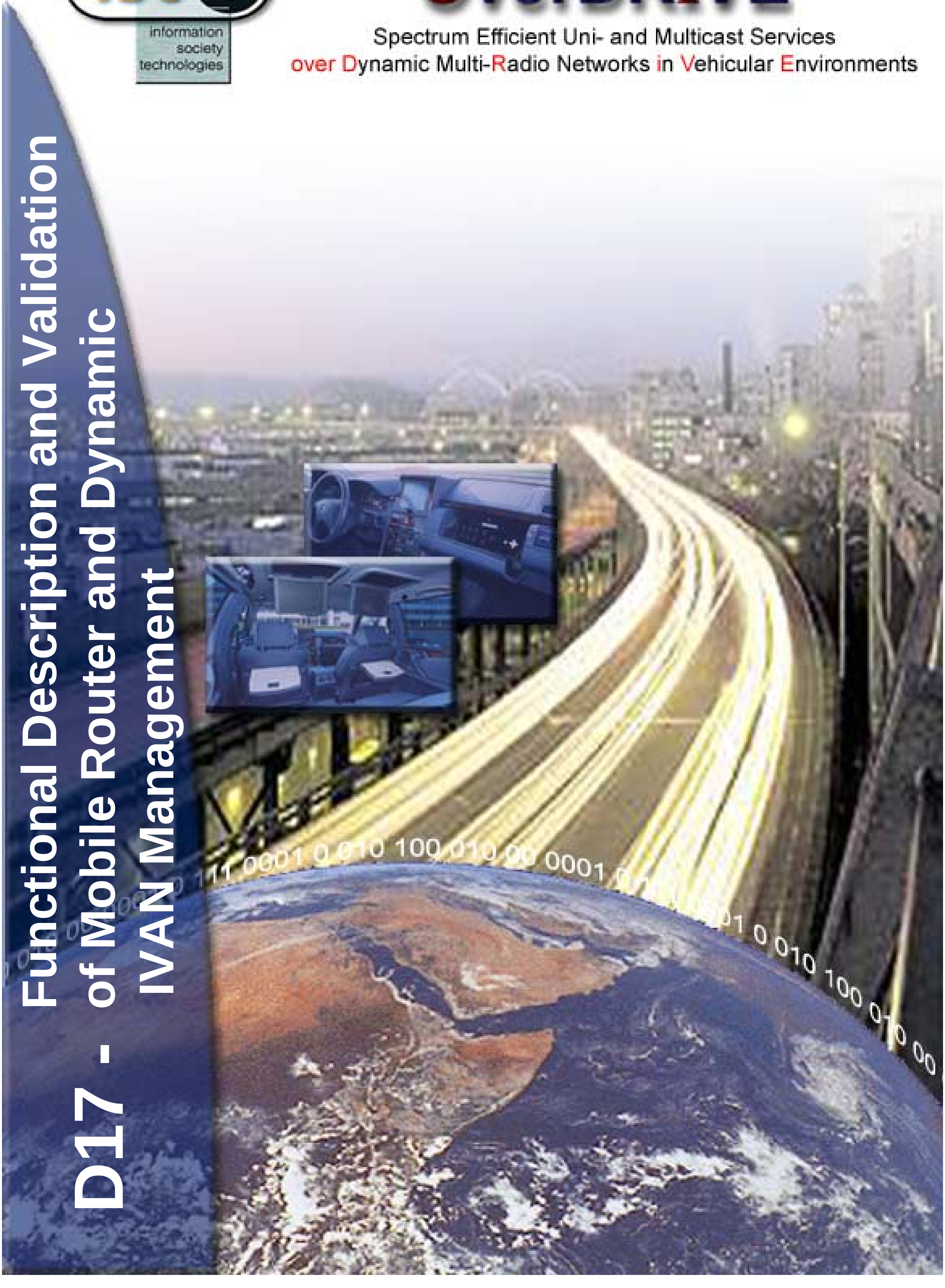
OverDRiVE

Spectrum Efficient Uni- and Multicast Services
over Dynamic Multi-Radio Networks in Vehicular Environments



D17 - Functional Description and Validation of Mobile Router and Dynamic IVAN Management

IST-2001-35125 (OverDRiVE)

D17

Functional Description and Validation of Mobile Router and Dynamic IVAN Management

Contractual Date of Delivery to the CEC:	03/2004 (Project Month #24)
Actual Date of Delivery to the CEC	03/2004
Author(s):	Markus Pilz (Editor, UBN) Christoph Barz (Editor, UBN) Jens Tölle (UBN) Matthias Frank (UBN) Wolfgang Hansmann (UBN) Miklós Aurél Rónai (ETH) Ágoston Szabó (ETH) Kristóf Fodor (ETH) Tim Leinmüller (DC) Michael Wolf (DC) Alexandru Petrescu (CRM)
Participant(s):	Ericsson, DaimlerChrysler, Motorola, University of Bonn
Project Title:	Functional Description and Validation of Mobile Router and Dynamic IVAN Management
Workpackage contributing to the Document:	WP3
Estimated Person Month:	48
Security Type: (Int/Res/IST/FP5/Pub)[1]	Pub
Document Number[2]**:**	IST-2001-35125/OverDRiVE/WP3/D17
Nature of the Document[3]**:**	(R)eport
Version (Status of the Document: D1/R1/D2/R2/F)[4]**:**	V1.0 (F)

[1] *Int* *Internal circulation within project (and Commission project Officer if requested)*
Res *Restricted circulation list (specify in footnote) and Commission PO only*
IST *Circulation within IST Programme participants*
FP5 *Circulation within Framework Programme participants*
Pub *Public document*

[2] Format: IST-2001-35125/OverDRiVE/<source>/<Deliverable Number: Dxx | Running Number for other: Rxx>:
Example: IST-2001-35125/OverDRiVE /WP1/D01 (Document comes from the WP1)

[3] (R)eport, (P)rototype, (D)emonstrator, (S)pecification, (T)ool, (O)ther

[4] V0.x=Draft, V1.x=Final. (D1=First Draft, R1=Technically Revised, D2=Final Draft, R2=Final Revised, F=Final)

Total number of pages:	125

Abstract: This deliverable describes the OverDRiVE functional description and validation of concepts in order to connect an IPv6 based network inside a vehicle to the Internet. These descriptions also include extensions of concepts for the Mobile Router and the Dynamic IVAN Management. In the scope of mobility management a close look on route optimizations and threats is given. Based on these foundations, implementations supporting mobility of an entire network are validated. As an optimized mobility management inside large IVANs is favoured, also interactions of micro-mobility approaches are included. Dynamic IVAN Management covers simulations about route optimizations for OverDRiVE mobile networks and the traffic management based on measuring bandwidth. Bandwidth estimations are validated on an analytical basis and within field trials.

Keyword List: mobility management, mobile networks, moving networks**,** mobile router, tunnelling, mobile multimedia, bandwidth measurements, route optimization, threat analysis, packet pairs

Authors

Markus Pilz (University of Bonn, UBN),
Christoph Barz (University of Bonn, UBN),
Jens Tölle (University of Bonn, UBN),
Matthias Frank (University of Bonn, UBN),
Wolfgang Hansmann (University of Bonn, UBN),
Miklós Aurél Rónai (Ericsson Research Hungary, Traffic Lab, ETH),
Ágoston Szabó (Ericsson Research Hungary, Traffic Lab, ETH),
Kristóf Fodor (Ericsson Research Hungary, Traffic Lab, ETH)
Tim Leinmüller (DaimlerChrysler Research, DC)
Michael Wolf (DaimlerChrysler Research, DC)
Alexandru Petrescu (Motorola Labs, CRM)

Revision History

Revision	Date	Issued by	Description
V0.1	2004/01/09	Markus Pilz	Started Document
	2004-01-28	Miklós Aurél Rónai	ETH authors and existing text of ETH part added
	2004-02-04	M. Pilz, M. A. Rónai	Refined structure, editorial corrections (LB)
	2004-02-23	M. A. Rónai, K. Fodor	Draft version of ETH part added
	2004-02-27	M. A. Rónai, K. Fodor	Finalized ETH part
V0.2	2004-03-09	C. Barz, M. Pilz	Validation of traffic management concept added
V0.3	2004-03-10	Michael Wolf, Tim Leinmüller	Introduction, MR Introduction, RO part (incl. references),
V0.3-ETH	2004-03-11	M. A. Rónai	ETH’s source code moved to Appendix
V0.4-AP	2004-03-19	A. Petrescu	Added section on NEMO threat analysis
V0.5	2004-03-22	C. Barz, M. Pilz	First version for technical review
V0.5	2004-03-23	A. Petrescu	Added sections on RO, traceroute and ping appendices, created the D17 Companion with ping measurements.
V0.6	2004-03-25	C. Barz, M.Pilz	Released new version for integration of reviews
V0.7	2004-03-26	C. Barz, M. Pilz	Released new version for political review
V0.8	2004-03-30	C. Barz, M. Pilz	Editorial tasks and minor modification (typos, image formatting,…)
V1.0	2004-03-31	C. Barz, M. Pilz	Editorial task and minor modification

“Functional Description and Validation of Mobile Router and Dynamic IVAN Management” - Table of Contents

Acronyms

AAA	Authentication, Authorisation, Accounting
AAAF	Foreign AAA server
AAAH	Home AAA server
AAAL	Local AAA server
ACS	Access System
ANP	ANchor Point
AR	Access Router
BAck	Binding Acknowledgement
BAN	Body Area Network
BCMP	BRAIN Candidate Mobility Protocol
BG	Border gateway
BGP	Border Gateway Protocol
BR	Border Router
BRAIN	Broadband Radio Access for IP based Networks
BReq	Binding Request
CAN	Controller Area Network
CBQ	Class Based Queuing
CN	Correspondent Node
CoA	Care-of-Address
CPU	Central Processing Unit
DAD	Duplicate Address Detection
DRR	Deficit Round Robin
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service
DRiVE	Dynamic Radio for IP-Services in Vehicular Environments
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
DVB	Digital Video Broadcast
DVB-T	Terrestrial Digital Video Broadcasting
EAP	Extensible Authentication Protocol
EAPOL	EAP Over LAN
FMIP	Fast Mobile IP
GPRS	General Packet Radio Service
GSR	Global State Routing
GW	Gateway
HA	Home Agent
HAWAII	Handoff-Aware Wireless Access Internet Infrastructure
HBR	Host Based Routing
HMIP	Hierarchical Mobile IP
HMIPv6	Hierarchical Mobile IP version 6
HO	Handover
HoA	Home Address
HSR	Hierarchical State Routing
ICEBERG	Internet Core Beyond the Third Generation
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPSEC	Internet Security Protocol

IPv4	IP version 4
IPv6	IP version 6
ISP	Internet Service Provider
IST	Information Society Technologies
IVAN	Intra Vehicular Area Network
LAN	Local Area Network
LCoA	Local Care-of-Address
LFN	Local Fixed Node
LMN	Local Mobile Node
MANET	Mobile Ad hoc NETworks
MAP	Mobility Anchor Point
MER-TORA	Mobile Enhance Routed – Temporally Ordered Routing Algorithm
MH	Mobile Host
MHTP	Multi Homing Translation Protocol
MIND	Mobile IP-based Network Development
MIP	Mobile IP
MIPv6	Mobile IP version 6
MLD	Multicast Listener Discovery
MMP	Multicast for Mobility Protocol
MN	Mobile Node
MONET	MObile NETwork
MOST	Media Oriented System Transport
MR	Mobile Router
MRHA	Mobile Router – Home Agent bidirectional tunnel
MRTP	Mobile Router Tunnelling Protocol
MS	Mobile Station
MSN	Multi Access Support Node
MT	Mobile Terminal
MTU	Maximum Transfer Unit
MU	Mobile User
MURP	MIND User Registration Protocol
NAT	Native Address Translation
ND	Neighbour Discovery
NEMO	NEtwork MObility (working group)
ORC	Optimized Route Cache Management Protocol for Network Mobility
OSPF	Open Shortest Path First
OverDRiVE	Spectrum Efficient Uni- and Multicast Services over Dynamic Multi-Radio Networks in Vehicular Environments
PAN	Personal Area Network
PANA	Protocol for carrying Authentication for Network Access
PDA	Personal Digital Assistant
PIM-{SM|DM}	Protocol Independent Multicast – {Sparse Mode | Dense Mode}
PPP	Point-to-Point Protocol
PSBU	Prefix Scoped Binding Updates
QoS	Quality of Service
RA	Router Advertisement
RCoA	Regional Care-of-Address
RFC	Request For Comments
RIP	Routing Information Protocol
RIPng	Routing Information Protocol next generation
SIP	Session Initiation Protocol

TCP	Transmission Control Protocol
THEMA	Transparent Hierarchical Mobility Agents
TLA	Top Level Aggregation
TORA	Temporally Ordered Routing Algorithm
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunication System
UR	User Registry
USB	Universal Serial Bus
UTRAN	UMTS Terrestrial Radio Access Network
VMN	Visiting Mobile Node
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network

Executive Summary

The OverDRiVE project aims at UMTS enhancements and co-ordination of existing radio networks into a hybrid network to ensure spectrum efficient provision of mobile multimedia services. An IPv6 based architecture enables interworking of cellular and broadcast networks in a common frequency range with dynamic spectrum allocation (DSA). The project objective is to enable and demonstrate the delivery of spectrum efficient multi- and unicast services to vehicles. OverDRiVE issues are:

- Improve spectrum efficiency by system coexistence in one frequency band and DSA.
- Enable mobile multicast by UMTS enhancements and multiradio multicast group management.
- Develop a mobile router supporting roaming into the Intra-Vehicular Area Network (IVAN).

This report focuses on the last point by investigating the idea of mobile hosts and mobile networks in vehicular environments. This research takes into account scenarios like smaller networks in passenger cars up to mobile networks in public transport vehicles (e.g. buses and trains) that comprise of a dynamic number of nodes. In these scenarios vehicles are seen as moving IPv6 networks which can use several access technologies to provide Internet connectivity. OverDRiVE denotes these networks as IVANs (Intra Vehicular Area Networks). The objective of OverDRiVE within this vast research area is to concentrate on two topics: Mobility management and IVAN management. Mobility management includes special issues like nesting of mobile networks, mobility within large vehicles, and optimized use of the network in terms of routing. Moreover, IVAN management tasks are regarded with respect to the effect of mobility management on transport protocols, bandwidth scheduling and protection against congestion. These topics are taken from the field of authentication, authorization and accounting (AAA) and reflect the issues, which have to be considered in heterogeneous wireless environments.

The functional description and implementation of the developed Mobile Router (MR) concept fits in OverDRiVE's envisioned scenarios, reaching form basic constellations up to scenarios that include nested mobility, multi-access and micro mobility. The mobility management is based on bi-directional tunnelling between the Mobile Router and its Home Agent (MR-HA). This concept extends the well-known Mobile IPv6 host mobility approach to handle the network mobility. Complementing this approach, issues with respect to route optimization and micro-mobility are investigated. As the MRHA tunnel prevents the packets from taking the optimal path to a correspondent node, solutions in this field are described; including the optimization of in-vehicular traffic from and to visiting mobile nodes.A threat analysis for the basic mobility solution is given to aid the further process of route optimization.

The Dynamic IVAN Management tries to tackle complementary issues for OverDRiVE moving networks. On one hand, a particular concept of route optimization for moving networks is analysed by means of simulation. Here, the main topic is the influence of route optimization of transport layer communication. On the other hand, reflections of the traffic management concept are made. The utilization of comparable low bandwidth wireless links to connect whole networks to the Internet requires concepts to avoid overstressing the wireless link. The refinement of OverDRiVE's traffic management concept presented, integrates bandwidth measurement.

Finally, the validation of OverDRiVE concepts is shown in this document. This includes descriptions and implementation details of two implementations of the Mobile Router. This is followed by extensions with respect to GPRS, UMTS, vertical handover, fast handover, local

fixed node and multicast support. In order to understand the handover outage measurements are presented. They are performed in the moving network testbeds including UMTS/WCDMA measurement results. Furthermore, the traffic management testbed and the realization of the concepts are presented. The bandwidth measurements take a close look on estimates about links using different packet sizes. An analytical model underpins these measurements.

1 Introduction

The European research project OverDRiVE (Spectrum Efficient Uni- and Multicast Services Over Dynamic Radio Networks in Vehicular Environments) aims at UMTS enhancements and coordination of existing radio networks into a hybrid network to ensure spectrum efficient provision of mobile multimedia services. An IPv6 based architecture enables inter-working of cellular and broadcast networks in a common frequency range with dynamic spectrum allocation (DSA). The project objective is to enable and demonstrate the delivery of spectrum efficient multi- and unicast services to vehicles. OverDRiVE issues are: (i) improve spectrum efficiency by system coexistence in one frequency band and DSA, (ii) enable mobile multicast by UMTS enhancements and multi-radio multicast group management, and (iii) develop a vehicular router that supports roaming into the intra-vehicular area network (IVAN).

