affected with conditions. For example, a user profile may be used only between 9 o'clock and 12 o'clock, otherwise a different profile is used.

The LAC implements a Dynamic Priority Multiple Property algorithm. In [PIV], several interface selection algorithms are analyzed using Matlab. The interface selection is a Multiple Attribute Decision Making (MADM) problem. With regards to the above classification, all the Dynamic Priority Multiple Property algorithms are MADM algorithms. Each link property corresponds to an attribute. Every property is one dimensional and numerical with a minimum and a maximum value. Non-numerical properties are either cannot be considered or must be represented numerically. For example, a dimension of (low, medium, high, very high) can easily be converted to a (0, 1, 2, 3) numerical dimension. On the other hand, a dimension of (purple, red, black) may be difficult to be converted. In general, the analyzed MADM algorithms support only those dimensions whose values can be ordered.

For the discussion of the results, we define some variables:

- q = The number of alternatives, or network interfaces.
- p = The number of properties per alternative.
- R = Is a matrix consisting of q rows and p columns. $R_{i,j}$ is the value of property j of the network interface i.
- w = The property weight vector.
- A_a = A selection algorithm. It takes R and w as input.
- S_a = Is a row vector of R. It is the selected row by a selection algorithm a.
- t = Denotes round t of the comparison.
- $R_{i,pri}$ = A fixed priority value of network interface i.
- R_i = Row i of R. Alternative i.

First, we compared the following algorithms:

- $A_{pri}(R,w) = S_{pri} = R_l$, where $R_{l,pri} = \max_{i=1..q} R_{i,pri}$. Static priority. This algorithm selects the alternative with the highest priority value.
- $A_1(R,w) = S_1 = R_l$, where $R_{l,1} = \max_{i=1..q} R_{i,1}$. Dynamic priority, single property. It selects the alternative with the highest value of property #1.
- $A_{saw}(R,w) = S_{saw} = R_l$, where $\frac{\sum_{j=1}^{p} w_j * {R'}_{l,j}}{\sum_{j=1}^{p} w_j} = \max_{i=1..q} \frac{\sum_{j=1}^{p} w_j * {R'}_{i,j}}{\sum_{j=1}^{p} w_j}$. Dynamic priority, multiple property. Simple Additive Weighting (SAW) method. R' is the normalized R.
- $A_{spw}(R,w) = S_{spw} = R_l$, where $\prod_{j=1}^{p} ({R'}_{l,j})^{w_j} = \max_{i=1..q} \prod_{j=1}^{p} ({R'}_{i,j})^{w_j}$. Dynamic priority, multiple properties. Simple Productive Weighting (SPW) method. R' is the normalized R.
- $A_{top}(R,w) = S_{top}$. Technique of Ordered Preference by Similarity to the Ideal Solution (TOPSIS). See in [24].

For R, we can define an *ideal algorithm* that selects the *ideal alternative*, $A_*(R,w) = S_*$, where $S_{*j} = \max_{i=1..q} R_{i,j}$. The ideal alternative S_* is a composite of the maximum of the property values for each alternative.

At every round t, every selection algorithm was run with the R^t matrix. The property values for every alternative in R^t was randomly generated. The distance from S_* was calculated for every S_a, $d_a = \|S_* - S_a\|$. Both Euclidean and Manhattan distances (norms) and their weighted variants were calculated. The Euclidean norm of vector a is calculated as $\sqrt{\sum a_i^2}$ and as $\sqrt{\sum (a_i * w_i)^2}$ in the weighted case. The Manhattan norm of a is calculated as $\sum a_i$ and as $\sum a_i * w_i$ in the weighted case. The w weight vector was the same for all rounds.

TABLE 9 Average distance from the ideal alternative

	d_{pri}	d_1	d_{saw}	d_{spw}	d_{top}
Manhattan	3.001	2.730	2.746	2.791	2.879
Manhattan (weighted)	1.517	1.471	1.445	1.450	1.480
Euclidean	1.367	1.315	1.312	1.320	1.341
Euclidean (weighted)	1.004	1.017	0.996	0.993	0.999

TABLE 10 Average distance of A_b from the ideal alternative

	d_b
Manhattan	2.432
Manhattan (weighted)	1.302
Euclidean	1.178
Euclidean (weighted)	0.891

Table 9 shows the result of the calculation. Each column represents the measured distances from the ideal case S_*: d_{pri} for Static Priority, d_1 for Dynamic Priority with Single Property, d_{saw} for SAW, d_{spw} for SPW and d_{top} for TOPSIS. The R matrix included four alternatives and six properties, a 4x6 matrix. In theory, a matrix of any size is possible but, at the time of this dissertation, mobile nodes with around 4 interfaces and around 6 link properties were used for prototyping. The table shows that out of the MADM algorithms, the SAW got the closest to the ideal alternative.

A *combined algorithm* (or best algorithm) can be deduced from the results of multiple algorithms. The idea is to calculate the ideal distance for every algorithm and select S_a where d_a is the smallest. So, the combined algorithm $A_b(R, w) = S_b$, where $S_b \in \{S_1, ..., S_l\}$ and $d_b = \min_{i=1,l} d_i$. Table 10 shows the ideal distance calculation for the combined algorithm. Such a combined algorithm may be used in cases where the network interface properties do not change frequently or the algorithm CPU usage is not an issue. The VERHO system is meant for end-user MNs that are likely to be resource constrained and exist in a dynamic wireless environment. Therefore a combined algorithm was not chosen.

In the next calculations, only the following six MADM algorithms were compared:

- SAW: Simple Additive Weighting.
- SPW: Simple Productive Weighting.
- DFT: Distance From Target. Selects the alternative with the smallest weighted Euclidean distance from the *ideal alternative* (S_*).
- DistId: Distance from Ideal. Selects the alternative with the smallest Euclidean distance from the *ideal alternative* (S_*).
- TOPSIS: Technique of Ordered Preference by Similarity to the Ideal Solution, [24].

TABLE 11 Average distance from the ideal alternatives

	AHP	DFT	SAW	DistId	TOPSIS	SPW
%	11.0825	11.9802	25.4166	32.8697	33.3912	33.8959

TABLE 12 Handover rate

	SAW	SPW	DFT	TOPSIS	DistId	AHP
Change Count %	13.525	18.8625	13.8687	13.225	15.7188	47.125
Max Steady Count %	16.5875	15	15.9125	16.4312	15.4563	4.03125

- AHP: Analytical Hierarchy Process, [24].

A random 6x7 R matrix was generated which represents six different network interfaces (or access technologies) and each interface has seven different properties. For the sake of AHP, the weight vector was calculated from a randomly generated *relative importance* matrix. At every round MN movement was simulated by altering slightly the value of every property of every alternative in R. 400 rounds were done with the same weight vector. We call this a *cycle*. Then the weights were changed and the rounds were run again. In total, 40 cycles were done ($400 * 40$ rounds).

Table 11 shows the average distance of each algorithm from the ideal alternatives. At every round, an algorithm calculates a priority value for every alternative in R. The distance from the ideal is calculated as $\frac{prio(S_*)-prio(S_a)}{prio(S_*)}$, where S_* is the ideal alternative, S_a is the selected alternative.

During these rounds, the sensitivity related to property changes was also measured. At every cycle, Change Count shows in percentage how often did the algorithm chose a different network interface (a different row in R). On the other hand, Max Steady Count shows in percentage how long did the algorithm chose the same network interface despite the changing properties. Table 12 shows the results.

This shows us how *well* these algorithms behave in a dynamically changing environment. It is beneficial if an algorithm can minimize the number of handovers (minimize Change Count and maximize Max Steady Count). It can be seen that the SAW algorithm performs well in this regard. When compared to the other algorithms, it is close to the ideal alternative and provides a low handover rate.

Based on these results, in the VERHO prototype implementation, the LAC component uses SAW for calculating the network interface priority values and for selecting the active interface.

3.5 Applications

This section describes in brief the demonstration applications as introduced in article [PVI].

User applications can benefit from the information provided by the LIP and LAC components. The VERHO system includes some demonstration applications for this purpose.

A mobility aware application can use the provided information to adapt to the current link characteristics (e.g. a media player). Information provided by VERHO includes

- Link information from LIP (LM),
- Access point information from LIP (APM),
- Handover indications from LAC.

3.5.1 Control Interface

Comui is the Control and Monitoring User Interface of the VERHO system. Figure 22 shows the monitoring interface. It presents the user with a list of the available interfaces. The active one is highlighted. The interface updates in real-time. For example, if a handover is performed due to LAC selecting a different interface as the active one, the user interface is updated correspondingly.

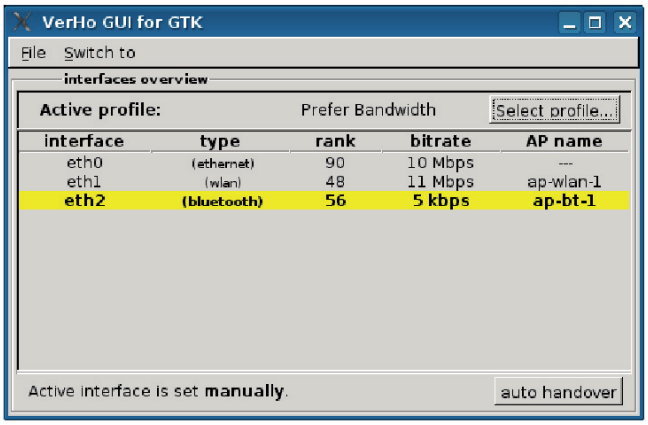

FIGURE 22 The Comui monitoring interface

Comui also allows the user to control the behavior of the system. It allows for the following actions:

- Manage profiles: select active profile, add new profile, delete existing profile.
- Enable and disable interfaces manually.

- Specify preference (priority) values manually.
- Select the currently active interface manually.
- Select the currently used Access Point for a given interface manually.

3.5.2 Multimedia Streamer

The Multimedia Streamer (MS) is a client server application. Image, audio and video streaming are supported. The client runs on the MN and it requests the media stream from the server in different qualities based on the characteristics of the active interface. When the active interface is about to change, the client evaluates the properties of the new interface and requests a stream with a better or worse quality from the server.