The work presented within this document covers the efforts done in work package 3 dealing with vehicular mobile router and dynamic IVNA management. Starting with Deliverable 03 [1], which defines the scenarios and requirements for the project subsequently Deliverable 07 [2] describes the basic findings and concepts for supporting moving networks and a dynamic IVAN management. The second half of the project was devoted to two working areas. First there were demonstration activities that are described in Deliverable 14. Besides that the work package worked further on validation and concepts and advanced topics that are in the realm to extend the work done in Deliverable 07. The results of that work are described in the following document. The term functional description is used to illustrate that working areas where on optimizations and further refinements of the concepts.

Subsequent to the introduction section a functional description of the mobile router is given that concentrates on route optimization, IPv4 inter-working issues and a threat analysis. The section 3 describes work on the topic of dynamic IVAN management. It covers mainly aspect of traffic management that are used to keep the IVAN in a stable condition and to allow for adaptive applications. Section 4 describes the validation activities that were mainly accomplished doing trials in a laboratory and public environment.

The appendices give a description of configuration used for validation purposes.

2 Functional Description of Mobile Router

2.1 Introduction

The basic solution of a mobile router as a central entity that manages mobility for moving networks was described in Deliverable 07 [2]. While this solution supports a wide range of network mobility configurations, it leaves out several issues especially with respect to route optimization. The MRHA tunnel prevents the packets from taking the optimal path to a correspondent node. Likewise the optimization of in-vehicular traffic from and to visiting mobile nodes was not analyzed in the previous deliverable. During trials, the dificulties of inter-working with IPv4 environments became evident since a wireless publicly-deployed native IPv6 infrastructure is far from being realized (i.e. all 2.5G, 3G and WiFi production – that is, commercially exploitable – networks deployed in EU are IPv4 exclusively). Finally, a comprehensive threat analysis for the basic network mobility solution is given.

2.2 Optimization

2.2.1 Route Optimization

Route Optimization (RO) in Mobile IPv6 environments addresses one of the major problems of IP based mobility, namely the traffic overhead resulting from packet tunnelling using non-optimal routes.

Route optimization is a well-known challenging problem of the Mobile IP family of protocols. Succinctly, the problem is induced by the artificial necessity to forward all packets between CN and LFN through the MH's HA, even if HA is not in the optimal (shortest) IP path between MH and CN.

Route optimization is a process that is used to enable packet delivery along the (topologically) shortest path between two communicating nodes. Basically, in Mobile IP scenarios this means to eliminate tunnelling over a Home Agent (HA) and to establish a direct connection between two communicating nodes.

Optimization for the communication between a single mobile host and a correspondent node is described in [13].

In the case of moving networks, the route optimization problem is much harder; while for Mobile Hosts the un-optimal path usually has a "triangle" shape involving MH, HA and CN (hence the name "triangular routing"), the moving networks shapes can be much more complicated due to the effects of nesting and the presence of several HA's. In short, the RO problem for MH is upper-bounded by a maximum level of 4 ("rectangular" paths of MH-to-MH communication) while the RO problem for MR is potentially not upper bounded leading to "multi-angular" paths.

2.2.1.1 IP End-to-End Distance

When evaluating optimal paths, it is important to have a meaningful view of what the "distance", or "length" of certain path segments is. Note first that physical distance between two IP nodes is entirely irrelevant to the IP path length between those two nodes. The practical experiments with a MH attaching to a WLAN hotspot area and to a GPRS network confirm this "irrelevance" assumption. The GPRS and WLAN HotSpot networks are relatively close in terms of physical distance: in a city, one often has visibility to both a GPRS Base Station and a WLAN Access Point antenna. However, in IP terms, this physical closeness is irrelevant. On one hand, even if the obtained IPv4 address from each system has the same form 10.x.y.z (and sometimes can even be the same e.g. 10.1.2.3), these addresses are most certainly not connected by a short IP path. The two networks are entirely different and independent, their first meeting point being somewhere on the worldwide Internet. The length of the direct IP path between the two addresses obtained at the same physical place (the number of IP-addressable hops) is quite large. This paradox of physical closeness but large distance in IP terms gives an intuitive example of the complexity of the aspects that should be paid attention to when considering Route Optimization enhancements.

Thus, we define the IP End-to-End distance between two communicating nodes as being the number of IP hops through which the application-level packets flow. This is, of course, a generalization that makes several simplifying assumptions:

- The path taken by the packet flow in one direction is the same as the path taken in the reverse direction.
- All intermediary point-to-point links have approximatively the same costs, bandwidth and are symmetric.
- Paths are stable during the entire application-level communication.

We used this simplifying definition of the IP path length when analyzing the benefits that can be obtained from using Route Optimization (see section on Measurements).

2.2.1.2 RO for MR is More Complex than for MH

For illustration, consider the two configurations depicted in Figure 1. In the left diagram, note the triangular shape composed by the two "legs" of the un-optimal path (pictured red) CN-HA-MH and by the optimal path (pictured blue) between CN and MH; in the right diagram, note the rectangular shape of the paths that involve the two HA's of two communicating MH's. The RO problem of Mobile IPv6 for Mobile Hosts can involve triangular or, at most, rectangular paths.

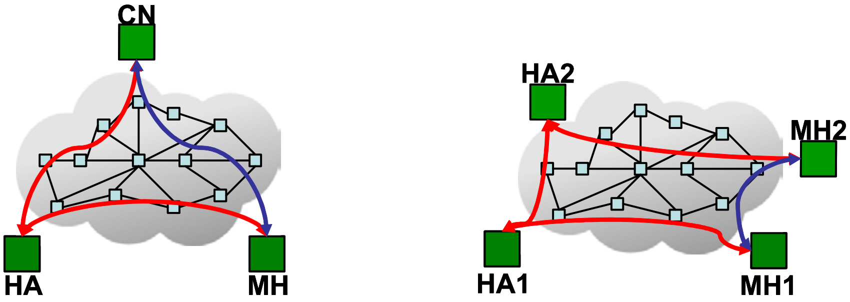


Figure 1: Route Optimization for MH: Triangular and Rectangular

For moving networks, we illustrate below two different cases of route optimization. The simplest case (pictured on the left diagram of the Figure 2 is no different than the triangular shape of Mobile IPv6 for Hosts in the left diagram of the Figure 1: Route Optimization for MH: Triangular and Rectangular; the communication between LFN and CN involves a short additional leg between MR and LFN (not pictured, because this additional short leg does factor in the RO problem since it is itself part of an optimal path within the moving network). Similarly, communication between two LFN's that are part of different moving networks, that are not nested, and that have different HA's is equivalent to the "Rectangular" shape of the MH RO problem (this case, again, not pictured). The diagram at the right of the picture below shows the shape the RO problem takes when two moving networks nest one under the other; again, red paths indicate the actual path of packets when using the NEMO base support, while the green paths indicate the shortest paths between the two LFN's.

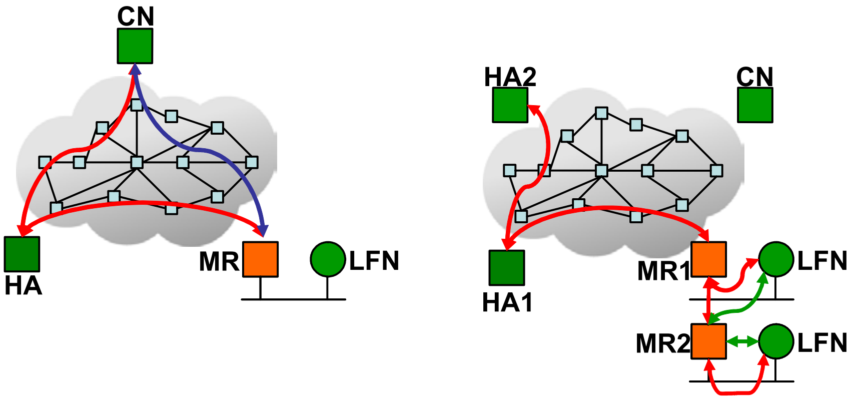


Figure 2: Route Optimization for MR: Triangular and Multi-angular Paths

From these illustration, it is clear that the RO problem for moving networks is not only inheriting all the complexity (e.g. security) of the RO problem space for Mobile Hosts, but it adds in new complications due to the theoretically arbitrary shapes that nested mobility configurations can take.

2.2.1.3 Tradeoffs in the RO Problem Space

Mobile IPv6 protocol enhancements for route optimization involve new signalling messages between MH, HA and CN; a main concern with this signalling is constituted by the potential security risks of attacker MH claiming the Home Address of a victim MH, when the victim communicates to CN. The RO procedure of Mobile IPv6 for hosts uses a three-party protocol which is inherently more complex than a two-party protocol.

Thus, it is important to identify the particular cases when using optimal paths between CN and MH does bring in actual benefits, worth the effort of the complexity of the initial binding setup procedure (CN and MH maintain bindings after the RO procedure was successful).

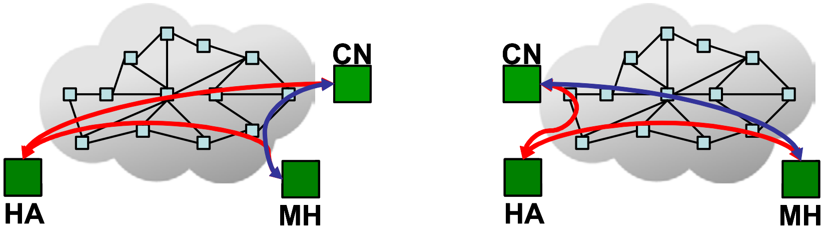


Figure 3: Usefulness of Optimal Paths

In Figure 3 note that when CN and MH are positioned relatively close one to the other and the HA is remote (left diagram), using an optimal path can bring in significant advantages in terms of communication delays and badwidth savings; however, this is not the case when CN and HA are close to each other while the MH is remote (right diagram).

It is expected that identifying the tradeoffs and benefits in using optimal paths with moving networks is a more complicated analytical exercise than the simple MH case presented above.

When designing route optimization protocols for moving networks, it is important to be able to identify the benefits and tradeoffs of such procedures; doing so analytically can be an intellectually challenging task; important help in this exercise can be found in practical experiments. In other sections of this document we use results of experiments performed during the Turin demonstration (cf. [3], [66]) to show that using optimal paths is necessary in some cases, but not all.

2.2.2 Defining the Problem Space

The earlier sub-section presented the general problem of RO in Mobile IPv6 and moving networks. In this subsection we describe the complex data path every packet has to take in the case of a simple scenario when a moving network is supported by the original Mobil IPv6 . Based on this example we outline some shortcomings of this solution.

Assume that a mobile node somewhere in the Internet (a user checking his vehicle's status from his work desk) has connections to several nodes within a mobile network (e.g. sensors, an onboard unit, or a car web-server) that do not support standard MIPv6 route optimization. This mobile node then joins the mobile network, i.e. the user enters the vehicle, and thus the mobile node becomes a visiting mobile node of this network. Without route optimization, the entire traffic between the mobile node and the local fixed nodes in the mobile network is routed via at least the home agent of the mobile node, or in case the mobile network is not at home, also via the mobile router's home agent. This is in many cases not acceptable, since in-vehicle entertainment, e.g. a video played on the car DVD player and displayed on a PDA display is a quite likely scenario. Without route optimization this is not possible at all (due to limited bandwidth of the wireless link or much too expensive)

To look into this problem in more detail, we consider a simplified example with only one ongoing communication between the mobile node and a (local fixed) node in the mobile network, as shown in Figure 4[5]. The mobile network is connected to a foreign access system and the mobile node is directly attached to one of the mobile network sub-nets. The packet flow of such a communication is depicted in Figure 5. This means, if the visiting mobile node sends a packet to the local fixed node, the visiting mobile node first encapsulates the packet and forwards it in direction towards its home agent. In this path, the mobile router receives this packet and decides that the packet destination is not inside its mobile network (the destination of the encapsulated packet is the mobile node's home agent) and thus encapsulates the packet once more and sends it to its MRHA. At the MRHA the packet is decapsulated once and forwarded to the mobile node's home agent. This one does the second decapsulation, and thus receives the original packet, which destination is the local fixed node. Therefore, it sends the packet in direction to the home link of the mobile network. On this link, the packet is intercepted by the MRHA and tunnelled to the MR, which finally delivers the packet to its destination, the local fixed node.

[5] We left out packet flow indicators in this figure due to visibility reasons.

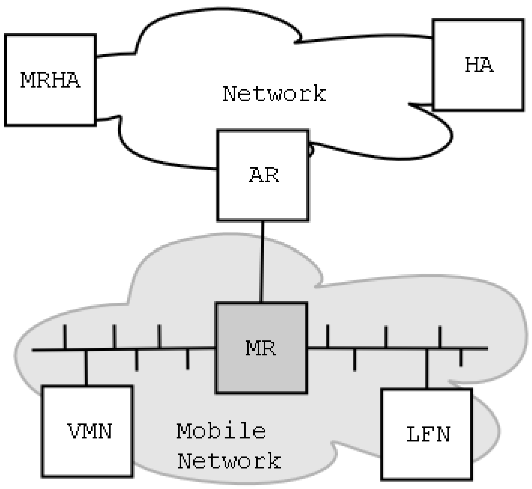


Figure 4: Simplified Example Scenario

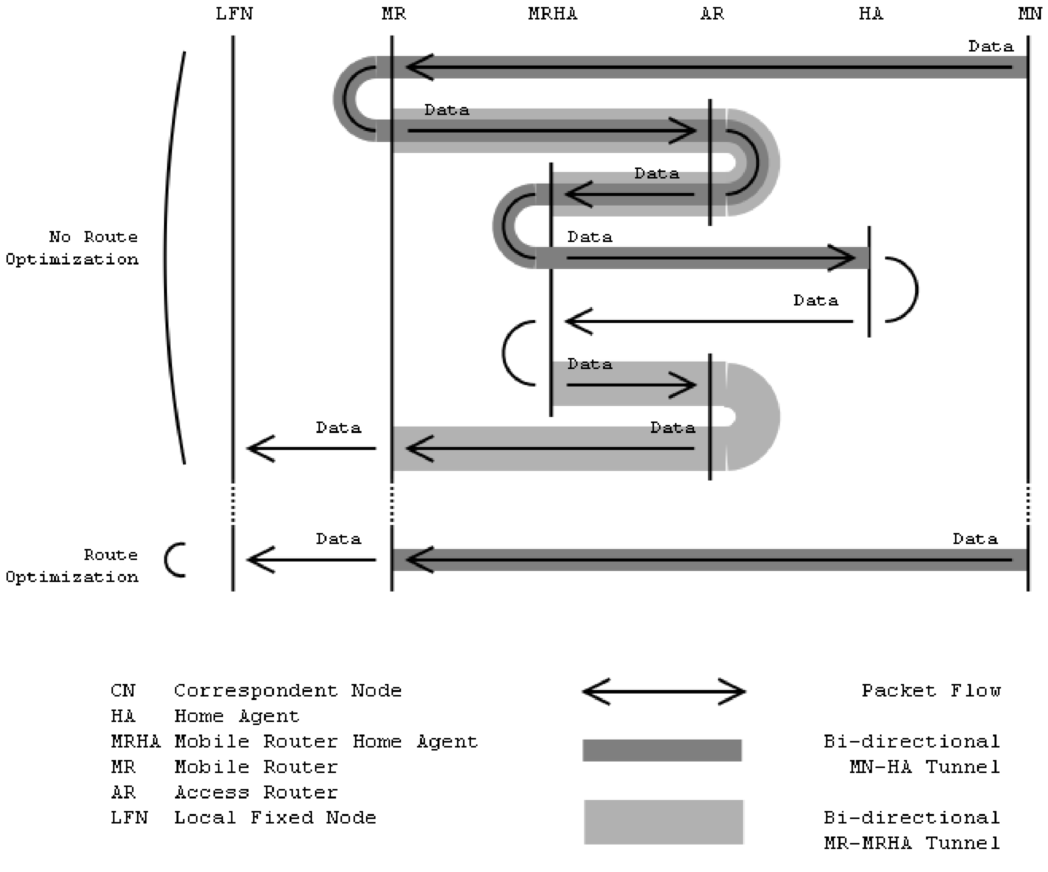


Figure 5: Packet Flow

Since in general the up-link connection of a mobile network, connected to any foreign access network, is a non high-speed connection, e.g. a UMTS connection as used in the OverDRiVE demonstrations, this kind of tunnelling is not desirable. Even more, apart from being a tremendous waste of scarce resources, in case of a breakdown of the up-link connection, communication between the mobile node and any other node within the mobile network is impossible.

2.2.3 Dividing the RO Problem Space

The above example is illustrative for the problem space in the case of one MH visiting a moving network. It further leads to the conceptual division (splitting) of the RO problem space in two large parts: (1) optimal paths within the moving network and (2) optimal paths between the Home Agents spread across the Internet.

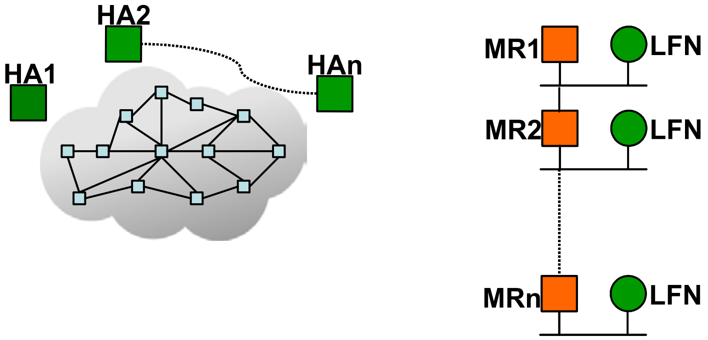


Figure 6: RO Problem Space Split

The left diagram in Figure 6 shows independent HA's placed at the edges of the Internet while the right diagram shows a nested aggregation of different moving networks; to each MR corresponds a different HA. A complete solution for Route Optimization for moving networks should encompass all paths of the two diagrams. Some proposed RO protocols only address the inner part of the moving network aggregation while other protocols deal with inter-HA route optimization exclusively.

As a side note, remark that in the case of a single moving network (non-nested aggregation), the problem of optimal paths within the moving network is hidden since it is assumed that all paths are optimal within that network; this is ensured by the use of administratively correct static routes or – when the moving network is large – by the proper use of a dynamic routing protocol.

2.2.4 Solutions

The IETF draft [22] tries to establish taxonomy on the route optimization problem area. Regarding to those documents the problem scope of this section relates to the case of MIPv6 route optimization over NEMO which in turn is a special case of nested mobile networks. With respect to [22] the approaches relevant to the scope of this section are Hierarchical MIPv6 (HMIPv6) [25] based approaches, route optimization based on prefix delegation [23] and route optimization based on neighbor discovery proxy functionality [24].

Hierarchical MIPv6 (HMIPv6) has been developed by Ericsson and INRIA. It is specified in an Internet-Draft [25] and was further developed regarding route optimization for mobile nodes in mobile networks [26]. A new Mobile IPv6 node, called mobility anchor point (MAP), is introduced, which can be located at any level in a hierarchical network of routers. In our scenario the MR would be a mobility anchor point. In HMIPv6, two different types of care-of addresses are distinguished: beside the topologically correct care-of address, called "local care-of address" (LCoA) in this context, a mobile node also obtains an address from a mobility anchor point referred to as the "regional care-of address" (RCoA). The RCoA is an address on the mobility anchor point's subnet. If there is more than one hierarchy level, a mobile node may even have several RCoAs. In theory, the correspondent nodes are not affected. The mobile anchor point essentially acts as a local home agent, limiting the signalling outside a local domain.

In [23] route optimization is reached via prefix delegation (PD) which in turn requires that the access routers support that protocol. Through prefix delegation the route could be held optimal by delegating sub-prefixes of the original prefix acquired from the access router down to a moving subnetwork. Naturally that approach is not well suited in fast changing network topologies since it would require the whole networks to reconfigure.

The approach described in [24] relies on the principle that the mobile router relays the prefix of its care-of address to its mobile nodes by playing the role of a neighbour discovery (ND)-proxy. Through binding updates associated with the network prefix of an access network, the mobile nodes can perform route optimization.

For the reason of completeness we would like to add also a short description of Prefix Scope Binding Updates (PSBU). The utilization of Prefix Scope Binding Updates has been proposed by MOTOROLA Labs Paris and INRIA. It is specified in an Internet-Draft [27]. Basically, a Prefix Scope Binding Update is an enhanced Mobile IPv6 Binding Update associating a care-of address with a prefix instead of a single address. It is assumed that all nodes in a moving network share a common prefix, and MR's ingress interface is configured with this prefix. As in MIPv6, MR's egress interface is configured with the home prefix (when the MR is at home) or with the care-of address received by a foreign access network. The draft does not consider visiting mobile nodes so that with respect to the scope of the paper a major requirement is not full filed.

One can say none of these drafts covers the case we set up in section 2.2.2 but only, if at all, route optimization between a mobile node and a correspondent or a local fixed node and a correspondent node (the correspondent node located in the infrastructure). The closest approach might be the one described in [26]. But it requires a HMIP infrastructure (HA, MAP, etc.) to be working with, whereas our approach keeps to optimization local within the moving network.

2.2.5 Route Optimization for Visiting Mobile Nodes

To exemplify our approach, we consider the example scenario as depicted in Figure 7 (based on the network example from Figure 4). The mobile network consists of two separate subnets, interconnected by the mobile router. The entire mobile network prefix is 48 bit long, the subnet prefixes are only 64 bit long. A visiting mobile node connects to subnet 1 and wants to communicate with a local fixed node in subnet 2. Without route optimization, the entire traffic resulting from this communication is tunnelled twice through the external link (the packet flow results in what has already been shown in Figure 5).

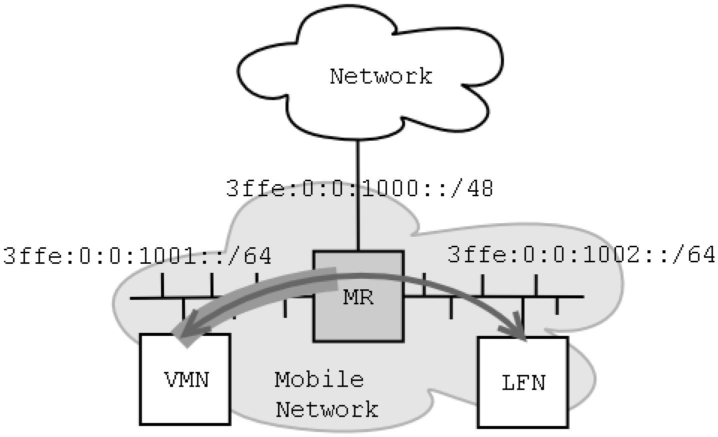


Figure 7: Example Scneario

When the mobile node joins the mobile network, the node detects its movement by the reception of a router advertisement that contains a previously unknown prefix. In addition to the IPv6 standard [28], the router advertisement in our solution has a supplementary option that contains also the prefix of the entire mobile network (from which, the prefix on the respective link is only a subset) as well as the MR's IPv6 address on this link (see subsection 2.2.5.1 for a detailed explanation of this option). This option is required since the visiting mobile node needs to know the entire prefix to be able to do route optimization for the entire mobile network, otherwise route optimization would only be possible for the respective subnet.

After the reception of the router advertisement, the visiting mobile node updates the bindings with its home agent first (as it usually does, according to [13]). But instead of trying to do route optimization directly with the local fixed nodes, the visiting mobile node sends a binding update message to the MR. The MR responds with a binding acknowledgment.

As soon as the registration is completed, packet transfers between the visiting mobile node and the local fixed nodes in the mobile network work as follows. When the visiting mobile node wants to send packets to local fixed nodes, it finds the MR's address in its binding cache, associated with the mobile network's prefix. That's why it decides to encapsulate the packets and to send them to the MR. The MR decapsulates the packets and forwards them to the destination nodes, which are the local fixed nodes.

Routing in the other direction, packets from local fixed nodes to a registered visiting mobile node is similar. The local fixed sends a packet to visiting mobile node's home address, which has to be routed via the MR. MR detects the visiting mobile node's home address in the IPv6 header destination address field and using its binding cache MR determines the actual care-of address of the visiting mobile node. MR encapsulates the packet and forwards it to the visiting mobile node, which finally decapsulates the packet.

In both directions, tunnelling between the MR and the visiting mobile node is necessary to maintain topological correct routing and addressing. Moreover, there might be intermediate routers between the MR and the visiting mobile node, that would not be aware of the visiting mobile node's care-of address.

2.2.5.1 Protocol Extensions

To be able to announce the presence of a MR that supports route optimization for visiting mobile nodes, we define a new option (see Figure 8 and Table 1) that has to be included in the router advertisements inside the mobile network. This option contains the entire mobile network prefix as well as one of the MR's internal IPv6 addresses. In case MR has several internal interfaces on different subnets, as shown in the example scenario, MR should send the address that is used on the respective interface.

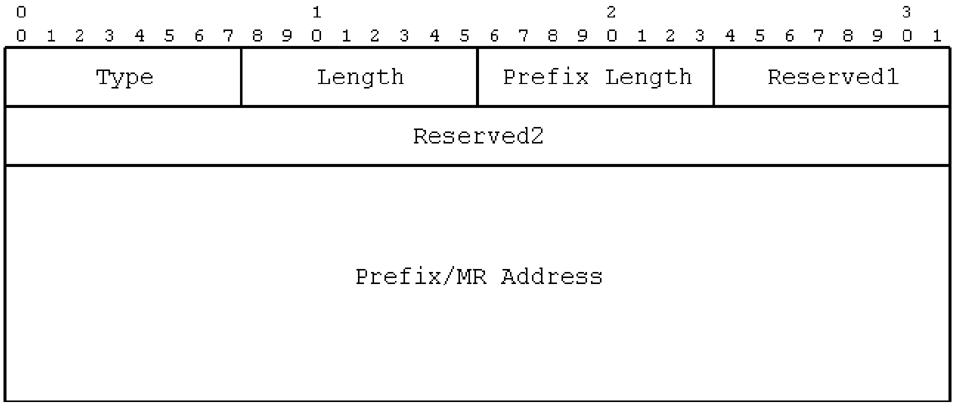


Figure 8: Mobile Network Prefix Option Format

Type	not yet assigned
Length	4
Prefix Length	8-bit unsigned integer. The number of leading bits in the Prefix/MR Address field that define the prefix. The value ranges from 0 to 128.
Reserved1	8-bit unused field. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.
Reserved2	32-bit unused field. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.
Prefix/MR Address	An IP address of the Mobile Router in the prefix space of the mobile network. The Prefix Length field contains the number of valid leading bits in the prefix. The bits in the prefix after the prefix length contain what is missing to complete the MR address.

Table 1: Message Format Description

2.2.5.2 Requirements on Participating Nodes

The MR should be configurable to send the new router advertisement option specified in subsection 2.2.5.1 to announce the mobile network's prefix and its own address. Furthermore, when sending this option, it should accept binding request from visiting mobile nodes and participate in route optimization as explained before.

A visiting mobile node should detect the new router advertisement option. On reception of a router advertisement containing this option, it must update its existing bindings and then it should send a binding update to the mobile router address that is indicated in the option. After the reception of a binding acknowledgment (i.e. the successful completion of the binding process), the visiting mobile node should act as described in this section and send packets to nodes inside the mobile network via the MR.

Intermediate routers inside the mobile network (especially routers that are meant to function as access routers for visiting mobile nodes) should be able to propagate the contents of the new router advertisement option to their sub-networks. This can be done either in a manual mode, or in an automatic mode. In manual mode, the contents of the option have to be configured by an administrator. In automatic mode the router detects the option in router advertisements of another router (e.g. the MR) and re-distributes it to its sub-networks.

2.2.5.3 Security Considerations

A security issue for MR is to accept binding updates from the visiting mobile node. This problem can be eliminated by using network access control for OverDRiVE mobile networks as it was described in [29]. Then we can assume that the MR is able to authenticate the visiting mobile node and vice versa. In case the authentication data of both, the visiting mobile node and the MR, has been previously store inside the mobile network, secured route optimization is also possible, if the mobile network is temporarily disconnected from the Internet.

2.2.5.4 Comparison to other Solutions

The advantages compared to other route optimization solutions for mobile networks are the following. Our approach works also in case the mobile network is at home. It works in disconnected mode, i.e. when the MR's up-link connection is interrupted or even totally broken. And, route optimization will only take place if visiting mobile node joins the mobile network, i.e. the optimization remains local inside the mobile network.

Compared to standard Mobile IPv6 we are able to provide route optimization for nodes that do not support route optimization themselves, i.e. for nodes that support only standard IPv6 (sensors or onboard units that are not upgradeable to support Mobile IPv6 route optimization).

2.3 Threat Analysis for Basic Network Mobility

In this section we describe the main results of the threat analysis of the basic network mobility support protocol. The threat analysis was performed in two iterations. First, during the inception phase of designing the basic NEMO support, several security threats and their respective defences were considered: attacks on the MNP at HA and the protection afforded by IPsec to protect the MRHA tunnel, spoofing of packets through the HA and the protection afforded by ingress filtering at HA. A second iteration of the analysis was performed after the protocol became more mature (being in IETF Last Call phase). This second iteration considered the exact message formats and message exchange diagrams to expose more details on the potential risks. It has thus been found that most threats are indeed protected by a correct use of a NEMO IPsec architecture but also that certain risks remain open, such as the location privacy of LFN's and the DHAAD R-bit.

2.3.1.1 First Iteration of Threat Analysis

During the first iteration, the base protocol was not yet fully specified; message formats were not yet defined. However, it was known that the protocol would use a bidirectional tunnel maintained by the MR and by the HA; it was also known that the MR would communicate prefix information, or a pointer to the prefix information, to the HA. When prefix information is communicated to the HA, the HA uses it to establish forwarding state (e.g. routes in its routing table) and binding cache state storing the current Care-of Address of the MR. The prefix information sent by MR in a Binding Update is normally the IPv6 prefix (or prefixes) of all nodes placed within the moving network.

The obvious security risk with this behaviour lies in that an attacker MR is able to send to HA the prefix information of a victim MR, together with the attacker MR's Care-of Address. The realization of this threat is that attacker might be able to "steal" traffic of an entire network (the victim MR's moving network), through the HA. More variations of this attack are possible, such as attacker using faked Home Address and faked Care-of Address.

Protection against this risk is afforded by using IPsec AH and ESP headers and by sharing common keys between each MR and its HA. If this is done, the Binding Updates sent by attacker

MR to HA that contain fake prefix information will be dropped by HA because of lack of a proper Security Association between the attacker MR and the HA. However, more complicated cases may exist, when attacker and legitimate MR's have correct SA's with the HA, for example when both MR's belong to the same HA within the same administrative domain. Against this latter type of threat, a new mechanism was introduced by the NEMO base support: a *Prefix Table* is used to check every Moving Network Prefix.

A second identified security risk during this iteration is the possibility of an LFN to use a fake source IPv6 address when sending packets to the Internet, while the moving network is not at home; the attack works as follows: LFN fakes the source address and sends to MR, MR encapsulates towards HA, HA decapsulates and forwards to the Internet. A protection against this threat was specified in the base NEMO support by requiring the MR and the HA to perform "ingress filtering" checks in the following way: the source address of a decapsulated packet must match the prefix information maintained in the routing table of MR and in the routing table and in the binding cache of the HA (the entries related to the Home Address and Care-of Address of the MR that controls the moving network where the LFN is found).

The original text of these requirements can be found in section 9 "Security Considerations" and section 6.1.2 "Prefix Table" of [19].

2.3.1.2 Second Iteration of Threat Analysis

The second iteration of threat analysis starts with the completely specified NEMO base protocol and looks at details of message formats and ways to apply IPsec headers to NEMO control messages (BU, Back, DHAAD Request and DHAAD Reply). The main results of this analysis are presented in [46]. We adapt parts of that text here for several reasons: (1) a personal Internet Draft has a short life span and (2) the threat analysis itself may further evolve together with other Internet Drafts on the same topic (see for example a different NEMO threat analysis in [47]).

The threats described herein pertain to: (1) interactions between MR and its HA, (2) interactions between several MR's served by same HA, (3) threats relating to forwarding information updates at HA and, finally and (4) threats related to nested mobility.

The current network mobility base specification [19] requires that all signalling messages between the Mobile Router and the Home Agent MUST be authenticated by IPsec. The use of IPsec to protect Mobile IPv6 signalling messages is described in detail in the HA-MN IPsec specification [48]. Using AH and/or ESP between MR and HA is of paramount importance in order to protect against a wide range of attacks.

The Internet IPsec architecture can be particularized for Mobile Routers by distinguishing two cases: (1) IPsec protection in transport mode for NEMO signalling and (2) IPsec protection in tunnel mode for NEMO traffic between LFN and CN.

AH/ESP used in transport mode for NEMO signalling protects Binding Updates, Binding Acknowledgements (but does not protect Home Agent Address Discovery Requests and Home Agent Address Discovery Replies). These messages are exchanged between the Mobile Router (Mobile Security Gateway) and Home Agent (Home Security Gateway), as presented in Figure 9.

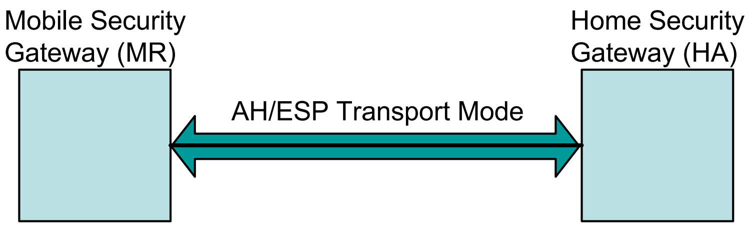


Figure 9: IPsec protection of BU and BAck

AH/ESP used in tunnel mode for NEMO traffic protects all fields of all IP datagrams exchanged between LFN and CN, including application-level data (see Figure 10).