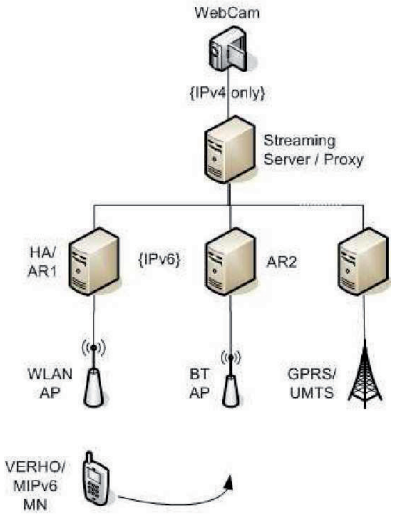

FIGURE 23 Multimedia Streamer Network Topology

In our prototype implementation (shown in Figure 23), the MS supports the following streams:

- Camera Stream: The server is connected to a WEB Camera. The camera can provide still JPEG images in different sizes and qualities. The server plays the role of an IPv6-IPv4 proxy. It relays the IPv4 image datagrams to the client over IPv6.
- Audio Stream: The server stores the same audio files in different qualities and serves to the client the one that is requested. When the server receives a quality change request from the client (the MN), it starts streaming the new file from the position (seconds) where the streaming stopped in the previous file.
- Video Stream: It is handled similarly to the audio stream.

3.6 Conclusions and future trends

Current trends in IP networking introduced node mobility and multihoming. Also, mobile network nodes are being equipped with multiple network interfaces. These introduce new challenges for adjusting the legacy stationary end-nodes, the IP network infrastructure and the underlying protocols to address the new demands.

The currently most mature mobility management protocol is Mobile IPv6. We analyzed the performance of Mobile IPv6 handovers. This led to the conclusion that utilizing Link Layer information can significantly reduce communication delays that occur during handovers. More importantly, from an interface management point of view, Link Layer information can be utilized for making more intelligent decisions about the choice of the target network.

The VERHO system is a policy controlled network interface management solution for Mobile IPv6. It gathers Link Layer information from different access technologies and disseminates this information to interested applications in the host. Among others, this information is used to influence handover decisions. These decisions include the selection of the target network (network interface) and the starting of the handover process. Furthermore, the gathered and unified link layer information and handover events can be utilized by multimedia applications to adapt to changes in the network environment, as demonstrated with a couple of prototype applications.

The VERHO system, as it was developed, uses only a single network interface for communication in a multi-interfaced node. Multiple interfaces are used only during soft-handovers. With the introduction of extensions to Mobile IPv6, such as [66], it becomes possible to use multiple interfaces simultaneously for regular data communication. Currently, in VERHO, a single network interface is assigned to the selected user profile. In order to support multiple interfaces, different profiles need to be used for different flows. For example, web traffic and VoIP traffic may use different profiles, resulting in different network interfaces being selected for their packet flows.

The VERHO system attempts to take advantage of the information available at the Link Layer. Every Access Technology is unique but it is possible to find a subset of information that can be extracted from each of them. Furthermore, the link indications, hints and triggers used by VERHO were defined based on experimental measurements and on experiences by using the system. This approach may not lead to optimal or desired behavior, even though it provided adequate results during tests.

The topic of utilizing Link Layer information in Upper Layers for optimization has been an active area of research. In this section, we look at some of the related results and try to draw some conclusions.

The authors in [4] give an overview of experiences gathered throughout the years by various research projects and gives some advice with regards to implementation and standardization. Link indications to be used by ULPs need to be

made independent of the underlying Link Layer, otherwise the ULP would depend on a specific access technology. For this purpose, it identifies the following indications:

- Link Up: indicates that the link is capable of sending and receiving IP datagrams. The corresponding interface may not have a valid IP configuration (e.g., IP address).
- Link Down: indicates that the link is not capable of transmitting IP datagrams.
- Bitrate, Frame Error Rate and Delay.

Even if ULPs are made aware of link indications, implementations should avoid to introduce link layer dependencies to ULPs *only* for optimization. When they do, the dependencies must be *soft*, the system must work without link indications, possibly with degraded performance.

These indications are propagated from the Link Layer up to other layers. A layered indication model is suggested. In this model, as shown in Figure 24, indications are used by non-direct-neighboring layers only when necessary.

Link indications, such as Link Up/Down, may be generated by the Link Layer in rapid successions due to frequent changes in wired or wireless states. For this reason, some damping or hysteresis is suggested.

Different layers utilize the link indications for different purposes. The Network Layer may set routing metrics based on bitrate and Frame Error Rate, this way affecting the path of outgoing IP datagrams.

At the Transport Layer, indications can be used for more accurate parameter estimation and connection management. A Link Up indication is likely not to be acted upon by the Transport Layer, since it depends on a valid IP configuration which may not be available yet. Rather, the IP layer may act upon the Link Up indication by configuring the IP parameters of the link (e.g., IP address, default route, etc.) and send an IP Address/Config Changes indication to the Transport Layer (and other ULPs). Also, upon a Link Down indication, a transport connection should not be torn down, rather it should be delayed until an IP Address Changes indication or if no response arrives for retransmissions. Reacting on Network Layer indications, such as ICMP Destination Unreachable, should also be done carefully, since such a state may be corrected later by the network.

At the Application Layer, instead of consuming Link Up indications directly, IP Address Changes indications from the Network Layer should be used. Similarly, instead of Link Down indications, connection tear-down indications from the Transport Layer may be more useful, since a link may go down and come up without braking the transport layer connections. In general, applications should not consume link indications directly.

Another important realization is that raw signal strength is not a good indicator of frame-loss or throughput due to, e.g., multipath interference. Therefore, signal strength alone should not be used as direct indication of Link Up/Down events.

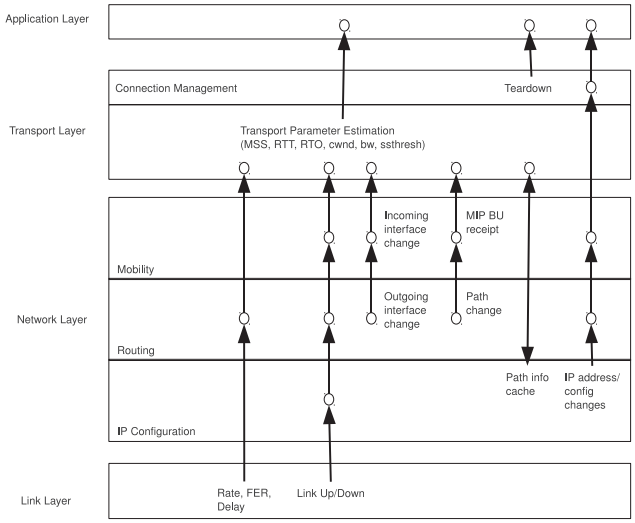

FIGURE 24 Layered Indication Model

The issue of using Link Layer indications for Detecting Network Attachment (DNA) is considered in [33]. When the mobile node moves from one Access Point to another (regardless of Access Technology), the IP subnet may be different at the new link. This necessitates the reconfiguration of IP addresses, routes, DNS server addresses, etc. The DNA procedure is used to detect if the subnet behind a given link has changed. The detection can be done by using the IPv6 Neighbour Discovery (NDisc) protocol [44]. NDisc allows a network node to discover routers on the link (Router Solicitation), routers to send periodical information about the link (Router Advertisement). The information routers provide contain sufficient knowledge so that network nodes can configure their own IP addresses and routes. This kind of IP setup is also known as Stateless Address Autoconfiguration [63]. The authors of [33] define a single major link indication for DNA, Link Up. This indication is defined as the event after which the link is capable of carrying IP datagrams. It is generated when the network node associates/connects with an Access Point. Naturally, this event may lead to a change in the subnetwork which must be detected so that the IP configuration of the corresponding network interface can be updated accordingly. Different Access Technologies handle the change in Access Points differently, as expected. Since the consumer of a Link Up indication, in case of DNA, is the Network Layer, it should be made Access Technology (or Link Layer implementation) independent. In [33], certain Link Layer events are identified for 802.11 WLAN, 802.3 Ethernet, GPRS/3GPP and CDMA2000/3GPP2 Access Technologies that may be used to generate Link Up indications. Furthermore, it is suggested that some auxiliary information, such as network prefix, router address, etc., may be conveyed in the Link Up indication to the Network Layer, so that an IP configuration can be established before/without [63] or DHCPv6 [17].

The authors of [62] define Link Layer primitives that can be used in han-

dover decisions in the Network Layer. It identifies nine different primitives. There are three primitive classes, queries, indications (or events) and controllers. The information provided by these primitives is abstracted away from the underlying Access Technology (they are Access Technology independent). Queries are requests sent from the Network Layer to the Link Layer to get some information. Events are asynchronous notifications sent from the Link Layer to the Network Layer (the Network Layer has to subscribe for these first). Controllers are primitives that allow the Network Layer to control the behavior of the Link Layer.

- L2-LinkStatus: Query: Used to request information about the link of a given network interface. The returned information contains the Access Point identifier and the link *condition*.
- L2-PoAList: Query: Used to request the list of available Access Points (Point of Attachments, PoAs) on a given network interface.
- L2-PoAFound: Event: Sent when a new PoA appears on the specified network interface link.
- L2-PoALost: Event: Sent when a new PoA disappears on the specified network interface link.
- L2-LinkUp: Event: Sent when a given link becomes capable of carrying IP datagrams.
- L2-LinkDown: Event: Sent when a given link becomes incapable of carrying IP datagrams.
- L2-LinkStatusChanged: Event: Sent when the link condition (as specified in the subscription) crosses the given threshold.
- L2-LinkConnect: Controller: Instructs the Link Layer to attempt to associate/connect to the given PoA.
- L2-LinkDisconnect: Controller: Instructs the Link Layer to attempt to deassociate/disconnect from to the given PoA.

The *condition* parameter in link status related primitives contains the current *bandwidth* of the link and the *link quality level*. The quality level is deduced from Access Technology specific parameters and is represented to the Network Layer as a technology independent value. It can have five levels (excellent, good, fair, bad, none). However, it is noted that making decisions upon the link quality level only may not be optimal and may be error prone. Nevertheless, it is calculated from the Received Signal Strength Indicator (RSSI), transmission/ reception rate indication, transmit/receive latency, bit-error rate, frame-error rate, and noise level. This is in contrast with the VERHO system that takes into account only one link parameter (e.g., Signal-to-Noise Ration for WLAN) to calculate a similar abstraction (the *Signal Strength* value) which may be even less reliable for decision making.

4 MULTIHOMING AND FLOW MANAGEMENT

In IP networks, a node is considered multihomed when it is reachable via multiple different paths by other nodes in the network. Also, not only single nodes, but entire networks can be multihomed, when there are multiple paths leading to the network from outside nodes.

Multihoming has become essential for providing robustness in the face of network failures to demanding services. It can provide fault tolerance and load balancing, but it also helps performing make-before-break handovers (or soft handovers) during node mobility. The following list shows some of the requirements presented in [3] for multihoming:

- Motivations: What do we want to achieve?
  - Fault tolerance, redundancy: If a path goes down the network nodes in the site should still be reachable on alternative paths.
  - Transport layer survivability: Established transport connections should not break when the underlying path changes.
  - Performance: Reduction of jitter, delay, loss.
  - Load sharing: Utilizing the multiple paths at the same time for both outbound and inbound traffic.
- Constraints: Quality of the solution.
  - Simple to deploy.
  - Minimal impact on the DNS.
  - Immune to ISP ingress filtering.
  - Scalable: does not put pressure on global BGP routing tables.
  - Minimal changes to routers and end hosts.

Multihoming has been around for years, even though the Internet was not designed with multihomed nodes in mind. It was added and engineered in along

the way as the need arose. In the current IPv4 Internet, engineers decided to introduce multihoming into the core network without touching the end nodes. This approach turned out to be inadequate in the long run. With the advent of IPv6, this problem is about to be addressed, end-nodes not being spared in the process.

In this chapter, we use the terms *end-node* and *host* interchangeably.

4.1 Trends and standardization efforts

This section, briefly summarizes articles [PVII] and [PIX] that survey the trends and standardization efforts with regards to host-centric multihoming.

In IPv4 networks, multihoming is provided to entire networks, called *sites*. Site-multihoming is popular among ISPs who want to offer robust network connectivity for their customers. The end-nodes inside the site do not need to be configured in any special way. That is, customers have no knowledge whether their site is multihomed or not, they just unknowingly enjoy its benefits. On the other hand, all currently used site multihoming solutions have negative effects on the global routing system.

Addressing in the Internet is hierarchical. An IPv4 network address is a 32 bit number. The entire pool of addresses is split up to large chunks by the IANA (Internet Assigned Numbers Authority). The IANA gives away these chunks to RIRs (Regional Internet Registry). The RIRs split up their given chunk to smaller chunks which is provided to LIRs (Local Internet Registry), ISPs or end-users. Figure 25 shows how this hierarchical split-up of the 32 bit address space happens in principle. In IPv6, the idea is the same but the initial address pool is 128 bits wide.

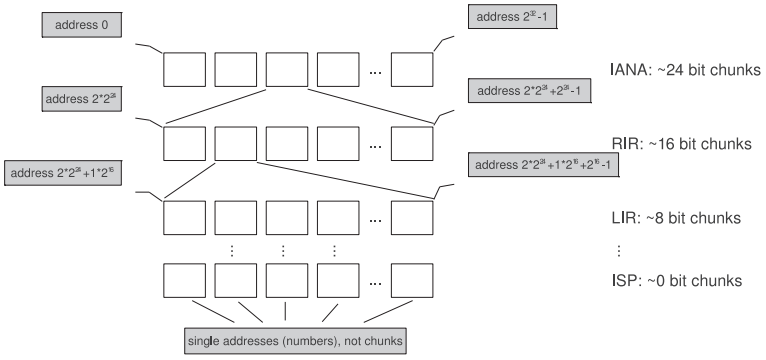

FIGURE 25 Principle of hierarchical addressing.

When an ISP receives an address range (chunk) from above the hierarchy it *aggregates* all the addresses included in that range. This means that by default, the core routing system of the Internet is set up so that every address belonging to that range will be routed to this ISP.

When a network uses an address range that it received from its ISP the above way, this range is called Provider Aggregatable (PA) address range. On the other hand, when a network uses an address range that does not hierarchically belong to any of its ISPs then this range is called Provider Independent (PI) address range. PI ranges have the advantages that the user (i.e., network) does not need to renumber when its ISPs renumber.

Figure 26 shows a simplified overview of the Internet. The Internet is a collection of Autonomous Systems (AS). An AS is a set of networks administered by a single technical administration (an administrative domain) and has its own routing policies. Border Gateway Protocol (BGP) is used as the Exterior Gateway Protocol (EGP) for the Internet, by AS routers that form the core. Usually, routers and end-nodes have a relatively small routing table containing routes to directly connected networks and a default route for every other traffic. On the other hand, there exists a set of routers in the core of the Internet that do not have default routes, they know exactly where to route any traffic. This set of routers is called the Default-Free-Zone (DFZ). The scalability of BGP depends on route aggregation [39], the more aggressive route aggregation leads to less routing table entries in the DFZ.

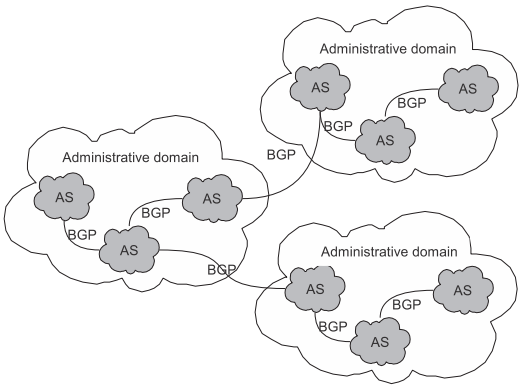

FIGURE 26 The Internet as a set of Autonomous Systems

Site-multihoming is achieved by introducing global routing table entries in routers in the DFZ of the Internet. There are two major ways to do this, both may introduce extra routing table entries in the DFZ.

Figure 27 shows how PA ranges can be used for site multihoming. Site C connects to two ISPs. From each ISP it receives a PA range. In order to be reachable via both ISPs, C advertises its reachability via A to B. B then passes this information to other routers. This may trickle up to the DFZ resulting in a global routing table entry.

Figure 28 shows how PI ranges can be used for site multihoming. The "pollution" of the DFZ is done the same way in this case as with PA ranges, there is no difference in this sense.

With more and more demands for multihoming, the performance of the global routing system is negatively affected. To solve this, a new approach to

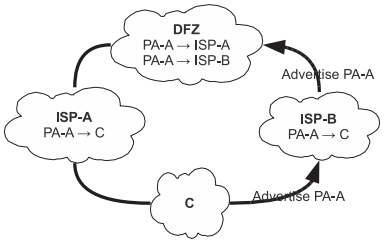

FIGURE 27 Multihoming with Provider Aggregatable address range

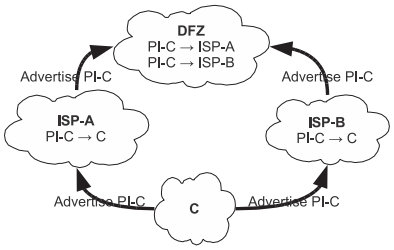

FIGURE 28 Multihoming with Provider Independent address range

multihoming is developed with IPv6. Instead of burdening the core network with additional logic and knowledge about multiple paths to the same destination, the logic is pushed to the end-nodes, where it really should belongs. This is done by allowing end-nodes to have multiple IP addresses (whether the multiple addresses are on the same network interface or different is not relevant for our discussion). Figure 29 shows this approach, called host-centric multihoming.

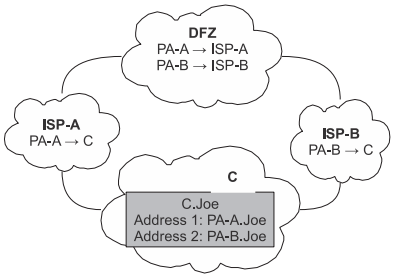

FIGURE 29 Host-centric multihoming

In the host-centric case, the end-nodes inside the site (e.g., C.Joe in site C in Figure 29) get as many IP addresses as many ISPs the site is connected to. The end nodes are responsible for managing the available multiple paths from source to destination. This includes:

- Selecting the proper source IP address for outbound datagrams.
- Instructing peers about the desired destination IP address for inbound datagrams.

These tasks are considered as Traffic Engineering (TE). In general terms, TE is the process of affecting the path that IP datagrams take from source to destination. TE can be done at different parts of the network:

- Network-centric TE: TE is done inside the network. In the core network or subnetworks the path of datagrams may be affected by tuning the routing protocols [53].
- Host-centric TE: TE is done at end-nodes. End-nodes can affect the source and destination IP addresses of datagrams. This changes the entry and exit points of network packets.

These two TE approaches complement each other. This way both end-nodes and ISPs have a say in the path of datagrams in the network.

Multihoming and host-centric TE brings up the possibility for *load sharing*. Load sharing happens when different datagrams are sent on different paths to *share the load*. Two cases can be distinguished:

- load balancing: per-session (e.g., connection) load sharing: different sessions are sent through different paths.
- load spreading: per-packet load sharing: packets belonging to the same transport session are sent through different paths. Load spreading may cause problems in the congestion control algorithms due to packet reordering caused by different path RTTs.

In [PVII] and [PIX], various host-centric multihoming solutions are reviewed. The issue can be handled at different layers of the network stack. When it is handled at lower layers (e.g., in the Network Layer) the solution can be transparent to ULPs. This transparency is useful from application development point of view since the ULP developer does not need to take care of multihoming issues. On the other hand, ULPs have more knowledge about the type and characteristic of the network traffic they handle. This latter means, as we go higher in the network stack, we can make better decisions about what a particular datagram needs with regards to QoS. This duality, transparency and knowledge, gives the birth to multihoming solutions working at different layers of the sack. Out of these, we reviewed the most active Transport and Network Layer solutions to date.

4.1.1 Transport Layer

Stream Control Transmission Protocol (SCTP, [57]) is a connection and message oriented, reliable protocol that has built-in multihoming capabilities. A connection is called *association*. One association consists of multiple streams. Every association may have multiple source and destination IP addresses (address-sets). Out of these addresses, only a single (source address, destination address) pair is

used. When a network failure happens, a different address pair is selected. By default, in the same association, every stream is sent using the same address pair.

The active address pair determines the entry and exit points of the network path. This opens up the opportunity for host-centric TE. [57] and [59] define mandatory functions for ULP-SCTP communication. With these primitives the ULP can specify the source address-set, force the usage of a specific destination address when sending a packet, change the primary path, retrieve the current primary path and other association information. An SCTP extension defined in [58] allows end hosts to dynamically add and delete IP addresses from an association during its lifetime. Without this extension, SCTP cannot react to changes in the address sets. This extension also allows an end host to tell its peer what IP address it prefers to receive data over, i.e., changing the destination address in the primary path at the peer node.

Multipath TCP (MPTCP, [22]) is an extension of the Transmission Control Protocol (TCP). It is a relatively new protocol actively worked on by the MPTCP Working Group at the IETF. MPTCP, just like TCP, is a reliable, in-order, stream- and connection-oriented transport protocol. It supports load spreading by its core specification. It provides better throughput by utilizing the bandwidth of multiple available paths between communicating peers. A multihomed node can utilize MPTCP by setting up a connection with its peer that consists of multiple subflows. Each one of these subflows represents a different path and is bound to a different IP address pair (one for both peers).