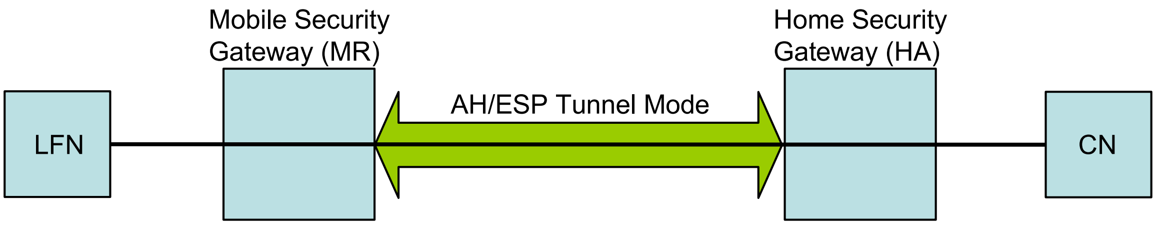


Figure 10: IPsec protection of Application Data

This IPsec architecture for moving networks can be extended to nested network mobility configurations, by means of encapsulating tunnelling. See section A for illustrations of IPsec for nested mobility.

Other means of protecting communication between MR and HA are needed in certain cases; they include the use of the NEMO Prefix Table, the prefix-extended ingress filtering technique [49] used by the NEMO Home Agent and the tunnel encapsulation limiting. If further additional tools are needed, a good overview of authentication mechanisms in the Internet can be found in [50].

Last but not least, even if IPsec, ingress filtering, Prefix Table and tunnel encapsulation limiting are used, we acknowledge the existence of other security risks with the NEMO base protocol. They stem mainly from the lack of certain security features of the underlying Mobile IPv6 protocol. For example: attacks on the *R* bit within the Home Agent Discovery messaging, and location privacy risks.

Interactions between MR and HA

T.1 Threat on the MNP field (redirection threat): the simplest and most important (but avoidable) threat of the NEMO basic support protocol is the redirection of traffic of all addresses within a Mobile Network Prefix. An attacker sending an un-protected NEMO Binding Update to a Home Agent for a certain MNP is actually instructing that Home Agent to forward all traffic for MNP towards the address of the attacker. The gravity of the risk is more important than in the case of Mobile Hosts; an attacker Mobile Host could re-direct only one legitimate Home Address, while with NEMO an attacker MR could re-direct all the addresses within an MNP. Moreover, the risk is all the more important since attacker MR can be positioned anywhere in the Internet, NEMO is not restrained to a closed system. In order to avoid this risk actually realizing, it is important to protect all signalling messages between MR and HA by IPsec (this is also required by [48] and [13]). In general, if HA uses AH/ESP transport mode for all NEMO signalling with the legitimate MR then attacker MR is not able to realize such a re-direction attack, because AH/ESP in transport mode covers the MNP field of the BU.

T.2 Threat on the R bit of BU: An attacker Mobile Host asks its Home Agent to forward all traffic addressed to addresses within MNP to its current Care-of Address. Normally, Mobile Hosts do not send the R-bit in the Binding Update. An attacker Mobile Host can specify the R-bit and thus receiving traffic addressed to other addresses than simply to its Home Address. However, if IPsec is used, the R-bit in the BU is covered both by AH and ESP in transport mode, so if HA and MH have a trust relationship it is assumed that MH will not specify the R bit.

T.3 Threat on the Status field of BAck: an attacker entity on the path between legitimate MR and HA modifies the Status field of the Binding Acknowledgement sent by the HA to MR. It is assumed that MR has previously sent a BU to HA with the R-bit set and that HA replied with Status 140 (Mobile Router Operation not Permitted). The attacker entity substitutes 141 (Invalid Prefix) for 140 and thus leads MR into re-sending Binding Updates to Home Agent (instead of stopping sending Binding Updates). However, the AH/ESP headers cover the Status field of the BAck and thus attacker can not tamper with the Status field, invalidating this threat.

T.4 Threat on switching between modes: MR sends BU in implicit mode to HA, HA replies back positively, using MNP from external means (not from BU). During this time, the attacker gained knowledge of the MR's Home Address, sends BU to HA in explicit mode for the same Home Address but a MNP specified in the BU, different than what HA already has. HA replies back positively to MR and switches to explicit mode and a different MNP. Threat is two-fold: on one hand HA would stop forwarding packets of the legitimate MNP towards the legitimate MR; on the other hand HA would start forwarding packets of a false MNP towards the legitimate MR.

IPsec can not help protecting against attacker MR obtaining the Home Address of the legitimate MR (it is not covered by ESP when legitimate MR sends BU or receives BAck). However, IPsec can protect against the attacker MR specifying an illegitimate MNP within a BU; the MNP field in the BU is covered by ESP in transport mode.

The description of this threat started with MR using implicit mode and attacker trying explicit mode. The description applies equally well if the initial step was in explicit mode and second step used implicit mode.

T.5 Threat on the *R* bit of Home Agent Discovery Request: an attacker on the path between legitimate MR and HA transforms the R bit from 1 to 0. The Home Agents thus receive a request for non-NEMO Home Agents and will not set the R bit in the Reply message. Thus the MR is led into believing there is no HA on the home link supporting Mobile Routers. IPsec does not protect against this threat since the Home Agent Address Discovery Request is not protected neither by AH nor by ESP headers.

T.6 Threat on the *R* bit of Home Agent Discovery Reply: an attacker entity on the path between legitimate MR and HA transforms the R bit from 1 to 0. It is assumed that, initially, the MR has sent a Home Agent Address Discovery message to the home network with the R-bit set, thus requesting responses from HA's that support Mobile Routers; it is also assumed that the HA replied a legitimate Reply containing the R bit set. The effect of this threat is that MR is falsely led into believing that no HA on the home network can support Mobile Routers. IPsec does not protect against this threat since the Home Agent Address Discovery Reply is not protected neither by AH nor by ESP headers.

T.7 Threat of address spoofing: when attacker needs to send an unreasonably large amount of IP packets to a target without risk of its current address being identified, it could do so by two means. First, it would set the src address of the packets to another address, at random (thus "spoofing" a legitimate address, or "masquerading" as that address). However, the first-hop router might forbid forwarding packets whose source address are not topologically correct at that particular router (ingress filtering [49]). Second, if ingress filtering at the access router is in place, the MH might first encapsulate towards HA, thus tricking the access router; HA decapsulates and "bombs" the actual target by using MH's Home Address as source address. However, the ingress filtering technique is used at the HA as well; Mobile IPv6 requires HA of MH to only forward those packets from MH if the src address of the outer header to match a Care-of Address entry in the BC and the src address in the inner header to match the home address field of the same entry. The NEMO base specification offers further help by requiring the Home Address to match a Mobile Network Prefix owned by the Mobile Router. It is obvious that this threat applies to Mobile IPv6 for Mobile Hosts and, where Mobile IPv6 for Mobile Hosts offers protection, it automatically offers protection for Mobile Routers as well.

T.8 Threat on location privacy: In the context of Mobile Hosts (not Mobile Routers), location privacy represents the desire of a Mobile Host to not reveal, or hide, its current association Care-of Address (its location) – Home Address (its permanent identifier) from an attacker listening on the path between MH and HA. It is not a desire to hide only one address, but the association. It is sufficient for an attacker wishing to find the current location of a victim Mobile Host to snoop traffic between the victim and its Home Agent. When the Mobile Host changes its location and updates the Home Agent, a pair of Binding Update/Acknowledgement messages is communicated. An attacker on the path can find the association Home Address - Care-of Address of the Mobile Host, even if AH and/or ESP headers are used to protect the two packets. Both AH and ESP for Binding Updates and Acknowledgements are used in transport mode (not tunnel mode), thus the base header (containing the Care-of Address and the Home Agent address), the Destination Options header and the Routing Header Type 2 (containing the Home Address) are transmitted in clear (but message integrity, and implicitly integrity of the Home Address and the Care-of Address, is afforded by AH), see section 2.3.1.4.

In the context of NEMO, the location privacy can be described as the desire of a Local Fixed Node within a moving network to not reveal, or hide, the location triplet LFN Home Address - MR Care-of Address - MR Home Address. An attacker outside the moving network and on the path between the Mobile Router and its Home Agent could snoop packets. If the bidirectional tunnel between the Mobile Router and its Home Agent is not protected by ESP, then attacker can find the LFN Home Address in the src field of the inner packet sent by MR to HA and the MR Care-of Address in the src field of the outer base header. The MR Home Address could have been obtained from the Binding Update or the Binding Acknowledgement, as described in the previous paragraph. In this way, attacker can gain knowledge of the triplet LFN Home Address - MR Home Address - MR Care-of Address. However, if MR uses ESP tunnel mode protection for the bidirectional tunnel, then attacker has no means to gain visibility of the LFN Home Address.

Thus, even if location privacy might be considered as a security threat, it is mostly a risk for Mobile Hosts, and can not be qualified as a NEMO risk; the association Home Address - Care-of Address of a Mobile Host might be revealing location information but the location triplet can not be revealed if ESP is used for non-signalling traffic between MR and HA.

T.9 Threat on the Routing Header Type 2: attacker modifies the type of the routing header type 2 of a Binding Acknowledgement sent by HA to MR and substitutes 0 for 2. In addition attacker may specify a number of addresses within this fake type 0 routing header. The risk is that attacker provokes bombing attacks and stays hidden (its address does not appear in the packet); it is the Home Agent's and the Mobile Router's addresses that appear in the attack. The risk is typically a Mobile IPv6 for hosts risk, but it is more important in the case of Mobile Routers because Mobile Hosts are not expected to implement routing header software, or are expected to implement type 2 routing headers exclusively (for Mobile Hosts). However, all routers (Mobile Routers included) are expected to implement routing headers type 0, thus they are more at risk with this threat. Protection against this risk is again offered by AH which covers all fields of the Routing Header Type 2.

Interactions between several MR's of same HA

T.10 DoS threat on peer MR by attacker spoofing a legitimate MR's Care-of Address. A similar threat exists in the case of Mobile IPv6 for Mobile Hosts, but is less important than in the case of NEMO.

In the context of Mobile IPv6 for Mobile Hosts, consider two Mobile Hosts belonging to the same Home Agent; each MH is trusted by the Home Agent (with IPsec). The victim MH and the attacker MH are both visiting the same foreign network. The attacker MH reads the Care-of Address of the victim from the Binding Update or Acknowledgement that victim exchanges with HA. The attacker MH sends Binding Update for the victim's CoA and its own Home Address. Thus the HA will forward all traffic intended to attacker's Home Address towards victim's Care-of Address, even though IPsec is correctly being used.

This threat applies in the NEMO context as follows: consider a legitimate MR with prefix MNP and an attacker MR with a different prefix, both served by the same HA. Each MR shares a set of keys with HA. The attacker MR could instruct the HA to add MNP in the binding cache, relating it to its own Home Address (instead of to the legitimate MR's Home Address), thus effectively denying service to the legitimate MR and redirecting the entire traffic to MNP towards the attacker MR. Even if HA uses IPsec, it could not protect against attacker MR's claiming the legitimate MR's MNP. However, the prefix table specified by NEMO base protocol associates a MR's Home Address to the MNP that it owns, thus constituting a means for MR to check against attacker MR claiming a prefix it does not actually own.

Nested Mobility

T.11 DoS threat on TLMR due to too many levels of nested networks: several moving networks may attach one under the other thus forming a nested aggregation of moving networks ("levels" can be pictured as follows: first MR attached under TLMR makes it for a one-level aggregation of moving networks; a second MR attached under TLMR is still a one-level aggregation; were the second MR attached under the first MR, it would have been a two-level aggregation).

Naturally, the top-level MR will forward traffic of all moving networks attached under it. When the number of levels is large, this may become an overload on the original expectations of the capabilities of this Mobile Router (overload in the form of more cycles dedicated to IPv6 Fragmentation and Reassembly, as well as Path MTU calculations), thus becoming a DoS attack.

It is thus possible for MR to need to limit the number of levels of moving networks nesting under it; it could use the Tunnel Encapsulation option by setting a limit on the number of levels of mobile networks below it.

Nested mobility configurations appear also when Mobile Hosts visit mobile networks. However, all Mobile Hosts will always attach to a same level; given a mobile network, it is not possible to build a more than one-level nested aggregation only by adding Mobile Hosts (MH's don't attach one under the other). Thus, the above mentioned threat of nested configurations is pertinent to nested moving networks exclusively.

2.3.1.3 IPsec Architecture for Nested Mobility

The IPsec architecture can be particularized for nested mobility cases by using nested encapsulation. In the figure below we picture the protection of NEMO signalling between MR1 and its HA (HA_MR1), when the moving network of MR1 is nesting within the moving network of MR2 (it is assumed that the MR2 has already performed NEMO signalling with its own HA - HA_MR2). The first level of IPsec protection is offered by the AH/ESP transport mode between

MR1 and HA_MR1 (1). The second level is offered by AH/ESP tunnel mode between MR2 and HA_MR2 (2) in Figure 11

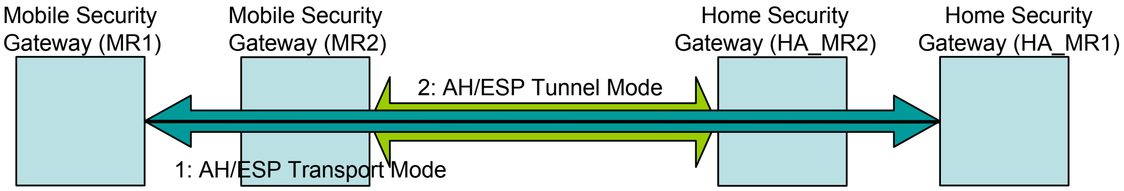

Mobile Security Gateway (MR1)
Mobile Security Gateway (MR2)
Home Security Gateway (HA_MR2)
Home Security Gateway (HA_MR1)
2: AH/ESP Tunnel Mode
1: AH/ESP Transport Mode

Figure 11: IPsec protection of MR1-HA_MR1 signalling (nested)

The IPsec protection of application-level traffic between LFN and CN, when LFN belongs to a nested moving network is pictured below. The first level of protection is offered by the AH/ESP tunnel mode between MR1 and HA_MR1 (1) while the second is offered by the AH/ESP tunnel mode between MR2 and HA_MR2 (2) in Figure 12

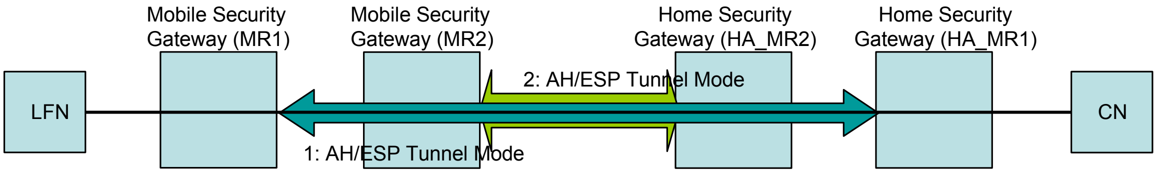

Mobile Security Gateway (MR1)
Mobile Security Gateway (MR2)
Home Security Gateway (HA_MR2)
Home Security Gateway (HA_MR1)
LFN
CN
2: AH/ESP Tunnel Mode
1: AH/ESP Tunnel Mode

Figure 12: IPsec protection of LFN-CN application-level data (nested)

A particular case of nested mobility configuration is the visit of a MH within a moving network. The signalling protection is offered by AH/ESP in transport mode between MH and HA_MH (1) and by AH/ESP offered by AH/ESP in tunnel mode between MR and HA_MR (2) in figure 13:

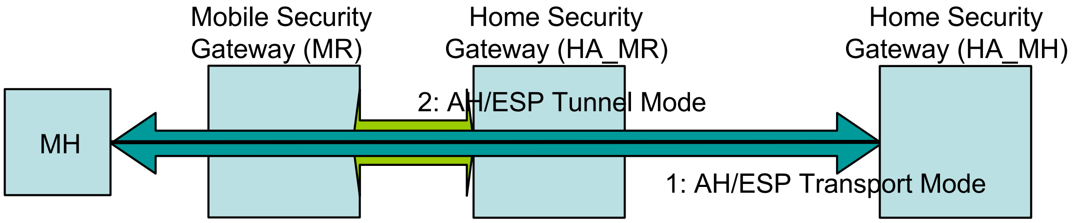

Mobile Security Gateway (MR)
Home Security Gateway (HA_MR)
Home Security Gateway (HA_MH)
MH
2: AH/ESP Tunnel Mode
1: AH/ESP Transport Mode

Figure 13: IPsec protection for MH signalling visiting moving network (nested)

Still in the case of nested mobility of a MH within a moving network, the application-level traffic between MH and CN is offered a first level of protection by the AH/ESP tunnel mode between MH and HA_MN (1) and a second level by the AH/ESP tunnel mode between MR and HA_MR (2):

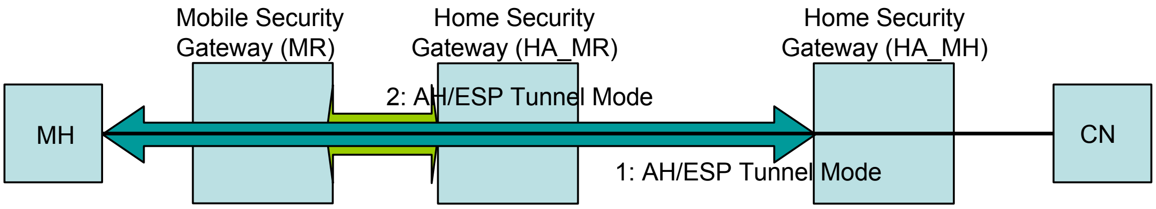

Mobile Security Gateway (MR)
Home Security Gateway (HA_MR)
Home Security Gateway (HA_MH)
MH
CN
2: AH/ESP Tunnel Mode
1: AH/ESP Tunnel Mode

Figure 14: IPsec protection of application-level data MH-CN (nested)

2.3.1.4 IPsec Protection of Binding Updates and Acknowledgements

In this section, we used the following NEMO messages for threat analysis: Binding Update, Binding Acknowledgement. The following fields have been considered as relevant for NEMO threat analysis:

- source (src) and destination (dst) addresses in the base header.
- the Home Address in the Destination Options 0 header of the Binding Update.
- the R bit in the AHLKR field of the Binding Update.
- the Prefix Len and Mobile Network Prefix fields in a Binding Update sent in explicit mode.
- the Routing Type, Segments Left and Home Address fields in the Routing Header Type 2 of the Binding Acknowledgement.
- the Status field of the Binding Acknowledgement.

In building the packet formats below, the following notation was used:

- *: NEMO field, or bit or value introduced by NEMO base protocol, or containing helpful information for realization of the NEMO-related risks described in this document.
- x: authenticated field, as covered by AH ICV.
- y: encrypted field, as part of ESP payload data.
- z: authenticated field, as part of ESP authentication data.

For example, fields that are marked (*xyz) are helping realizing threats, but are protected by AH and ESP non-NULL authentication, thus rendering most NEMO threats impossible; fields that are only marked (*) are not protected, thus might constitute security risks.

In section Annex A pairs consisting of a Binding Update and the corresponding Binding Acknowledgement are illustrated. Each section describes two pairs: the pair when MR is in a foreign network followed by the pair when MR is returning to the home network. Section 2.3.1.5 presents unprotected pairs while section 2.3.1.4 (actually the second half of that section) presents pairs protected both by AH and ESP in transport mode (ESP with non-NULL authentication algorithm). Intermediary sections use transport mode AH exclusively or transport mode ESP exclusively (ESP with or without authentication algorithm).

IPsec non-Protection of Home Agent Discovery Messaging

In this section, we used the following NEMO messages for threat analysis: Home Agent Address Discovery and Home Agent Address Reply. The following fields have been considered as relevant for NEMO threat analysis:

- src and dst addresses in the base header.
- the R bit in the ICMPv6 discovery and reply messages.
- the R bit in the Home Agent Information Option of the Reply message.

In building the packet formats below, the following notation was used:

- *: NEMO field, or bit or value introduced by NEMO base protocol, or containing helpful information for realization of the NEMO-related risks described in this document.

- x: authenticated field, as covered by AH ICV.
- y: encrypted field, as part of ESP payload data.
- z: authenticated field, as part of ESP authentication data.

For example, fields that are marked (*xyz) are helping realizing threats, but are protected by AH and ESP non-NULL authentication, thus rendering most NEMO threats impossible; fields that are only marked (*) are not protected, thus might constitute security risks.

2.3.1.5 Unprotected Home Agent Address Discovery and Reply

```
Base Header                        Base Header
  src: CoA                   (*)     src: Home Agent address    (*)
  dst: Home Agents anycast   (*)     dst: CoA                   (*)
ICMPv6 message                    ICMPv6 message
  Type                              Type
  Code                              Code
  Checksum                          Checksum
  Identifier                        Identifier
  R                          (*)     R                          (*)
                                     Home Agent Information Option
                                       Type
                                       Length
                                       R                        (*)
```

Remark the the Agent Discovery messages are not protected at all by IPsec and may lead to important security threats. NEMO threat analysis is concerned solely by the use of the R bit of these messages, and by the risks involved on tampering with the integrity or the confidentiality of this bit exclusively.

3 Functional Description of Dynamic IVAN Management

3.1 Introduction

The Dynamic IVAN Management tries to tackle complementary issues for OverDRiVE moving networks. This chapter contains two issues, which emerged from the conceptual work. The first section presents simulation results in the scope of route optimization. The main topic is the influence of route optimization of transport layer communication. The next section comprises a reflection the traffic management concept presented in D07. The subsequent refinement integrates bandwidth measurement into the concept. The integration of bandwidth measurements is based on a detailed description, which lies out the fundaments for the implementation and validation described in chapter 4.

3.2 Effects of Route Optimization

More and more attention is given to the aspect of optimization in the scope of being mobile by the means of MIPv6 [13]. In following fields optimizations are carried out:

- choosing a more appropriate network path (e.g. route optimization),
- minimize handover determined packet losses (e.g. fast handover) and
- micro-mobility management (e.g. hierarchical handover).

These topics are introduced because MIP's (both MIPv4 and MIPv6) basic idea is to tunnel packets between the home network and a mobile node while latter is not at home. The usage of the home agent for tunnel establishment and actual packet encapsulation in the home network allows for many mobility scenarios and paves the way for a migration path, i.e. an integration of mobility in the Internet routing fabric which is transparent to layers above IP.

However, topics mentioned above are not needless because there is an inherent overhead in tunnelling which causes packet delays, drops and – to some extend – degradation of network performance. While the basic support for route optimization is part of [13], the IETF working groups MIPSHOP [14] and NEMO [15] address additional optimizations for host and network mobility.

OverDRiVE's interest in this topic is clearly from the network mobility point of view which is also intensely regarded in the NEMO working group. While section 2.2 thoroughly addresses the aspect of optimization for the mobile router, this section takes a look on optimizations from the Dynamic IVAN Management perspective. Network Access Control (cf. [2], section 3.4) already presents one optimization in this field and to gain a deeper insight into and implications of optimization methods, this section presents a closer look on a dedicated route optimization method for OverDRiVE mobile networks.

An analysis of route optimization methods can be carried out by different approaches. Here, simulation is the main method used to inspect and evaluate route optimization for mobile networks. The treatment was based on simulation as measurements could difficultly be conducted with mobile networks comprised of several nodes.

The following sections present the evaluation of route optimization in the scope of mobile networks. In the following the description of the two types of Mobile Routers, the simulation model, and the scenario and traffic model are laid out. Afterwards, simulation results are described and a conclusion is given.

3.2.1 Which network should a node attach to?

When a node connects to a foreign link, a care-of address has to be generated. A prefix of this address can be taken from the Router Advertisements (RAs) broadcast on the link or the address can be assigned by a DCHP server. If the node is a Mobile Router (MR) at least two approaches can be used to generate addresses on the ingress interfaces. These approaches are the foundation of the conducted simulation and described in the following. As all Correspondent Nodes (CNs) within the simulation support Route Optimization (RO) (cf. [13]), some signalling details are also described.

3.2.1.1 MobileRouter with Encapsulation

If a node is a Mobile Router (MR) and basic network mobility ([19]) is used, the egress interface – which connects the MR to the Internet – of the MR is assigned a global address which is valid at the foreign link and for example derived from RA of the Access Router (AR). Addresses at the ingress interfaces of the MR are valid at the home network of the MR, i.e. the MR uses its home address in the "inner" network, when sending RAs. Local Fixed Nodes (LFNs) and Mobile Nodes (MNs) beneath the MR can use the prefix from the RAs and their MAC address to generate "plain" IPv6 or care-of addresses. In case of MNs, the address is reported to its HA. After the tunnel is established between the MR and its Home Agent (HA), all packets from nodes beneath the MR and destined towards the Internet are encapsulated and routed via the MR's HA, where they are decapsulated. This constellation is depicted in on the left side in Figure 15.

The first packet from a CN to a MN beneath a MR is routed to MN's HA, encapsulated and forwarded to the MR's HA, where it is encapsulated the second time and forwarded to the MR's CoA. Via the Access Router (AR) the packet arrives at the MR. Here, it is decapsulated once. The inner header advises the MR to send the packet to the MN, so the MN receives the packet, decapsulates it again and finally has the data. The overhead caused by each encapsulation using the standard encapsulation [68] is 40 bytes per packet.

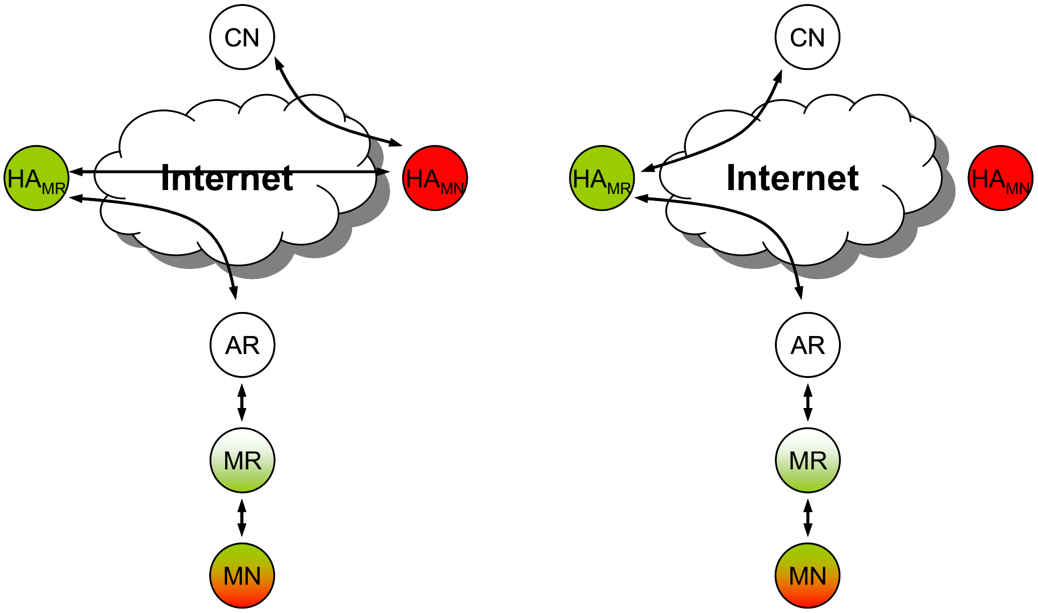


Figure 15: Basic MR and MN without RO (left) and MN with RO (right)

In order to perform RO, a MN sends a BU to the CN. Subsequently, future packets are not routed through the MN’s HA. However, they still have to pass the encapsulation step between MR and its HA. This is shown on the right side in Figure 15.

If the MR looses contact to its AR and connects to a different AR. All nodes beneath the MR are not aware of the mobility. They might only notice that packets are lost “somewhere in the network”. Addresses on MR’s ingress interfaces as well as addresses of LFNs and MNs remain unchanged – so there is no reason for MNs to send additional BUs.

3.2.1.2 Mobile Router with Route Optimization

The MR supporting RO for MNs inside its network broadcasts a subnet-prefix of the AR in the RAs, so MNs generate a CoAs valid at the AR, i.e. packet arriving at the AR are routed via the MR to the MN. As in the basic case, a MN sends a BU to its HA with the information about its CoA but the latter is derived from the address space of the AR.

When a CN sends its first packet to a MN, this packet is delivered to the MN’s HA where it is encapsulated and forwarded to the AR, then via the MR to the MN, where the packet is decapsulated. This packet flow is depicted on the left side in Figure 16

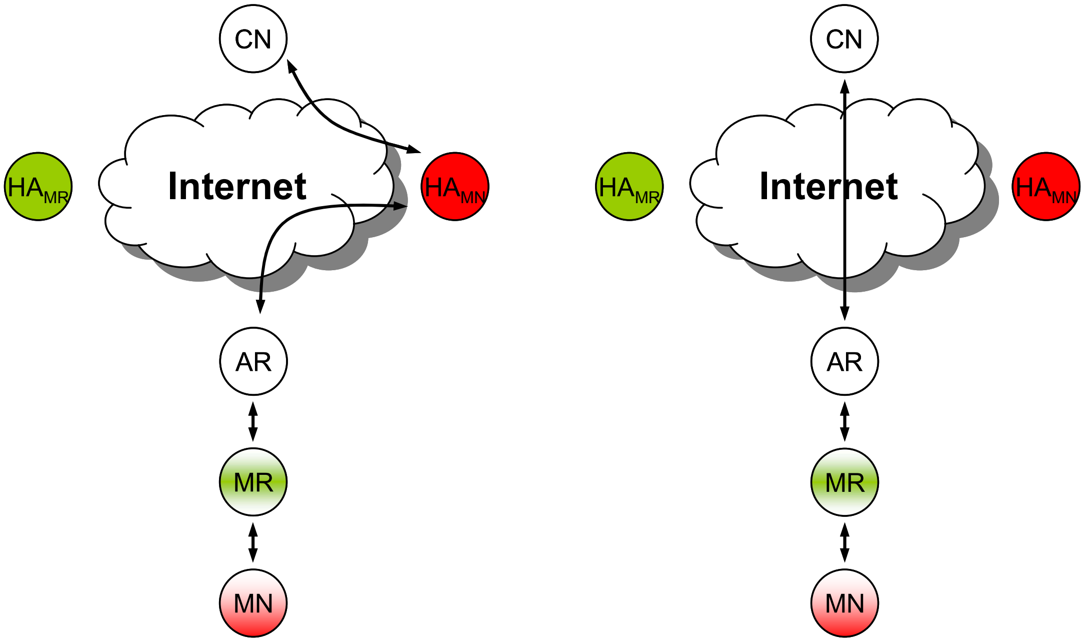


Figure 16: Alternative MR and MN without RO (left) and MN with RO (right)

When a MN sends a BU to a CN, future packets are neither routed via the MN’s HA nor via the HA of the MR. It is forwarded to the AR and does not get en- or decapsulated by HAs. This scenario is shown on the right side of Figure 16.

When the mobile router switches from one AR to another, the MNs are able to notice that they left the domain of the AR – i.e. the router has disappeared. They receive a new router advertisement from the MR containing a sub-prefix of the new AR and subsequently send BUs to their HAs and CNs to register the new point of attachment.

If the alternative MR is used, LFNs can be treated in a similar way, i.e. they reconfigure their IPv6 address according to the MNs, or they keep an address which is valid with respect to the home network. Detailed modifications of RAs are not further discussed here, as LFNs are not in the scope of the simulation.

3.2.2 Scenarios and Traffic

The first version of MR utilizes the basic mobility support used in the context of OverDRiVE networks and the NEMO working group. The alternative approach describes one possible approach in the field of route optimization for mobile networks. While the second approach benefits from avoiding multi-angular routing, it might be not a solution for the general case, e.g. nested networks are only embraced in dependence of the sub-prefix advertised by the AR.

The simulation performed in this scope of RO for networks does not meant to describe a thoroughly specified protocol; it is based on the idea that multi-angular routing can only be turned down if additional signalling is used. Here, we rely on the RO method already specified for MIPv6. When a MN beneath a MR should use this kind of optimization, the mobility of the MR must be visible. In the following scenario this approach allows for data flows, which bypass HAs as soon as possible. But on the other hand, BUs have to be send to all nodes that communicate with MNs beneath a MR.

In the situation of a handover of a MR the CoA of all MN change at the same time, so all nodes need to send a BU to their HAs and CNs at once. A large number of BUs sent by MNs within a short time frame is called a B*inding Update storm* (BU storm). It may be a challenging in the following ways:

- The large number of BUs sent may cause other packets on the network to be lost or delayed. It can be assumed that the user of a wireless network wants to use applications that rely on the network for sending or receiving TCP- or UDP-packets. Lost or delayed packets may (temporarily) prevent those applications from working as expected.
- During the BU storm, some binding updates may be lost. If this happens to the same client more than once, the home agent or correspondent node may assume that the connection is down.
- If the BU intervals are identical with all clients, they also send future binding updates synchronized.

Although the presented RO method for mobile networks can shorten the path between MNs within the network and CNs within the Internet, the effect of a BU storm might introduce a drawback. Therefore, the effects of BU storms are the central point investigated in the simulations.

The effects are shown as an evaluation between the basic MR and the alternative MR. The results of the simulation are not intended to favor one type of MR. The aim is merely to demonstrate the impact of a BU storm.