The usage of the available subflows is supposed to be governed by policies or profiles. However, these profiles may not satisfy all scenarios. Therefore, more control should be given to applications to control the usage of subflows. A socket Application Programming Interface (API) extensions for MPTCP is proposed in [52]. With the defined extensions, applications can, among other things, enable/disable MPTCP, define the set of addresses or network interfaces to be used for subflows, get information (e.g., IP addresses) of the currently used subflows, set the profile and trigger subflow creation or removal.

Datagram Congestion Control Protocol (DCCP, [29]) is an alternative to UDP. This proliferation of UDP traffic in the Internet is likely to cause trouble in the long run with regards to congestion, since UDP does not provide congestion control. DCCP is a connection oriented, message based, un-reliable transport layer protocol with congestion control.

An extension is defined in [30] for DCCP for multihoming. Multihoming is provided by combining multiple separate DCCP connections into the same main connection. On the other hand, an end host can use a control message to indicate the peer which component connection should be used for data communication, although this is taken only as a hint by the peer.

Figure 30(a) shows the mapping between SCTP sockets and local IP addresses. The same is shown for MPTCP in Figure 30(b) and for DCCP in Figure 30(c). The similarity is striking, which shows that it may be possible to make both of them use a common host-centric TE policy system for source and destination address selection.

4.1.2 Network Layer

The Site Multihoming by IPv6 Intermediation protocol (SHIM6, [48]) is an IPv6 multihoming solution. It is based on PA address ranges, aggressive route aggregation. It presents a stable locator, called Upper Layer Identifier (ULID), to ULPs and dynamically maps these ULIDs to actual locators that are used on the wire. Figure 30(d) shows this mapping and the place of the SHIM6 layer. ULPs bind to ULIDs, these ULIDs are mapped to locators used for the routing of packets. On the receiver side, a reverse mapping is performed causing ULPs to believe that they are using the ULIDs as locators.

There are several proposals for SHIM6 that support host-centric TE. In [7], the authors define how the (source address, destination address) locator pair may be selected based on policies using preference values, weights and network prefixes. In [31], a socket API extensions are defined with which applications (or ULPs) can programmatically control path selection and other aspects of SHIM6 (and possibly other similar shim solutions, e.g., Host Identity Protocol).

The Host Identity Protocol (HIP, [43]) allows two end hosts to securely establish a secure IP layer session. During session setup, a Sigma-compliant Diffie-Hellman key exchange is used using public key cryptography. After the session setup, the control messages are cryptographically secured. HIP uses a new cryptographic namespace to provide stable identifiers for ULPs, called Host Identity Tags. The mapping from HITs to locators is performed by the HIP layer.

HIP is not a multihoming protocol in essence, thus it does not have explicit support for it. So far, there has been no literature in the area of locator pair selection for HIP. Also, by default only a single locator pair is assigned to a HIP association, meaning that only a single path is used between the two nodes even if multiple is available. The authors of [47] define mobility and multihoming extensions. It covers basic end-to-end host solutions and more complex scenarios are left for further study. Host implementing this specification can register multiple locators to a single association. In [50], extensions are defined for [47] and enables a policy based selection of SAs. Figure 30(e) shows the mapping of HIP identificers to locators at end nodes.

Mobile IPv6 (MIPv6) has also gained multihoming support during its development. In the base specification, by default, only one CoA can be registered with a HoA. When the MN enters a multihomed site, it assigns more than one CoAs to its network interfaces. In [66], extensions are defined for MIPv6 that allow the MN to register multiple CoAs at the HA and at CNs. On the other hand, it does not define how and when the multiple registered CoAs are to be used for data traffic. For directing outbound traffic to use a specific CoA when multiple are available, one can use tools provided by the operating system and the local MIPv6 implementation. In [56], extensions are specified to MIPv6 to inform peers (HAs and CNs) about the usage of multiple CoAs. [65] defines Flow Description sub option types for [56]. It describes separate options for IPv4 ad IPv6 flows. In [34], a textual flow description language is defined that is independent from the underlying multihoming or mobility protocol, and is meant to be applicable also

for MIPv6. Figure 30(f) shows the mapping of identifiers to locators at end nodes for MIPv6 with multihoming extensions.

The Locator/ID Separation Protocol (LISP, [21]) is a direct approach to separate the locator and identifier aspects of IP addresses. LISP, in its original form, supports both IPv4 and IPv6 networks, can be deployed incrementally, and does not require changes to end-nodes. Only routers close to the edge may need to be modified.

LISP introduces a new namespace for identifiers. These identifiers (Endpoint Identifiers or EIDs) are regular IP addresses. Locators (Routing Locators or RLOCs) are also regular IP addresses. EIDs are assigned to end-nodes within a site; they are local to the site and are not globally routable. The EID of a node is supposed to change rarely once it has been assigned providing a stable identifier for ULP sessions. The EID-RLOC mapping is done by site-edge routers with the help of an overlay network called LISP+ALT ([20]). This overlay network consists of LISP+ALT routers and constitutes the control-plane of LISP. EID-RLOC mapping requests and replies are sent over this network between the LISP+ALT routers. Multihoming is supported by allowing the assignment of multiple RLOCs to the same EID prefix (an EID prefix represents the EIDs belonging to the same site).

The LISP Mobility Architecture (LISP-MN, [19]) describes a way to implement LISP ITR/ETR functionality in end-nodes. This way LISP can be used as a host-centric multihoming solution. LISP supports the assignment of weights and priority values to RLOCs in an EID-RLOCs mapping. This can be used by multihomed LISP end-nodes to influence RLOC usage of their inbound traffic. RLOCs with smaller priorities are preferred over larger ones. Data traffic is spread over RLOCs with the same priority values according to the weights. The weight tells the sender the percentage of the outgoing datagrams that should be sent over the given RLOC. Figure 30(g) shows the mapping of LISP identifiers to locators at end nodes.

4.1.3 Policy based Traffic Engineering

4.1.3.1 Path Selection

Policy support for host-side traffic engineering is usually constrained to outbound traffic. The sending node decides about the source and destination addresses based on local policies.

A standardized way of outbound path selection is described in RFC3484 [16]. It describes two algorithms. One for destination address ordering and one for source address selection. It takes into account user policies in the form of a policy table. It is a longest-matching-prefix lookup table and consists of entries in the form of "network prefix, precedence, label". When an address (either source or destination) matches an entry, it is assigned the precedence value and the label. The precedence value is used for sorting destination addresses. The label value is used for selecting the source address with regards to a destination address.

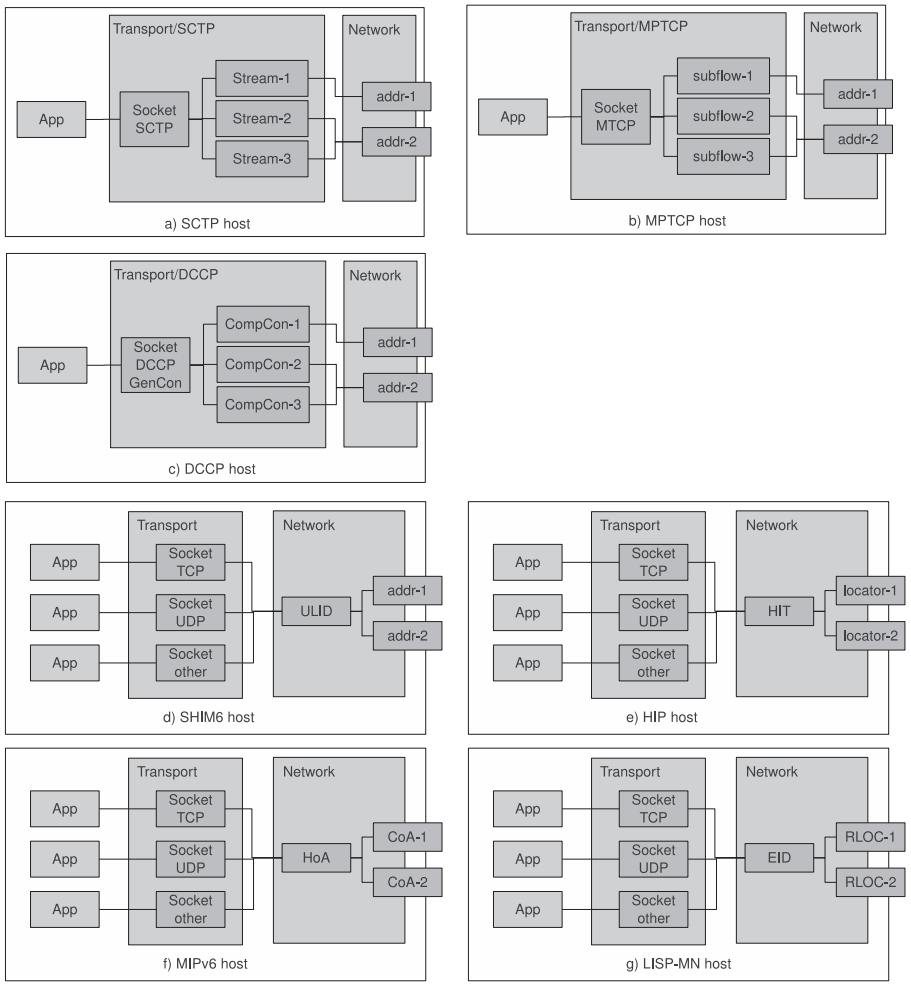

FIGURE 30 Binding between sockets and local IP addresses

In [6], extensions are defined for [16]. It proposes that besides a destination address list, an ordered source addresses list should be available to applications. For this purpose, it describes an ordering algorithm for source addresses.

Source based routing complements RFC3484. It makes sure that the outgoing packet will actually leave the host on the network interface that has the selected source address assigned to it. This makes sure that ISPs do not drop traffic due to ingress filtering. The availability of source based routing varies between operating systems.

A Name, Address and ROute System (NAROS) is defined in [13]. It deploys dedicated servers in the site whose task is to aid hosts in the selection of source and destination IP addresses for outgoing traffic. It is considered as an addition to RFC3484. [13] leaves open the question of what the NAROS server bases its decisions on, which makes this solution very flexible. Neither does not specify any communication between NAROS servers in different networks, which means that, by default, hosts cannot influence their peers about how to select paths for inbound traffic.