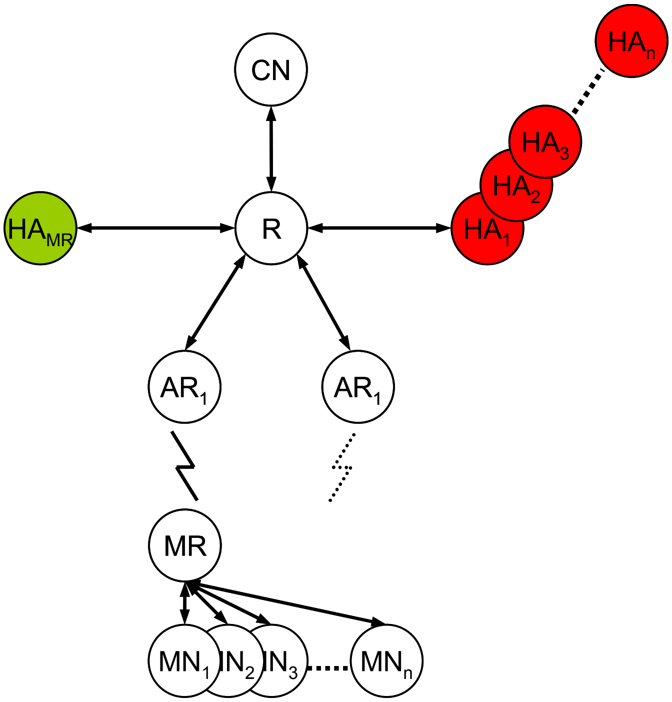


Figure 17: Simulation Scenario

Basics of the network configuration are depicted in Figure 17. The underlying scenario is a bus (MR) transporting a set of users with mobile devices (MNs) and repeatedly changing the Access Routers (AR_1, AR_2). In each simulation bottleneck link is between the MR and the ARs.

3.2.3 Simulation Model

The NS network simulation [16] tool is used for the simulation. NS is a packet level, discrete event-driven simulator targeting simulations in the field of networking research. NS provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless networks. Additionally, the mobility extension MobiWan [17] is used as basis for the implementation of both MRs. MobiWan is a NS extensions to study mobility in wide-area IPv6 networks and provides MIPv6-like extensions of ns.

As simulation reflects only parts of the real world, some remarks are following, which describe some important aspects of the simulation model:

- One difference between our simulation and the real world is that generally CNs are not obliged to support RO. With a server on the Internet that communicates with hundreds or thousands of clients at once, it would be a major task to maintain a list of each client's care-of address. The CN in the simulation does support RO, i.e. the HA of the MNs will be omitted after the RO messages are exchanged.
- In real life the lifetime of binding updates is set to 420 seconds. During this period of time the binding update is refreshed 3 times, if no packet gets lost. In the simulations the lifetime of a BU is set to 10 seconds and a MN refreshes its CoA binding using a rate of 1 BU each 3 seconds. Additionally, when a MN's CoA changes in the simulation, the node sends 5 BUs to its HA using a rate of 1 BU per second.
- The wireless network uses a simple MAC layer without collisions at a bandwidth of 128 kbit/s. When looking at link layer protocols used in Bluetooth or GSM systems it does

not seem too unlikely that in a scenario like ours MIPv6 may not have to deal with lost packets due to collisions. However, it should be noted that the most common MAC layer for our situation would be IEEE 802.11 (WLAN) which was not used here, mostly because of issues with the 802.11b MAC layer for the NS version 2.1b6a. The simple MAC layer makes it relatively easy to find the reasons for many of the effects encountered. But it also causes some of those effects, because it allows packets from different nodes to be transferred parallel. When node A and B simultaneously send packets to node C, both can arrive at the same time. Node C receives the packets with a data rate which is twice as high as the selected bandwidth of the MAC layer. Of course this gets worse when even more clients send their packets at the same time. On the other hand, when node C is a router (like in our case) which has to process and forward the packets and only has a limited bandwidth to the external network, packet loss in the router's sending queue compensates for the behaviour of the MAC layer.

3.2.4 Simulation Results

The simulation results are presented in three sections. The first two sections present results of one MN that is beneath a MR. Their main focus is the influence of the mobile scenario on the transport protocols UDP and TCP.

The third subsection inspects a number of parameters in combination with TCP. Parameters of interest are

- The number of MNs beneath the MR. The main issues is the sending of additional BUs and Binding Acknowledgments (BACKs) which need to pass the MR-AR bottleneck link. The BACKs are not actually needed here but they are still sent by MobiWan's HAs. Additional nodes do not send or receive TCP- or Constant Bit Rate (CBR) traffic to keep the simulation runs comparable.
- The length of the queues. Specially, the sending queue of the MR should be monitored, since it has to deal with the BU storm.
- The number of handovers performed.
- Different ways of handling BUs. BUs could be sent in randomized intervals to avoid the BU storm. BUs may be preferred in queues to decrease the probability of a BU being lost.

The traffic always flows from the CN to the MN_1. No other mobile node sends or receives any data except for the usual MIP packets like router advertisements and binding updates. This is important since the series simulations, each of which keeps all parameters but one constant, should be comparable in the end. The size of the TCP- and UDP-packets is always 1000 Byte.

3.2.4.1 Constant Bit Rate Traffic

In this scenario only one MN is “inside” the network of the MR. The traffic starts after t=10 seconds of simulation time. In this case the traffic is a simple CBR stream from the CN to the MN. At t=20 seconds the MR starts to move from AR_1 to AR_2. Every 10 seconds the MR changes its AR. At t=110 seconds the last handover is performed and the MR does not moved any more.

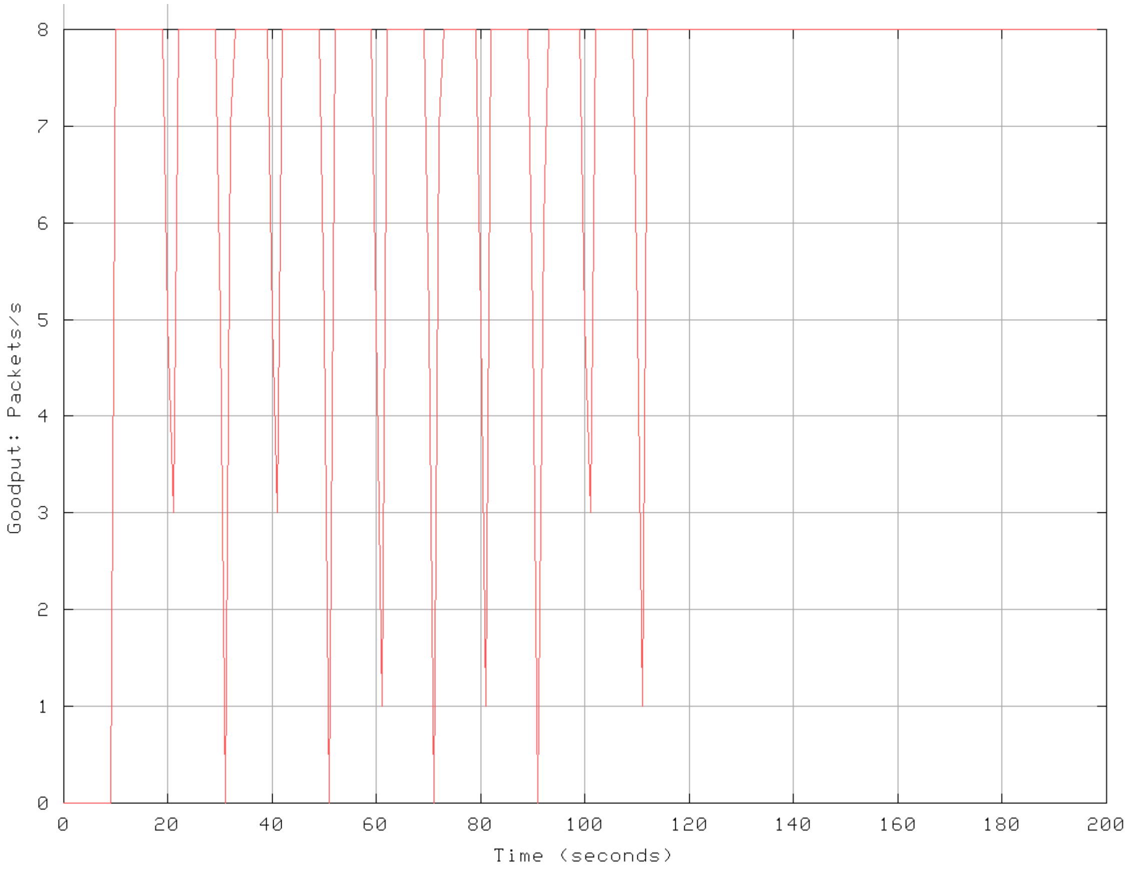


Figure 18: CBR throughput in a Mobile Environment

Figure 18 shows the throughput of the CBR stream at the MN. The throughput is identical for both types of MRs. After 10 seconds the CBR stream flows from the CN to the MN at the maximum sending rate of 8 packets per second. When the MR leaves the radio reception area of AR_1 and enters the area AR_2, the stream encounters packet drops. After each location update the stream continues. The slight differences between of the gaps are negligible as they are caused by the averaging time interval.

Taking a look at the delay differences encapsulation take about 4 ms longer when using the alternative MR , which is the time packets need to get from the core router (node R, Figure 17) to MR's HA and back to the core router. It turns out that the simulation parameters are fortunate for encapsulation. In the wired network packets from the CN to an AR take 22 ms, the long way via the MR's HA only adds another 4 ms. If this was different, the MR with encapsulation may have performed worse in the TCP simulations.

Both encapsulation and route optimization start with a long delay which is reduced by 4 ms about 3 seconds after the traffic started. This is because of the BU sent to the CN, so the traffic no longer has to be routed via the MN's HA. Remember that in these first 3 seconds in the case with encapsulation the data was actually encapsulated twice, the CN sent it to the MN's HA where it was encapsulated once; from there it was routed to the MR's HA which encapsulated each packet again.

In this simple scenario both MRs behave very predictable. At first, the differences can easily be seen:

1. The delay between the CN and the MN using the alternative MR is reduced by twice the delay between the core router R and the HA of the MR.
2. The used bandwidth (not further shown) is increased the basic MR is used. This is due to the introduced encapsulation overhead.

Depending on the used architecture (e.g. indirection caused by the HA traversal) and the relative header overhead (e.g. a 100 byte real-time traffic packet form the CN results in an encapsulation overhead of 40%), the use alternative MR can be preferred as one MN inside the mobile network does not cause a BU storm effect.

3.2.4.2 TCP and Handovers

In the second scenario the CBR stream is replaced by a saturated TCP stream, e.g. a FTP file transfer is used to describe the effect of a handover. The goodput is measured for TCP-Reno and the stream originates from the CN to the MN.

The traffic starts after *t=10* seconds of simulation time. Starting at *t=20* seconds, one can see the handovers occurring each *10* seconds. *t=110* seconds is the time of the last handover, after that the nodes are not moved any more.

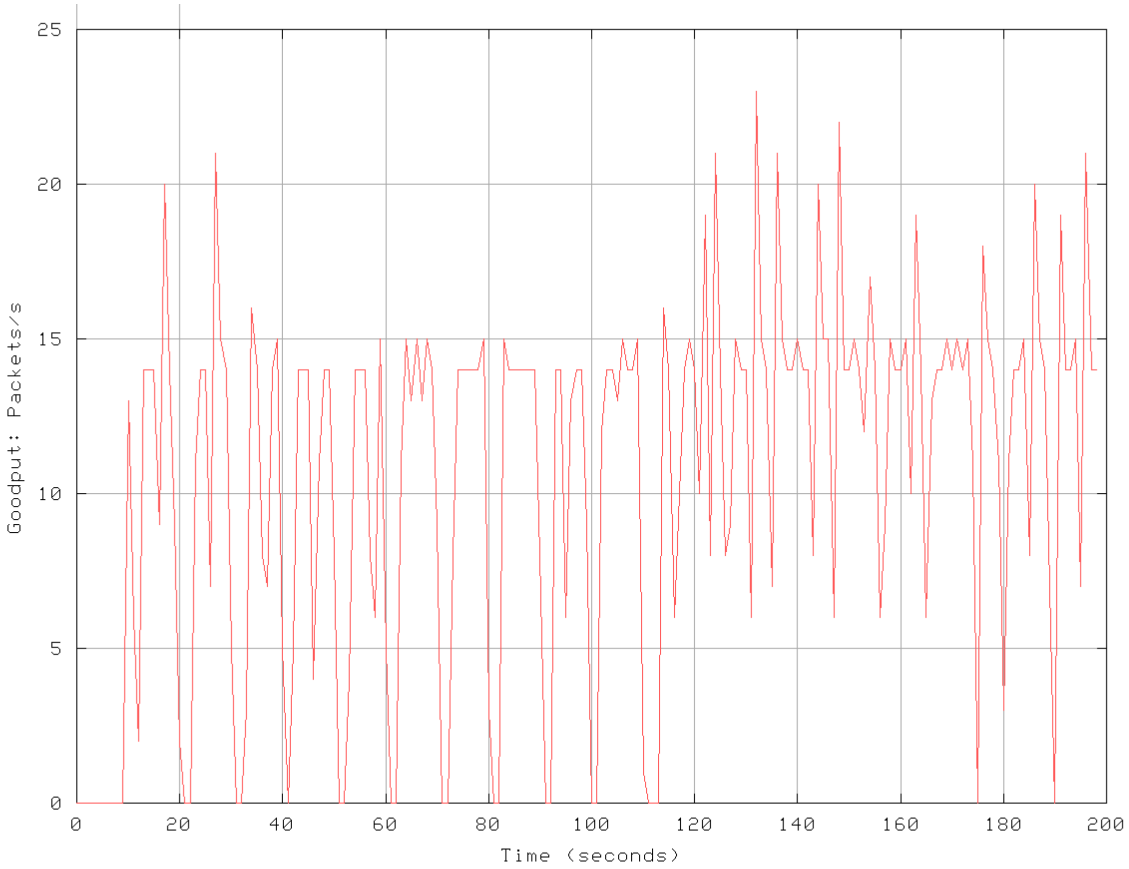


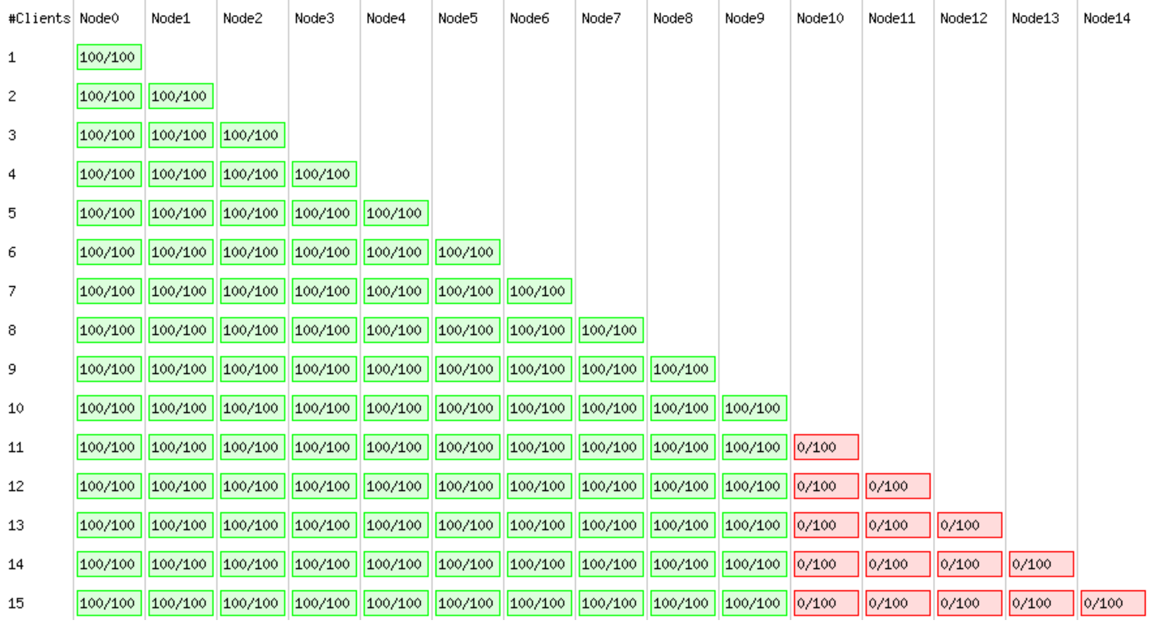
Figure 19: Handovers and TCP Goodput

Taking a look at the time between 0 and 120 seconds, Figure 19 gives an example of a TCP-Reno stream affected by several handovers. After the traffic has been started, the average number of packets received is 11.1 per second which corresponds to 88.8 kbit/s. Again the bandwidth of the channels between the access routers and the mobile router and between the mobile router and its clients is 128 kbit/s. Every 10 seconds the handover causes the TCP stream to fall into a timeout. Even if the handover delay is shorter than the Retransmission Timeout, the handover from one AR to another causes a burst of packets to be dropped. In case of the MR using encapsulation all packets which left the HA of the MR will arrive at the old AR and in case of the alternative MR all packets already sent by the CN can not reach the new location of the MN. If the amount of packets lost during the handover (including the time to update the location) can not trigger the TCP sender to recover from prior losses, the TCP-Reno stream will timeout and starts over with a slow start. This degradation of utilization happens in both cases.