Common Open Policy Service (COPS, [18]) defines a policy service which was designed for the administration, configuration and enforcement of policies. It consists of Policy Repositories (PR), Policy Decision Points (PDP) and Policy Enforcement Points (PEP). When a PEP needs to enforce a policy it issues a request to a PDP. The PDP evaluates the request and replies with a to-be-enforced policy. The NAROS system can be viewed as a specialized COPS.

4.1.3.2 Policy Description

It is common to distinguish between policies and rules. Policies define *when* to apply certain rules, rules define *what* to do. As we have seen, HIP [50] and MIPv6 [56] has certain extensions to exchange path selection rules between peers to influence inbound traffic. These rules are actually a mixture of policies and rules, they define a set of packets and assign a path to them, e.g., "send TCP traffic with destination port 22 using the source address 2001:10::1". Those solutions are specific to the protocol.

A multihoming protocol independent policy framework for flow distribution is proposed in [40]. Users can specify policies that are translated down to filter rules. Policies are stored in a Policy Data Set. The policies and filter rules are meant to be re-usable by different multihoming solutions. The filter rules are further translated down to operating system or multihoming protocol dependent representations. Filter rules can also be exchanged between peers to influence inbound traffic, although the protocol to perform this is not specified but a hint is made at COPS. Neither the policy language, nor the rule language is described in [40].

A textual language is defined in [34] to represent flow rules for per-flow path selection. Rules are stored in a rule set and are evaluated sequentially. If a packet matches a rule it is forwarded on the specified path. Each path is represented by a path identifier (PID), but the mapping of the PID to actual IP ad-

dresses is not described. For example, in the case of [66], the BID can serve the purpose of the PID. The language has limited support for conditional rules based on the availability or unavailability of PIDs.

The authors in [60] review various policy handling methods and policy description languages. It looks at inter-domain policy routing practices, network policy languages and traffic flow languages. The authors propose their own Path-Based Policy Language. They put the emphasis on language abstraction, unambiguous policy representation and policy conflict resolution. Policies can be created by different entities, thus every policy has a creator User ID. Policies are assigned to paths. A path is described as a list of nodes, where nodes are represented by numbers. A policy is applied to a packet if the packet goes through the specified path and fulfills the specified conditions. Conditions can be built from certain variables (e.g. traffic class, used bandwidth, IP address, time, etc.). Actions can be, e.g., drop packet, assign priority to the policy (for conflict resolution with other policies), etc. How these abstract policy descriptions are translated down to device or domain specific representations is not presented.

Whether or not a common protocol independent policy description method will emerge, is yet to be seen. Such a method has to be flexible enough that it could be extended to support future protocols. This seems to be one of the biggest challenges that awaits anyone trying to tackle this issue. On the other hand, considering the difficulty to grasp the aspects and requirements of different multihoming protocols and users, an incremental approach may be possible, where starting from protocol dependent methods, a protocol independent method might emerge as the need arises.

4.1.4 Other related work

There are various research projects that aimed to implement a multihoming management middleware for mobile nodes.

The authors in [70], present a policy based handover mechanism for multihomed mobile nodes. It allows the use of multiple interfaces simultaneously (*simultaneous multiaccess*). User defined policies and actions associate network flows (e.g., an TCP connection on destination port 80) with available network interfaces. Different multihoming and mobility capable protocols are considered. Their prototype is implemented with Mobile IPv6 (and its extensions) due to its mature specification with regards to multiple interface handling.

A profile-based multiple-interface management system is defined also in [9]. It also uses Mobile IPv6 and its multihoming extensions. Profiles can be defined by multiple different entities (e.g., network operator, user, application). The profiles are merged and network interfaces are selected based on multiple input attributes (e.g., link layer, access network and mobile device attributes).

The authors in [64], demonstrate the capabilities of multihoming with Mobile IPv6 via a real implementation of [66] and [56].

4.2 Separation of Policy Exchange

This section gives a summary of article [PVIII] that introduces a method for policy exchange for the Host Identity Protocol.

As can be seen in Figure 30, there is a lot of similarity between different mobility and multihoming solutions with regards to the binding of locators to identifiers (or sockets). This indicates that it is possible to create a common policy based framework for locator selection. Sections 4.1 and 4.1.3 show that, currently, most of the multihoming solutions try to address this issue in their own custom way. This includes also the exchange of flow rules or policies for inbound address selection between the peers.

In [PVIII], we introduce a policy exchange method for HIP. This method defines a separate protocol over TCP for policy exchange. Furthermore it uses a flexible and extensible language for policy descriptions. These two aspects make it possible to extend this method to be applicable also to other Network Layer Multihoming Protocols (NLMP) than HIP. In this section, we describe the designed policy system in a multihoming protocol agnostic manner. For more details how it can be integrated with a given protocol (HIP), see [PVIII].

The system has the following assumptions about supported NLMPs:

- Locator/identifier separation: the NLMP must employ some sort of separation between locators and identifiers. For example, HIP uses HITs as identifiers and real IP addresses as locators. Mobile IPv6 uses the HoA as identifiers and CoAs as locators.

- Association based: before policies can be exchanged between the peers, the NLMP has to set up an association between them. This can be for example a HIP Base Exchange, or a Mobile IPv6 CoA registration.

- Locator Identifiers (LID): every registered locator has an associated identifier as a positive integer number. This identifier abstracts away the underlying locator. The LID can stay the same while the locator changes.

Figure 31 shows a simple view of the policy exchange system. The system behaves the same way on both nodes, i.e., it is symmetric. The Policy Module consists of three main parts. The Policy Database, the Policy Transport and the Policy Enforcement.

4.2.1 Policy Database

The Policy Database stores the policies. There are two classes of policies, Local Policies and Peer Policies. Local Policies determine the selection of source addresses for outbound datagrams. Peer Policies determine the destination address. The system is built on the assumption that every node should be concerned only with its own addresses. That is, node A cannot instruct node B about the usage

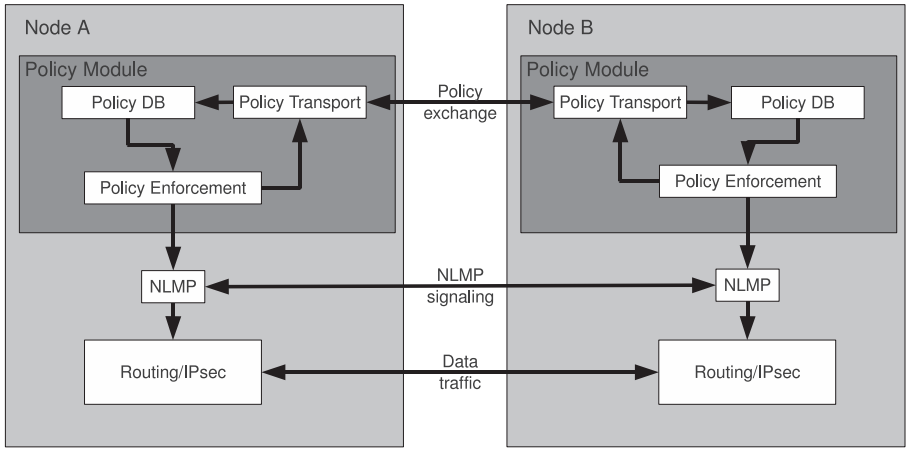

FIGURE 31 Architecture of the Policy Exchange System

of node B's addresses. Every node selects only its own addresses both for outbound (source address) and for inbound (destination address) datagrams. The only exception from this is when there are no Peer Policies present. In that case, the sending node has no knowledge about the address preferences of the receiver and default destination address selection rules apply [16].

The Local Policy Set contains Peer Policies. There is only one Local Policy Set. A Peer Policy Set contains Peer Policies. There are as many Peer Policy Sets as many peers the host has established an NLMP association with. The Local Policy Set is paired with all the Peer Policy Sets to produce the actual rules that select a (source address, destination address) tuple for outbound datagrams sent from the node to its peer.

Policies are represented in a textual form. The Lua scripting language was selected because it is extensible and expressive (being a general purpose scripting language), and its interpreter is small enough to be included even in embedded systems. We propose the usage of a stripped-down Lua environment, including only functions necessary to express flow rules. This results in a flexible domain specific language.

The following listing shows an example of a Policy Set:

```
0. NodeIdentifiers =
        { HIT = "00112233445566778899aabbccddeeff" }
1. WLAN = {1, 2, 3}
2. Ethernet = { 4, 5, 6 }
3.
4. if is_locator_available_any(Ethernet) then
5.      rule_add{ protocol = "TCP", to = Ethernet,
                  selection = "sequential" }
6. elseif is_locator_available_any(WLAN) then
7.      rule_add{ protocol = "TCP", to = WLAN,
```

```
                  selection = "sequential" }
8.      rule_add{ protocol = "TCP", dport = "80",
                  to = WLAN,
                  selection = "sequential" }
9. end
```

Every Policy Set carries the node identifiers of the owner. There can be multiple node identifiers belonging to different NLMPs. In line 0, there is only one identifier, a HIP HIT. Upon reception of this Policy Set, the receiver verifies that there is indeed an established NLMP association with the sender. For this purpose, the system communicates with the underlying NLMPs.

The tables in lines 1 and 2 define two network interfaces (locator tables). The WLAN interface is associated with LIDs 1, 2 and 3. The Ethernet interface with LIDs 4, 5 and 6. This means that the when the NLMP registers a WLAN address (locator) with the peer, it uses 1, 2 or 3 as the corresponding LID.

The is_locator_available function at line 4 asks the NLMP whether any LID of Ethernet is registered with the NLMP association. This assumes that before this function can be used, the peers have to have registered their multiple locators with each other over their chosen NLMP.

The rule_add function is responsible to insert the actual Operating System specific address selection rules into the networking subsystem. For example, it may be IPsec Security Policies or packet filtering/routing rules. The arguments of this function contain the flow description. The flow description describes the network datagrams to which this rule applies, and the desired LID (source address in case of Local Policy Set, destination address in case of Peer Policy Set).

The language supports the definition of properties for locator tables. This can be used to assign certain parameters to network interfaces. For example the following Policy Set fragment sets up a "cost" property for WLAN and Ethernet:

```
1. set_property(WLAN, "cost", 345)
2. set_property(Ethernet, "cost", 0)
```

Properties can be combined to conditionals in policy descriptions:

```
1. if get_property(Ethernet, "cost") <
      get_property(WLAN, "cost")
   then
2.     rule_add{to = Ethernet, selection = "sequential"}
3. elseif
4.     rule_add{to = WLAN, selection = "sequential"}
5. end
```

4.2.2 Policy Enforcement

The Policy Enforcement component is responsible for the translation of Policy Sets into actual Operating System (OS) specific routing and address selection

rules. It takes the Local Policy Set and pairs it with every Peer Policy Set. The result of every pairing are a set of (source address, destination address, flow description) tuples. These tuples are then translated to OS specific rules.

4.2.3 Policy Transport

Policies are exchanged between the peers over an established TCP connection. This connection is set up after a successful NLMP association. It is kept alive as long as the peers want. The establishment of this Policy Transport Connection can be initiated by either peer.

The entire policy description must be transferred to the peer in case of policy changes. Upon receiving the policy description, it is stored in the Policy Database, the Policy Enforcement module interprets it and deploys the resulting OS filter rules.

It is assumed that policy descriptions do not change often. Most likely, a well written policy description does not need to be changed at all after the initial transfer to the peer. On the other hand, locator properties are expected to change. Therefore, property setting instructions can be carried in a dedicated message type. This message can contain only set_property commands. Upon receiving a property setting message, the Policy Enforcement module re-runs the policy description with the new property values and modifies the OS filter rules accordingly.

4.3 Implicit Inbound Address Selection

In this section, we consider the address selection problem from the opposite side to the one in Section 4.2. Instead of exchanging full policy descriptions, we introduce a method that leverages the inherent flow information of datagrams.

In Section 4.1 and in [PVII], several Network Layer based host-centric multihoming protocols were reviewed. There are some important similarities between these solutions. They establish a context between the host and the peer. Both sides can register multiple locators with the same context. For outbound datagrams, the selection of the destination locator by the sender may be influenced by the receiver using explicit mechanisms. This is achieved by exchanging flow rules. A flow rule is a (flow description, locator) binding. It indicates that every datagram that matches the flow description should be sent to the given destination locator. Flow descriptions are described with the help of a flow description language. This language can either be textual or binary.

The description of flows usually consists of a set of (key, value) pairs. These (key, value) pairs describe the set of network packets that the particular flow description matches. The keys come from various parts of the IP datagram, such as transport layer port numbers, IP addresses, IPsec SPI numbers. For example, a rule like “tcp source-port 80” would match every IP datagram that carries a

TCP header with source port 80. These flow descriptions are transferred to peers and bound to registered locators. The descriptions and the bindings are carried usually in protocol specific control messages.

All of the reviewed solutions use some sort of tunneling (either IPsec, IP-in-IP, or some custom one based on IP header rewrite) to achieve the mapping between identifiers and locators. For simplicity, we assume the following. For outbound traffic, the ULPs pass to the shim layer IP datagrams of the following format:

```
IP(src=idSrc, dst=idDst, ULP(...))
```

We refer to this as the *ULP-datagram*. This is encapsulated in a tunnel with the associated locators and leaves the shim layer:

```
IP(src=locSrc, dst=locDst,
   IP(src=idSrc, dst=idDst, ULP(...)))
```

On the receiver side, the shim layer decapsulates the original IP datagram (with the idSrc and idDst identifiers) and passes it up to the ULP.

4.3.1 The INAS method

The exchange of flow rules to influence inbound path selection requires further extensions to multihoming protocols. Out of the ones we reviewed in Section 4.1, all need extra modifications or extensions on top of their base specifications.

With INAS there is no need to extend the multihoming protocols with extra control messages to carry flow descriptions and (description, locator) bindings. It generates and extracts the descriptions and bindings from the IP datagrams themselves.

The operation of a host-centric multihoming protocol with INAS is as follows. The multihomed host registers its multiple IP addresses with its peers using the mechanisms provided by the used multihoming protocol. This makes sure that the necessary states are established at both the host and the peers, so that the host is able to use the registered addresses to send or receive datagrams to/from the peers. Also, it is usually not allowed for the host to use non-registered addresses with the peers according to the requirements of the used multihoming protocol.

After the addresses have been registered at the peer, data communication can begin. We assume that every datagram can be associated to a *session*. A session may be thought of as the set of datagrams that are exchanged between a (peer-socket, host-socket) tuple. For example, every datagram that belongs to the same TCP connection makes up a session.

Figure 32 shows the mechanism of INAS. The host can use its local policies and rules to select the proper source locators for outbound datagrams. However, for inbound traffic, the destination locators used by the peer may not be appropriate. For example, in Figure 32, the peer sends datagrams to address Host.A

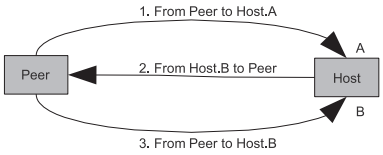

FIGURE 32 Re-directing inbound traffic with INAS.

instead of Host.B (arrow 1). In order to influence the destination locators used in inbound datagrams, the host sends outbound datagrams in the same session from the desired locator (arrow 2). The peer, upon receiving a datagram from the host, associates the source locator of the datagram with the session. From then on, every datagram that belongs to the session and originates from the peer to the host, will be sent using the associated locator (arrow 3). Therefore, it is assumed that the host always sends outbound datagrams using the source locator that it wants to associate with the corresponding session for inbound datagrams.

4.3.1.1 Flow description generation

The key point of INAS is the extraction of flow descriptions from sessions information. It is assumed that the necessary session information is included in the IP datagrams. By session information, we refer to the various fields and values in the protocol headers of the datagram.

For example, let us consider a TCP session. The shim layer at the peer receives a TCP datagram from the host in the following form:

```
IP(src=locSrc, dst=locDst,
   IP(src=idSrc, dst=idDst,
      TCP(src=portSrc, dst=portDst ...)))
```

In order to generate a flow description that matches the datagrams of this session that flow towards the host, the peer extracts the protocol, the source and destination port, the source and destination IP address from the datagram. Then it may create the following flow rule:

```
(src-addr=idDst, dst-addr=idSrc,
 proto=TCP,
 src-port=portDst, dst-port=portSrc)
--> locSrc
```

4.3.1.2 Security Considerations

INAS allows the host to redirect the traffic of the peer and could cause redirection related attacks. This is alleviated by relying on peers to deny redirection to unregistered addresses. Therefore, we rely on the security of the address registration mechanism of the used multihoming protocol.

4.3.1.3 Limitations

Let us consider Figure 32 and assume that the peer sends the first datagram of the session. This datagram (arrow 1) will be sent with the "incorrect" destination address (Host.A). It takes half of an RTT for the host to correct it (arrow 2). If this behavior is undesirable, it can be mitigated by combining INAS with explicit modes of flow rule exchange ([50], [56]).

In case of uni-directional peer-to-host sessions, the host has to send redirection-datagrams to the peer for traffic redirection. By *redirection-datagrams*, we mean datagrams that carry no payload (e.g., an empty UDP packet) and do not disturb the uni-directional session.

INAS is applicable only to sessionable ULP-datagrams. Usually, datagrams that carry transport protocols, are sessionable (UDP, TCP, SCTP, DCCP). In general, network layer protocols that do not carry transport protocols are usually not sessionable (some ICMP messages are). Therefore, INAS is applicable mostly to application data traffic.

Transport protocols encapsulated in IPsec Encapsulating Security Payload (ESP) do not carry enough information to identify reverse traffic. Specifically, the SPI number of the reverse traffic is necessary to generate the proper flow description. One alternative would be to carry the reverse-SPI (inbound SPI) explicitly in the outbound datagram (for example, as an IPv6 extension header).

INAS requires that the peer maintains and updates the flow rules of active sessions. This means that the peer has to do some extra processing for every inbound datagram and update the corresponding flow rules accordingly. In a mobility and multihoming scenario, most likely only locators would change during the lifetime of a session. Therefore, it may be enough to monitor only the locators for changes which reduces the amount of processing. Other local implementation decisions would also certainly affect the performance of INAS, such as the data structure used to store and look-up flow rules (for example the operating system's IPsec Security Policy and Security Association databases, or its packet filtering framework).

4.3.2 Integration with existing protocols

4.3.2.1 Mobile IPv6

With the extensions of [66], a MIPv6 MN can register multiple CoAs with the same HoA at its HA and CNs.

Data traffic between the MN and CN can either take a non-Route Optimized (non-RO) or RO paths. When no RO is used, all traffic between the MN and CN goes through the HA. The HA acts as a relay between the MN and the CN. IP datagrams sent from the MN to the CN are first encapsulated in either an IPv6-in-IPv6 tunnel or an IPsec ESP tunnel. This tunnel reaches from the CoA of the MN to the IP address of the HA. The tunnel is created when the MN registers the CoA with the HA. This means that for unregistered addresses, no tunnel exists. Also,

[27] and [66] require the HA to drop datagrams sent from unregistered CoAs. IP datagrams sent from the CN to the MN are sent to the HoA. These datagrams are captured by the HA in the home network. The HA encapsulates them and sends them over one of the tunnels to the MN. In case the MN is multihomed, there can be multiple tunnels and INAS can be used to select the proper one for datagrams sent from the CN to the MN (i.e. inbound datagrams).

Using INAS, when the MN sends datagrams to the CN via the HA using the desired tunnel, the HA inspects the de-capsulated datagram and associates the source tunnel with the session of this datagram. From then on, every datagram belonging to the same session and sent from the CN to the MN, will be encapsulated and relayed to the MN using the associated tunnel.

When Route Optimization is used, the MN registers its multiple locators directly with the CN. As a result of this registration, both the MN and the CN create the necessary states (Binding Cache and Binding Update List entries) that allow direct communication (using Type 2 Routing Header, HoA Destination Option). Similarly to the non-route-optimized case, the MN is expected to send outbound datagrams to the CN using the CoA that it wants to associate with the corresponding session. The MIPv6 layer at the CN, after verifying that the CoA is registered, extracts the necessary session information from the received datagram and associates the CoA with the session. From then on, every datagram belonging to the same session and sent from the CN to the MN, will be sent to the MN using the associated CoA.

4.3.2.2 Host Identity Protocol

HIP also has extensions for multihoming described in [47]. With these extensions a HIP host can register multiple locators with its peers.

In a non-multihoming case, a HIP association between the host and the peer is represented by a pair of IPsec Security Association (SA). The host has one SA for outbound and one for inbound traffic, similarly to the peer. An SA is unidirectional, this makes it necessary to create a pair for bi-directional communication. Also, a HIP SA is associated with the peer and host locators. For outbound datagrams, the host uses its outbound SA, which instructs the IPsec module of the host to use the associated locators to route the datagram. Under the hood, IPsec encapsulation is used that we do not elaborate on here. For inbound traffic, the peer sends datagrams using the inbound SA of the host.