Again, depending on the network architecture one type of MR can be preferred. Here, a theoretical scenario can favour the MR with encapsulation, as the time between handover and the new BU might allow for a faster rerouting of arriving packets. This assumes a relatively long distance by means of RTT or bandwidth-delay product between the HA and CN.

3.2.4.3 Increasing number of nodes

The main objective of this section is to show the effects of an increasing number of MNs, which are located beneath the MR. Here, the uplink queue of the MR is of interest. Figure 20 summarises typical simulation results. Each row describes one simulation run and the columns show the number of BUs send from a MN and received at the corresponding HA. The "0/100" in row 11, column "Node10" indicates that MN_{10} sent 100 BUs in the simulation run with 11 MNs and any BU arrives at its HA. Note that this type of diagram only counts BUs sent after t=10 seconds and only BUs from the MNs to their corresponding HA. Neither BUs from the MR nor BUs to the CN are shown. Since the MN_i, i>9 never manage to get a BU to their HA, they can not be reached from the wired network.



#Clients	Node0	Node1	Node2	Node3	Node4	Node5	Node6	Node7	Node8	Node9	Node10	Node11	Node12	Node13	Node14
1	100/100														
2	100/100	100/100													
3	100/100	100/100	100/100												
4	100/100	100/100	100/100	100/100											
5	100/100	100/100	100/100	100/100	100/100										
6	100/100	100/100	100/100	100/100	100/100	100/100									
7	100/100	100/100	100/100	100/100	100/100	100/100	100/100								
8	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100							
9	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100						
10	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100					
11	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	0/100				
12	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	0/100	0/100			
13	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	0/100	0/100	0/100		
14	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	0/100	0/100	0/100	0/100	
15	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	100/100	0/100	0/100	0/100	0/100	0/100

Figure 20: Loosing Binding Updates

This behaviour is determined by the queue length of the MR's uplink interface. According to Figure 20, the queue length is set to 10 packets and the queuing strategy is drop-tail, i.e. new packets that arrive at the filled queue are dropped.

The packet drop in the simulation scenario must be handled carefully as the simulation model uses a simple wireless MAC-layer (cf. 3.2.3) and all nodes are created virtually at the same time.

Firstly, this causes all MNs to start their registration process at the same time. Secondly, all BUs originated from the MNs arrive at the MR in a very narrow time interval, i.e. virtually at the same time. One the one hand this is a drawback of the simulation scenario but on the other hand the BU storm effect stands out when the ingress/egress bandwidth ratio is low and the uplink buffers is small. It is also worth mentioning that the uplink queue is the simulation scenarios only holds mobility messages and TCP feedback to the CN, i.e. BUs only have to compete with ACKs from a single TCP stream.

Two modifications are described in the following subsection to cope with the BU storm effect.

Longer Queues

The MR drops BUs in the uplink queues. A simple approach to solve this would be to make the queues longer. In the following simulation runs all sending queues – including the ones of the mobile router – are 65 packets long while the number of nodes that send BUs is less or equal than 60. Hence, no BUs should be dropped.

In Figure 21 the cumulative goodput of the TCP connection from the CN to one MN is shown. While the alternative MR is slightly better when the number of node beneath the MR is small, both trajectories meet at about 40 nodes. The BU storms still cause the TCP goodput to decrease significantly as the number of MNs increases. Sending 60 binding updates takes the MAC layer of the MR about half a second, during this time TCP ACKs have to wait, i.e. the TCP sender lacks feedback.

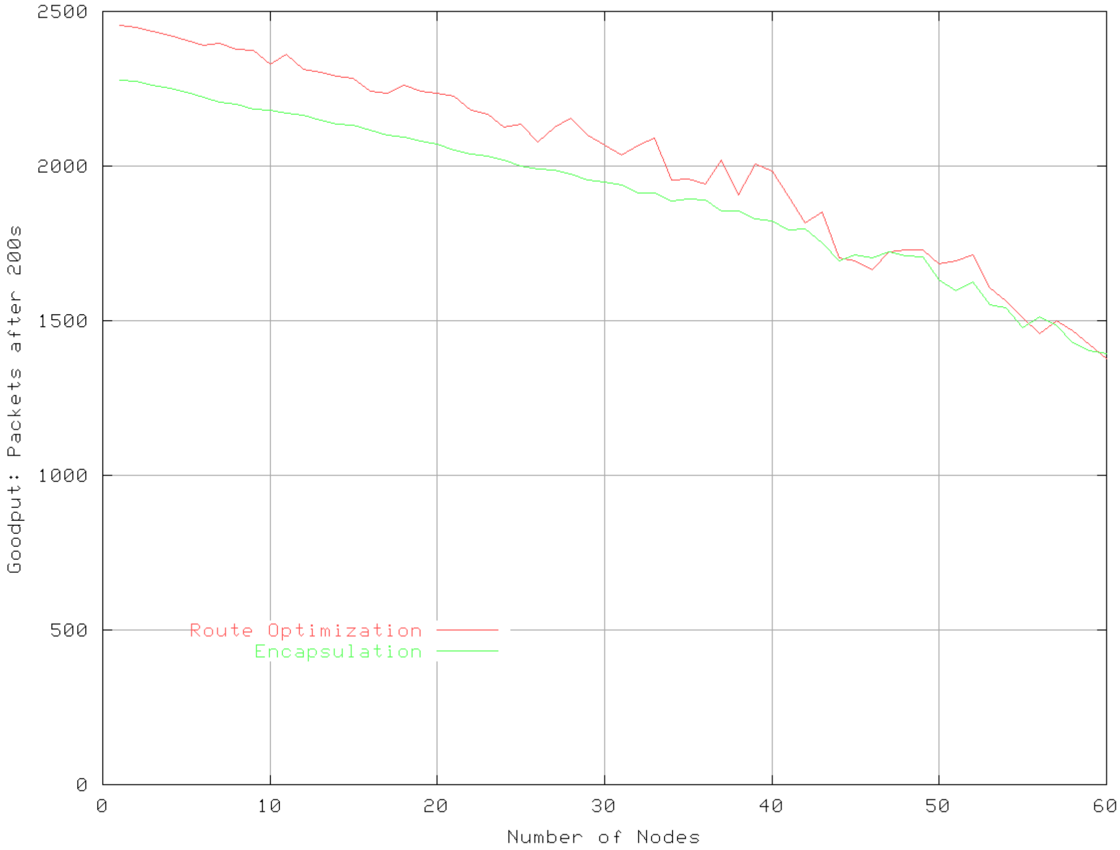


Figure 21: Increasing Numbers of Mobile Nodes

In real operating systems like Linux the default queue length is typically 100, so the queues are even longer than in this example. But if there are 50 MNs connected to the MR and each node has 10 CNs, the number of BUs in one binding update storm can already be as high as 550, 500 BUs from the mobile clients to the correspondent nodes and 50 BUs from the MNs to their HAs. Peer-to-peer applications benefit from a huge amount of peer, they can use up to 500 connections to correspondent nodes. This means that either an extremely long queue at the MR or some other solution is needed.

Randomizing Binding Updates

In this series of simulations the interval between the BUs is randomized, it is uniformly distributed on $\{0.5t,\ldots,1.5t\}$ where *t* is the interval that would be used without randomization.

For the simulations this means that a client sends its BUs

- immediately after the CoA changes (there is no randomized delay before this first BU),
- 4 more times after the change of the CoA, with the time between two binding updates being distributed on {0.5 seconds, ..., 1.5 seconds},
- in all other situations the interval is distributed on {1.5 seconds, ..., 4.5 seconds}.

So there still is one BU storm immediately after a handover when using route optimization. In all other situations there are no more BU storms. In these simulations only very few BUs are lost even when using more than 60 clients and with the length of the MR's sending queue being only 10.

This scheme of randomization could be justified like this: When a node enters a new network it does not know if it has to expect problems with BU storms. So it sends its first BU immediately to archive the optimal result in case that there is no problem at all – the connection is re-established as quickly as possible. If there is a problem, this BU will probably get lost. But further BUs are randomized anyway, so there is a good chance that the next one can reach its destination.

Of course one could think of better ways to handle this. A node may try to randomize the first binding update "just a bit" to avoid binding update storms. With MAC layers like WLAN the "collision avoidance" may even randomize the binding updates enough to archive this in some situations. If there still is a problem, a node could use information from layer 2 or from the number of lost BACKs to determine if there are BU storms. It could remember what happened the last time that it entered this particular network and act accordingly. A node that has data to send or that expects data from the Internet to arrive could be given a higher priority and sends its BU earlier than other nodes.

The simulations of the basic and alternative MR start under comparable conditions. In both cases all nodes in the mobile network are "switched on" at the same time. This even causes a BU synchronization – hence, a BU storm – in both cases. Each MN in the encapsulation scenario only sends BUs to renew the lifetime, while in the alternative case a node also needs to update the binding after a handover of the MR.

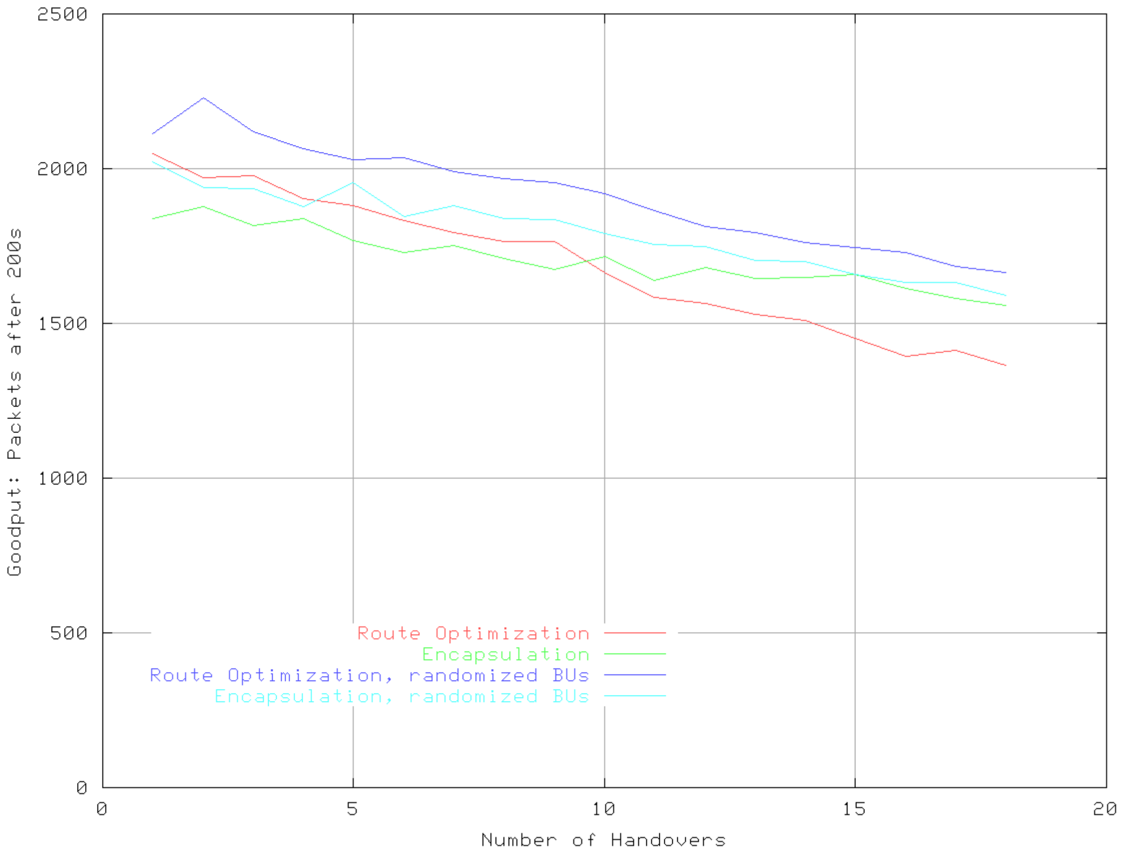


Figure 22: Encapsulation and Route Optimization combined with randomization

Figure 22 shows the behaviour of the different types of MRs when increasing the number of handovers. As one can see, the alternative MR with route optimization performs better than the version with encapsulation when there are only few handovers. However, with more than 10 handovers the basic MR with encapsulation gains an advantage, the fewer BU storms make it scale very well.

Randomizing the BUs increases the performance in both cases and specially improves the performance of the alternative MR with route optimization compared to encapsulation type. Comparing to Figure 21, the goodput is lower in these simulations than in the runs with queue length 65, since packets are lost from time to time. However, randomization eases the BU storm effect; all nodes can send their BUs.

With increasing numbers of handovers the number of BU storms also increases for the MR with route optimization. So the goodput is not only reduced by packets being lost during the handover, but also by TCP packets and ACKs that are dropped in the BU storm.

The simulations were only performed within a certain range of handovers, as can be seen in Figure 22. In case of using route optimization without randomization, the costs for sending BUs for each MN inside the network exceed the gain of an optimal route. If the numbers of handovers are further increased, this will also happen in case of route optimization with randomization of BUs. The load of the BU storm is only smoothed but not removed. An open issue for further studies is to find the break even between encapsulation and different types of route optimization regarding different real life scenarios.

3.2.5 Conclusion

The main conclusion is that randomizing BUs can ease the BU storm effect, so it is strongly recommended. However, immediately after a handover each MN needs to send a BU to its HA in order to receive packets at the new point of attachment. In this situation there will always be a larger number of BUs, which have to be processed in a relatively short time.

The main objective of this section is simulation based analysis of the BU storm. The effect is shown by an evaluation using a MR with encapsulation and a MR with route optimization in a basic scenario. In many situations route optimization is advantageous, it produces less overhead and allows sending packets with shorter delay and less packet overhead. However, route optimization causes nodes to send additional BUs after each handover. Depending on the scenario and configuration a BU storm might affect

- transport layer flows negatively as a burst of packets is injected in the network and causes transient congestion at a bottleneck link, and
- can cause mobility messages to be dropped, which subsequently interrupts flows until the first BU reaches the HA or CN.

A direction to continue this analysis is the refinement of the simulation model. A more realistic MAC layer like IEEE 802.11 should be investigated in detail. This can introduce a level of randomization for the BUs, but needs a transition to a newer version of ns and possibly a reimplementation of Mobiwan and the extensions to support both types of MRs. Furthermore, BACKs are sent in the simulations, but ignored by the MNs. Therefore, in the presented simulations the only effect of a BACK is the competition for bandwidth between them and the TCP or UDP packets from the CN. If the MN's decision when to send a BU would depend on the BACK received from the HA, the nodes are able to run out of sync.

3.3 Traffic Management

Based on the requirements of D03 [1] traffic management approaches for moving networks were discussed in D07 [2]. The MR and its HA were identified as entities that should do traffic shaping to avoid the overstressing of the wireless links between the AR and the MR. A traffic shaping mechanism at the AR was also considered but would have violated the requirements of D03 (section 3.5.2), i.e. no changes outside the OverDRiVE network are allowed. Shaping the MR-HA tunnel calls for the knowledge of the capacity of the network path between the MR and its HA. D07 suggested an estimation of the capacity of this tunnel by monitoring the incoming and outgoing traffic at the MR. The results should be sent to the HA in order to support the shaping of traffic to the MR. However, a simple algorithm that calculates an average throughput does not give a meaningful estimate for link capacity (see section 3.3.2). It only allows for estimation of the current throughput via a single link. In addition this approach does not estimate the capacity of the whole tunnel considering the whole network path but only one wireless segment. Especially in the nested case the monitored link does not need to be the bottleneck link of the MR-HA tunnel. To overcome this last problem to some degree a signalling mechanism was introduced to propagate the results of link monitoring to all nested mobile routers. However, this signalling approach has some disadvantages that will be discussed in the next subsection.

3.3.1 Problems of Signalling

In a nested scenario every lower level MR is dependant on the messages from upper layer MRs in order to estimate the capacity of the MR-HA tunnel. This is not only a security risk. Such approach requires an uninterrupted chain of signalization. Keeping this signalization chain alive premises the cooperation of every MR the MR-HA tunnel traverses. However in a mixed scenario with non OverDRiVE MRs, cooperation cannot be assumed. An interruption of the signalization chain might lead to an overestimation of the MR-HA tunnel capacity because the true bottleneck link is not the wireless link at the MR but another wireless link at a higher layer MR. This might result in the loss of prioritized data at the bottleneck link. An example of this effect is depicted in Figure 23.