In a multihoming case, there may be multiple in/outbound SA pairs at the host and INAS helps to select the proper one for peer-to-host traffic (i.e. inbound traffic). Each of the SA pairs represents a registered locator. For outbound traffic, the host can use its local policies and rules to select the proper outbound SA. When the peer receives a datagram from the host, the HIP layer extracts the session information and associates the session with the source locator of the datagram. Since SAs are bound to locators, the peer can select the proper outbound SA for datagrams belonging to the session sent towards the host. For this, the HIP layer at the peer selects an SA that is bound to the same host locator as the

session.

4.3.2.3 SHIM6

SHIM6 defines a ULID (Upper-Layer Identifier) pair. This is a pair of locators, one for the host and one for the peer. The ULID-pair is associated to the SHIM6 context when it is established. A SHIM6 context is used only by those outbound IP datagrams that were sent by using the addresses in the associated ULID-pair. For example, two TCP connections established with different ULID-pairs will need two different SHIM6 contexts.

By default, the IP addresses in the ULID-pair are used for outbound data traffic by both the host and the peer. The extra locators that were exchanged during context establishment (or later via control messages) are used if the ULID-pair fails or when explicitly requested. Explicit selection of the preferred local (host) and remote (peer) locators can be achieved with [31].

With INAS, the host can indicate to the peer what host locator should be used for peer-to-host data traffic for a given SHIM6 context. For outbound traffic, the host can use its local policies and rules to select the proper local locator. When the peer receives a datagram from the host, the SHIM6 layer extracts the session information and associates the session with the source locator of the datagram. Then, for outbound traffic towards the host, the SHIM6 layer at the peer can use the locator associated with the session.

4.4 Conclusion

The simultaneous use of multiple interfaces leads to multihoming. We investigated the state and trends in IP multihoming and concluded that host-centric solutions are likely to dominate in the future. An important observation is that the address selection procedure in end-nodes can affect the entry and exit points of outbound IP datagrams. Thus, address selection becomes a tool also for network interface selection in a per-packet basis.

Several host-centric multihoming protocols exist for both the Network Layer and the Transport Layer. We focused our investigation to Network Layer solutions and pointed out the similarities that lead to the design of a policy exchange method. The policies are used to influence the destination address selection (inbound address selection from the host's point of view) at the peer. This method was developed with supporting multiple Network Layer multihoming protocols in mind.

An alternative inbound address selection method (INAS) was also introduced that utilizes the inherent flow description information in regular IP datagrams. We analyzed the most active Network Layer multihoming protocols and proposed how INAS can be integrated to them.

When we look at current standardization efforts we see that they mostly

follow the path that was taken also by the research work introduced in this dissertation with regards to the utilization of Link Layer information.

The trend in policy exchange and flow description in mobility and multihoming protocols is to introduce specialized solutions. Despite this, due to the similarity of these protocols, the separation of the common parts to a shared protocol or solution may be considered. For example, this separation is beneficial if multiple different multihoming protocols are used simultaneously in future communications.

5 SUMMARY

Current trends in IP networking introduced node mobility and multihoming. Also, mobile network nodes are being equipped with multiple network interfaces. These introduce new challenges for adjusting the legacy stationary end-nodes, the IP network infrastructure and the underlying protocols to address the new demands.

In this dissertation, we focused on the management of network interfaces and IP addresses in future and current mobile network nodes. We addressed these issues step by step.

The most mature mobility management protocol is currently Mobile IPv6. We analyzed the performance of Mobile IPv6 handovers and this led to the conclusion that utilizing Link Layer information can significantly reduce communication delays that occur during handovers. More importantly, from an interface management point of view, Link Layer information can be utilized for making more intelligent decisions about the choice of the target network.

The VERHO system is a policy controlled network interface management solution for Mobile IPv6. It gathers Link Layer information from different access technologies and disseminates this information to interested applications in the host. Among others, this information is used to influence handover decisions. These decisions include the selection of the target network (network interface) and the starting of the handover process. Furthermore, the gathered and unified link layer information and handover events can be utilized by multimedia applications to adapt to changes in the network environment, as demonstrated with a couple of prototype applications.

The VERHO system, as it was developed, uses only a single network interface for communication in a multi-interfaced node. Multiple interfaces are used only during soft-handovers. With the introduction of extensions to Mobile IPv6, such as [66], it becomes possible to use multiple interfaces simultaneously for regular data communication. Currently, in VERHO, a single network interface is assigned to the selected user profile. In order to support multiple interfaces, different profiles need to be used for different flows. For example, web traffic and VoIP traffic may use different profiles, resulting in different network interfaces

being selected for their packet flows.

The simultaneous use of multiple interfaces leads to multihoming. We investigated the state and trends in IP multihoming and concluded that host-centric solutions are likely to dominate in the future. An important observation is that the address selection procedure in end-nodes can affect the entry and exit points of outbound IP datagrams. Thus, address selection becomes a tool also for network interface selection in a per-packet basis.

Several host-centric multihoming protocols exist for both the Network Layer and the Transport Layer. We focused our investigation to Network Layer solutions and pointed out the similarities that lead to the design of a policy exchange method. The policies are used to influence the destination address selection (inbound address selection from the host's point of view) at the peer. This method was developed with supporting multiple Network Layer multihoming protocols in mind.

An alternative inbound address selection method (INAS) was also introduced that utilizes the inherent flow description information in regular IP datagrams. We analyzed the most active Network Layer multihoming protocols and proposed how INAS can be integrated to them.

When we look at current standardization efforts we see that they mostly follow the path that was taken also by the research work introduced in this dissertation with regards to the utilization of Link Layer information.

The trend in policy exchange and flow description in mobility and multihoming protocols is to introduce specialized solutions. Despite this, due to the similarity of these protocols, the separation of the common parts to a shared protocol or solution may be beneficial, should multiple different multihoming protocols be used simultaneously in future communications.

YHTEENVETO (FINNISH SUMMARY)

Mobiililaitteet ja mobiiliverkot vapauttavat ihmiset liikkumaan ja silti pysymään yhteydessä ystäviin ja työkavereihin. Olemme yhä riippuvaisempia mobiiliverkkojen toimivuudesta. Tämän vuoksi on tärkeää luoda luotettava infrastruktuuri, joka mahdollistaa jatkuvan verkkoyhteyden mahdollisimman laajalla maantieteellisellä alueella.

Maantieteellisestä paikasta riippuen käytettävissä on erilaisilla teknologioilla toteutettuja mobiiliverkkoja ja mobiililaitteissa on niitä varten erilaisia yhteystekniikoita. Näiden yhteyksien hallinta on haasteellista.

Multihomed-laitteesta voi olla monta erilaista reittiä mihin tahansa toiseen laitteeseen. Multihomed-laitteen useiden reittien hallinta on vastaava haaste kuin mobiililaitteiden useiden yhteystekniikoiden hallinta.

Tämä väitöskirja, jonka nimi on Mobiilien ja multihomed-laitteiden verkkoliityntöjen hallinta, tutkii erilaisia lähestymistapoja yhteystekniikoiden ja polkujen hallintaan mobiili- ja multihomed-laitteissa. Analysoimme mobiilin IPv6-protokollan käyttäytymistä ja ehdotamme rajapinnanhallintasysteemiä, joka ottaa huomioon käyttäjän preferenssit ja saatavilla olevien verkkojen parametrit luodakseen mahdollisimman hyvän yhteyden verkkoon.

Päädyimme kahteen erilaiseen lähestymistapaan sisääntulevan liikenteen polunhallinnalle. Ensimmäinen lähestymistapa keskittyy polunhallintakäytäntöjen erottamiseen alla olevista multihoming-protokollista. Toinen lähestymistapa hyödyntää verkon datagrammien sisäistä yhteyksien tunnistusta, jolloin erityisiä käytäntöjä sisääntulevan liikenteen polunvalintaan ei tarvita.

REFERENCES

[1] Nautilus 6 Project. `http://www.nautilus6.org/`, May 2010.

[2] IEEE Std 802.21-2008. Part 21: Media Independent Handover Services, 2009.

[3] J. Abley, B. Black, and V. Gill. Goals for IPv6 Site-Multihoming Architectures. RFC 3582 (Informational), August 2003.

[4] B. Aboba. Architectural Implications of Link Indications. RFC 4907 (Informational), June 2007.

[5] R. Atkinson, S. Bhatti, and S. Hailes. Mobility through naming: impact on dns. In *MobiArch '08: Proceedings of the 3rd international workshop on Mobility in the evolving internet architecture*, pages 7–12, New York, NY, USA, 2008. ACM.

[6] Marcelo Bagnulo. Updating RFC 3484 for multihoming support. In *IETF Internet-Draft, draft-bagnulo-rfc3484-update*, June 2006.

[7] Marcelo Bagnulo. Default Locator-pair selection algorithm for the SHIM6 protocol. In *IETF Internet-Draft, draft-ietf-shim6-locator-pair-selection-04*, October 2008.

[8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Service. RFC 2475 (Informational), December 1998. Updated by RFC 3260.

[9] Jean-Marie Bonnin, Imed Lassoued, and Zied Ben Hamouda. Automatic multi-interface management through profile handling. *Mob. Netw. Appl.*, 14(1):4–17, 2009.

[10] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. RFC 1633 (Informational), June 1994.

[11] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205 (Proposed Standard), September 1997. Updated by RFCs 2750, 3936, 4495.

[12] Mostafa Zaman Chowdhury, Yeong Min Jang, Choong Sub Ji, Sunwoong Choi, Hongseok Jeon, Junghoon Jee, and Changmin Park. Interface selection for power management in umts/wlan overlaying network. In *ICACT'09: Proceedings of the 11th international conference on Advanced Communication Technology*, pages 795–799, Piscataway, NJ, USA, 2009. IEEE Press.

[13] Cedric de Launois, Olivier Bonaventure, Marc Lobelle, and Université Catholique De Louvain. The naros approach for ipv6 multi-homing with traffic engineering. In *in Proceedings of QoFIS, LNCS 2811*, pages 112–121. Springer-Verlag, 2003.

[14] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Draft Standard), December 1998. Updated by RFCs 5095, 5722, 5871.

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

[16] R. Draves. Default Address Selection for Internet Protocol version 6 (IPv6). RFC 3484 (Proposed Standard), February 2003.

[17] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 3315 (Proposed Standard), July 2003. Updated by RFCs 4361, 5494.

[18] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry. The COPS (Common Open Policy Service) Protocol. RFC 2748 (Proposed Standard), January 2000. Updated by RFC 4261.

[19] Dino Farinacci, Vince Fuller, Darrel Lewis, and David Meyer. LISP Mobility Architecture. In *IETF Internet-Draft, draft-meyer-lisp-mn-00.txt*, July 2009.

[20] Dino Farinacci, Vince Fuller, Darrel Lewis, and David Meyer. LISP Alternative Topology (LISP+ALT). In *IETF Internet-Draft, draft-ietf-lisp-alt-02.txt*, January 2010.

[21] Dino Farinacci, Vince Fuller, Darrel Lewis, and David Meyer. Locator/ID Separation Protocol (LISP). In *IETF Internet-Draft, draft-ietf-lisp-06.txt*, January 2010.

[22] Alan Ford, Costin Raiciu, and Mark Handley. TCP Extensions for Multipath Operation with Multiple Addresses. In *IETF Internet-Draft, draft-ford-mptcp-multiaddressed-02*, October 2009.

[23] I. Ganchev, G. Morabito, R. Narcisi, N. Passas, S. Paskalis, V. Friderikos, A. S. Jahan, E. Tsontsis, C. Bader, J. Rotrou, and H. Chaouchi. Always best connected enabled 4g wireless world. In *in IST Mobile and Wireless Communications Summit 2003*, 2003.

[24] C.-L. Hwang and K. Yoon. *Multiple Attribute Decision Making*. Springer-Verlag, 1981.

[25] Janardhan R. Iyengar, Paul D. Amer, and Randall Stewart. Concurrent multipath transfer using sctp multihoming over independent end-to-end paths. *IEEE/ACM Trans. Netw.*, 14(5):951–964, 2006.

[26] H. Jang, J. Jee, Y. Han, S. Park, and J. Cha. Mobile IPv6 Fast Handovers over IEEE 802.16e Networks. RFC 5270 (Informational), June 2008.

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

[28] Srikanth Kandula, Dina Katabi, Shantanu Sinha, and Arthur Berger. Dynamic load balancing without packet reordering. *SIGCOMM Comput. Commun. Rev.*, 37(2):51–62, 2007.

[29] E. Kohler, M. Handley, and S. Floyd. Datagram Congestion Control Protocol (DCCP). RFC 4340 (Proposed Standard), March 2006.

[30] Eddie Kohler. Generalized Connections in the Datagram Congestion Control Protocol. In *IETF Internet-Draft, draft-kohler-dccp-mobility-02*, June 2006.

[31] Miika Komu, Marcelo Bagnulo, Kristian Slavov, and Shinta Sugimoto. Socket Application Program Interface (API) for Multihoming Shim. In *IETF Internet-Draft, draft-ietf-shim6-multihome-shim-api-09*, July 2009.

[32] R. Koodli. Mobile IPv6 Fast Handovers. RFC 5568 (Proposed Standard), July 2009.

[33] S. Krishnan, N. Montavont, E. Njedjou, S. Veerepalli, and A. Yegin. Link-Layer Event Notifications for Detecting Network Attachments. RFC 4957 (Informational), August 2007.

[34] Conny Larsson, Michael Eriksson, Koshiro Mitsuya, Kazuyuki Tasaka, and Romain Kuntz. Flow Distribution Rule Language for Multi-Access Nodes. In *IETF Internet-Draft, draft-larsson-mext-flow-distribution-rules-02*, February 2009.

[35] Qing Li, Tatuya Jinmei, and Keiichi Shima. *Mobile IPv6: Protocols and Implementation*. Morgan Kaufmann, 2009.

[36] M. Liebsch, A. Singh, H. Chaskar, D. Funato, and E. Shim. Candidate Access Router Discovery (CARD). RFC 4066 (Experimental), July 2005.

[37] Changming Ma and Ka-Cheong Leung. Improving tcp reordering robustness in multipath networks. In *LCN '04: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, pages 409–410, Washington, DC, USA, 2004. IEEE Computer Society.

[38] P. McCann. Mobile IPv6 Fast Handovers for 802.11 Networks. RFC 4260 (Informational), November 2005.

[39] Danny McPherson, Shane Amante, and Lixia Zhang. The Intra-domain BGP Scaling Problem. In *RIPE 58, Amsterdam*, May 2009.

[40] Koshiro Mitsuya, Romain Kuntz, Shinta Sugimoto, Ryuji Wakikawa, and Jun Murai. A policy management framework for flow distribution on multihomed end nodes. In *MobiArch '07: Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*, pages 1–7, New York, NY, USA, 2007. ACM.

[41] Nicolas Montavont and Thomas Noel. Handover management for mobile nodes in ipv6 networks. In *IEEE Communication Magazine*, pages 38–43, 2002.

[42] Nicolas Montavont and Thomas Noel. Stronger interaction between link layer and network layer for an optimized mobility management in heterogeneous ipv6 networks. *Pervasive and Mobile Computing*, 2(3):233 – 261, 2006.

[43] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson. Host Identity Protocol. RFC 5201 (Experimental), April 2008.

[44] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor Discovery for IP version 6 (IPv6). RFC 4861 (Draft Standard), September 2007.

[45] N. Nasser, A. Hasswa, and H. Hassanein. Handoffs in fourth generation heterogeneous networks. *Communications Magazine, IEEE*, 44(10):96 –103, oct. 2006.

[46] Quoc-Thinh Nguyen-Vuong, Nazim Agoulmine, and Yacine Ghamri-Doudane. A user-centric and context-aware solution to interface management and access network selection in heterogeneous wireless environments. *Computer Networks*, 52(18):3358 – 3372, 2008.

[47] P. Nikander, T. Henderson, C. Vogt, and J. Arkko. End-Host Mobility and Multihoming with the Host Identity Protocol. RFC 5206 (Experimental), April 2008.

[48] E. Nordmark and M. Bagnulo. Shim6: Level 3 Multihoming Shim Protocol for IPv6. RFC 5533 (Proposed Standard), June 2009.

[49] C. Perkins. IP Mobility Support for IPv4. RFC 3344 (Proposed Standard), August 2002.

[50] Sebastien Pierrel, Petri Jokela, and Jan Melen. Simultaneous Multi-Access extension to the Host Identity Protocol. In *IETF Internet-Draft, draft-pierrel-hip-sima-00*, June 2006.

[51] Bruno Quoitin, Luigi Iannone, Cédric de Launois, and Olivier Bonaventure. Evaluating the benefits of the locator/identifier separation. In *MobiArch '07: Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*, pages 1–6, New York, NY, USA, 2007. ACM.

[52] Michael Scharf and Alan Ford. MPTCP Application Interface Considerations. In *IETF Internet-Draft, draft-scharf-mptcp-api-00*, October 2009.

[53] Jason Schiller. Inter-AS Traffic Engineering Case Studies as Requirements for IPv6 Multihoming Solutions . In *North American Network Operator's Group, NANOG 34*, May 2005.

[54] Henning Schulzrinne and Elin Wedlund. Application-layer mobility using sip. *Mobile Computing and Communications Review*, 4:47–57, 2000.

[55] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier. Hierarchical Mobile IPv6 (HMIPv6) Mobility Management. RFC 5380 (Proposed Standard), October 2008.

[56] Hesham Soliman, George Tsirtsis, Nicolas Montavont, Gerardo Giarreta, and Koojana Kuladinithi. Flow Bindings in Mobile IPv6 and NEMO Basic Support. In *IETF Internet-Draft, draft-ietf-mext-flow-binding-04*, November 2009.

[57] R. Stewart. Stream Control Transmission Protocol. RFC 4960 (Proposed Standard), September 2007.

[58] R. Stewart, Q. Xie, M. Tuexen, S. Maruyama, and M. Kozuka. Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration. RFC 5061 (Proposed Standard), September 2007.

[59] Randall R. Stewart, Kacheong Poon, Michael Tuexen, Vladislav Yasevich, and Peter Lei. Sockets API Extensions for Stream Control Transmission Protocol. In *IETF Internet-Draft, draft-ietf-tsvwg-sctpsocket-19*, February 2009.

[60] G.N. Stone, B. Lundy, and G.G. Xie. Network policy languages: a survey and a new approach. In *IEEE Network*, volume 15, pages 10–21, January 2001.

[61] Miska Sulander, Timo Hämäläinen, Ari Viinikainen, and Jani Puttonen. Flow-based fast handover method for mobile ipv6 network. In *IEEE 59th Vehicular Technology Conference, 2004. VTC 2004-Spring*, volume 5, pages 2447–2451, 2004.

[62] F. Teraoka, K. Gogo, K. Mitsuya, R. Shibui, and K. Mitani. Unified Layer 2 (L2) Abstractions for Layer 3 (L3)-Driven Fast Handover. RFC 5184 (Experimental), May 2008.

[63] S. Thomson, T. Narten, and T. Jinmei. IPv6 Stateless Address Autoconfiguration. RFC 4862 (Draft Standard), September 2007.

[64] Umar Toseef, Asanga Udugama, Carmelita Goerg, Changpeng Fan, and Frank Pittmann. Realization of multiple access interface management and flow mobility in ipv6. In *MOBILWARE '08: Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications*, pages 1–8. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.

[65] George Tsirtsis, Gerardo Giarreta, Hesham Soliman, and Nicolas Montavont. Definition of Binary Filter Description. In *IETF Internet-Draft, draft-tsirtsis-mext-binary-filters-00*, May 2009.

[66] R. Wakikawa, V. Devarapalli, G. Tsirtsis, T. Ernst, and K. Nagami. Multiple Care-of Addresses Registration. RFC 5648 (Proposed Standard), October 2009.

[67] Matthias Wellens, Janne Riihijärvi, Krisakorn Rerkrai, Marten Bandholz, and Petri Mähönen. Enabling seamless vertical handovers using unified link-layer api. In *Mobility '06: Proceedings of the 3rd international conference on Mobile technology, applications & systems*, page 12, New York, NY, USA, 2006. ACM.

[68] ITU-T Y.2001. Next Generation Networks - Frameworks and functional architecture models, General overview of NGN, 2004.

[69] ITU-T Y.2011. Next Generation Networks - Frameworks and functional architecture models, General principles and general reference model for Next Generation Networks, 2004.

[70] Jukka Ylitalo, Tony Jokikyyny, Tero Kauppinen, Antti J. Tuominen, and Jaakko Laine. Dynamic network interface selection in multihomed mobile hosts. In *HICSS '03: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 9*, page 315, Washington, DC, USA, 2003. IEEE Computer Society.

[71] H. Yokota and G. Dommety. Mobile IPv6 Fast Handovers for 3G CDMA Networks. RFC 5271 (Informational), June 2008.

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.

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.

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.

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, ELENAV., 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.