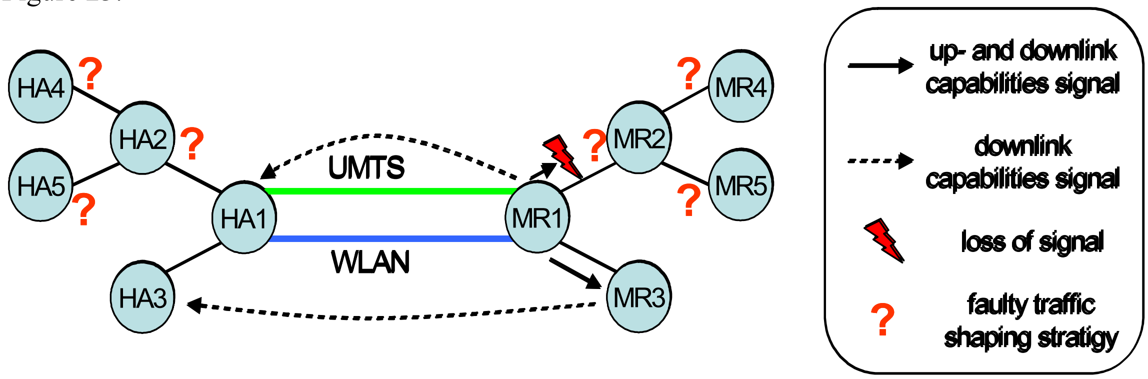

HA4
HA2
HA5
HA1
HA3
UMTS
WLAN
MR1
MR2
MR3
MR4
MR5
?
up- and downlink capabilities signal
downlink capabilities signal
loss of signal
faulty traffic shaping stratigy

Figure 23: Effects of an interrupted signalisation chain

As argued before wireless link capacity may vary both in time and location. Frequent propagation of monitoring results leads to up to date information at all MRs but also increases the signalling overhead. Infrequent updates result in outdated information and incorrect traffic shaping.
Another important drawback of this approach is that is does not consider the segment of the MR-HA tunnel outside the IVAN, i.e. the links between the AR and the HA. If none of the wireless links that connect a MR to the Internet is the bottleneck link of the MR-HA tunnel this again leads to an overestimation of the tunnel capacity.
To overcome this problems an end to end measuring approach might be suitable that does not try to estimate a single link capacity but the capacity of a whole network path, e.g. the whole MR-HA tunnel. Such approach would only need signalization between the MR and its HA. The next sections introduce non intrusive end to end capacity estimation technologies that give a meaningful estimate of link capacity.

3.3.2 Bandwidth Measures

The following sections present a brief overview of methods and techniques to measure bandwidth in packet switched networks. After defining and describing models for measuring bandwidth in data networks, different methods are laid out. The following subsection gives a closer look on packet dispersion techniques. This section results in calculating estimates for the measures on bases of packet pair techniques.

3.3.2.1 Bandwidth, Throughput, Capacity

Many applications in the field of data communication can benefit from knowing the bandwidth of the communication path between nodes. Some examples are

- peer-to-peer networks, where the bandwidth between peers can be used to optimize the distribution of files between multiple hosts,
- overlay networks that can configure routing tables based on bandwidth capabilities between networks,
- video streaming, where the quality can be chosen on the current network state, or
- end-to-end admission control to allow or reject guaranteed bandwidth communication services based on estimates determined by end-to-end measurements.

Within the ISO/OSI layer model the term bandwidth is used to describe different characteristics that are somehow related. At the physical layer bandwidth is used for the width of a frequency band. Link layer technologies often use the term bandwidth to describe the amount of data that can be transferred per unit of time – which is also denoted as throughput of the system. At the transport layer (e.g. TCP) goodput is sometimes referred as throughput. Hence, the terms bandwidth and throughput are somehow understood as being related to the amount of information flowing through a channel or are used as measure of absolute performance of an input/output system.

However, the informal use of these terms (within this and other documents) is sometimes misleading and in accordance with [51] the following definitions try to overcome this. These definitions intentionally introduce and use the term capacity, which can often be found in the context of packet pair measurements (e.g. [52]).

3.3.2.2 Capacity Measure

The following subsections contain the basic terminology for the measure of capacity for packet switched data networks. After describing the basic network model different measures of capacity are defined.

Network Model

The underlying network model for the following capacity measures consists of *vertices*, *edges*, *weights*, *layers* and *nodes*, i.e. a graph $G = (V, E)$ consists of the set of vertices V, a set of directed edges $E = \{(u,v) : u, v \in V\}$, and a function $L : V \rightarrow \{0, \ldots, n\}$, which assigns a layer to a vertex. Additionally, a set of vertices at different layers can be referred to as node $\{n_i \subseteq V : u, v \in n_i \wedge L(u) \neq L(v)\}$.

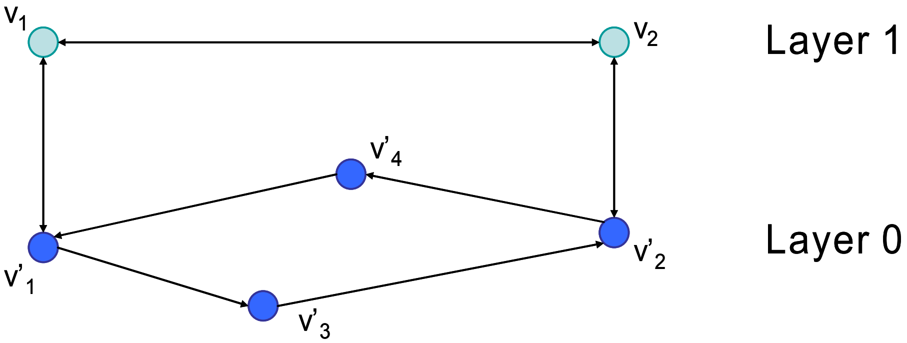


Figure 24: Example of a graph with layers

Figure 24 shows an example of a graph with different layers. The set of vertices is $V = \{v_1, v_2, v'_1, v'_2, v'_3, v'_4\}$ and L assigns layer 1 to the vertices v_1, v_2, e.g. $L(v_1) = 1$, and layer 0 to the other vertices. Here, v_1, v_2 can be interpreted as network hosts from the transport layer view, which are connected by routers at the IP layer. The IP layer is represented by vertices v'_1, v'_2, v'_3, v'_4. In this example the sets $n_1 = \{v_1, v'_1\}$ and $n_2 = \{v_2, v'_2\}$ represent nodes visible on both network and transport layer.

Capacity of an Edge

The capacity (e.g. bits per second ratio) of an edge (u,v) connecting two vertices can be specified by a function $C_{(u,v)}$. $C_{(u,v)}$ can describe the capacity of an edge in different ways based on the level of detail, e.g. as constant function or packet size dependent function $C_{(u,v)} : N \rightarrow \Re$.

If the weight of an edge is not specified, an upper bound can be given if weights are specified for lower layers (cf. Figure 25).

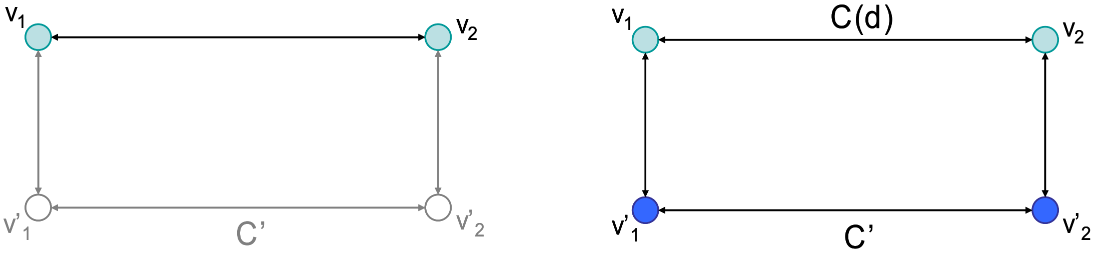


Figure 25: Capacity of an Edge

Assuming a packet switching network with a signalling overhead, the upper bound of the edge is determined by the transfer rate r and the packet size $s = d + h$. Here, d is the size of an upper layer packet and h is the (header) overhead specific to the edge. The time to transmit a packet is given by $t = {}^{s}\!/_{r} = {}^{d+h}\!/_{r}$. Therefore, the capacity of edge e is given by

$$C_e(d) = \frac{d}{t} = \frac{d \cdot r}{d+h} = r \cdot \frac{1}{1+\frac{h}{d}}.$$

On link layer technologies that provide variable packet sizes with an upper bound (e.g. Ethernet), capacity can be appraised independent of the packet size, i.e. using the Maximum Transfer Unit (MTU) available for the IP layer entities. This leads to the maximum capacity $\hat{C}$ as an upper bound for the capacities that can be offered. For example, 10Mbps and 100 Mbps Ethernet provide a 1500 Bytes MTU to the upper layer. Sending 1500 bytes of data results in a link layer packet containing the preamble and start delimiter (8 bytes), source and destination address (12 bytes), a length field (2 bytes), the actual data (<=1500 bytes), the frame check sequence (4 bytes). Additionally, a minimum interframe gap must be kept between consecutive packets, e.g. fast Ethernet has a minimum interframe gap of 96 bit times (equals to 0.96 µsec). The resulting link layer packet has a size of 1538 bytes. Therefore, the capacity is bounded by the maximum capacity of fast Ethernet (omitting differentiation between fullduplex/halfduplex modes and VLAN tagging) can be appraised.

$$C_{eth}(d) \le \hat{C}_{eth} = 100 \cdot \frac{1}{1+\frac{38}{1500}}\ Mbps \approx 97.5\ Mbps$$

If the capacity of the vertex using an IP packet size of 84 bytes (ICMP default) the capacity is bounded by $C_{eth}(84) \approx 68.9\ Mpbs$. On the other hand, the capacity of an edge using packets of size d>MTU, i.e. fragmenting the payload, can be calculated by averaging the capacities of all fragments.

Capacity of a Path

If two vertices v_1, v_n are connected via multiple edges $(v_1, v_2), \ldots, (v_{n-1}, v_n)$, the resulting capacity of the path from v_1 to v_n is the minimum capacity of all edges, i.e. the capacity $C_p(d)$ of a path $p = (v_1, v_2, \ldots, v_n)$ is determined by an edge with minimal capacity. These edges are called *narrow edges*.

$$C_p = \min_{i=v_1,\ldots v_n} C_{(v_i, v_{i+1})}$$

If at least one capacity function on the path is not constant (e.g. is depending on packet sizes) the resulting path capacity inherits the dependency. Taking into account that upper bounds for edges can be defined via lower layers, the capacities of every edge on this layer must be derivable before the capacity of the whole path can be described. An example is shown in Figure 26: The graph consists of 3 layers and in order to describe the capacity of the edge (v_1, v_2), the path capacity of (v'_1, v'_2, v'_3, v'_4) has to be defined which is dependent on the capacity of the lower layer. This example can be mapped on a real-word scenario where the IP layer capacity depends on multiple layers underneath.

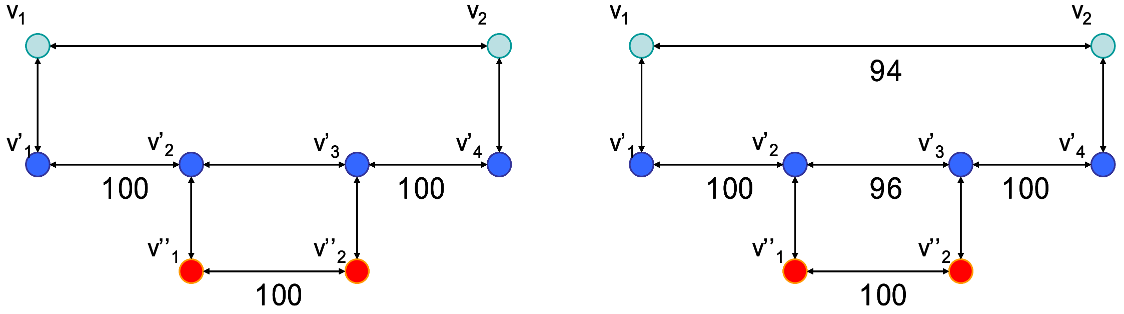


Figure 26: Packet Size dependent Capacity of a Path

The maximum capacity of a path $\hat{C}_p$ can be defined if the all capacities have an upper bound by

$$\hat{C}_p = \min_{i=v_1,\ldots v_n} \hat{C}_{(v_i, v_{i+1})}.$$

Utilization

The former capacity definitions for edges and paths are free of time aspects. They describe upper bounds for capacity. Before introducing a capacity model that reflects time to some extend, the utilization of an edge or a set of edges is presented. This approach is based on definitions given in [51].

The utilization $u(t): \Re \to [0,1]$ is the instantaneous utilization of an edge or a set of edges. Note that $u(t)$ is usually a binary function. A value of 0 denotes that the edge is idle and a value of 1 denotes that the edge is busy. The average utilization $\overline{u}(t_1, t_2)$ between time t_1 and t_2 $(t_2 \geq t_1)$ for the continuous case is given by the following Lebesgue integral.

$$\overline{u}(t_1, t_2) = \frac{1}{t_2 - t_1} \int u \, d\mu \, .$$

If the utilization is defined for a set of edges, they can be interpreted as a baseband medium.

Spare Capacity

The spare capacity S of an edge is defined according to the average utilization $\overline{u}$ of the edge and the maximum capacity $\hat{C}$.

$$S = (1 - \overline{u}) \cdot \hat{C}$$

It represents the amount of capacity that is available w.r.t. the averaging time scale of the utilization function. In this way the spare capacity of a path $p = (v_1, v_2, \ldots, v_n)$ is defined as

$$S_p = \min_{i=v_1,\ldots v_n} (1 - \overline{u}_{(v_, ,v_{i+1})}) \cdot \hat{C}_{(v_i ,v_{i+1})} .$$

The edges that are minimal according to the prior definition are called *tight edges*.
The term spare capacity is introduced instead of available bandwidth (cf. [51]) as it represents the time a certain edge is not fully utilized.

3.3.2.3 Bulk Transfer Capacity Measure

The introduced capacity and spare capacity try to describe upper bounds for a data per time unit ratio. Based on an abstract network graph model, these definitions are somehow independent of a specific network layer. Compared to the introduced measures, Bulk Transfer Capacity (BTC) relies on measurements in real networks instead of a graph model.

BTC (cf. [53], [54]) emphasizes the capacity at the transport layer within the OSI layer model. The BTC measure is closely related to the standard TCP congestion control defined in [55]. BTC is tightly coupled to the algorithmic approach of TCP-Reno and goes along with an application layer implementation, i.e. being as platform independent as possible.

BTC describes the byte per second ratio delivered to an application, which is also called (duplicate free) throughput or goodput in the scope of TCP. The resulting capacity of the network path is actively probed (cf. 3.3.3) using the mentioned TCP-Reno approach.

The benefit of BTC is the platform independent approach of the TCP Congestion Control algorithms (slow-start, congestion avoidance, self-clocking, fast retransmit, fast recovery). Therefore, it can lead to a uniform measure of capacity of the predominant transport protocol used in the Internet.

3.3.3 Measuring Methods

Acquiring quantitative data in communication networks is also a matter of where to put measuring points. In order to compute estimations out of a set of samples, it is also important where measuring points can be placed. Many interesting measurement can only be performed in networks across different administrative domains, e.g. using data paths across multiple providers within the Internet. These *field trials* include the opportunity of realistic network behaviour in the large scale. On the other side nodes (routers, switches, hosts, and servers) are often not accessible, i.e. trying to gather factors that might influence the measurements is not possible, and even end-to-end measurements sometimes need dedicated access to the hosts at the ends of the network path, e.g. in order to capture packets from a network card.

As field trials are sometimes influenced by many unknown factors or at least hard to identify, *laboratory test* offer a less strict access control and are dedicated for a more precise analysis of collected data. Usually, all nodes can be advised to gather information needed for an analysis, e.g. router traffic can be collected via SNMP and MRTG [56], packets at a node can be collected via well-known network sniffers, and a variety of statistics form switches can be collected via RMON [57]. But laboratory tests are often limited by the number of nodes, type of traffic, and – last but not least – technologies available for testing.

Following, methods – the list is surely not exhaustive – are described by categories that are not mutually exclusive.

3.3.3.1 Sender and Receiver based Methods

One classification can be made between sender based and receiver based methods. Sender based methods consider information which are available at the sending host. In this context ICMP is one protocol that allows for sender based techniques. The tool pathchar [58] and [59] use or at least suggest ICMP to infer path characteristics or node traversal costs. These sender based methods send ICMP messages to routers on a network path. The ICMP replies are used to determine characteristics mentioned.

The *sender based* method needs only access on a single machine and can use for example ICMP message exchanges. Additionally, techniques based on ICMP echo messages only need restricted access to the sender. In order to reduce errors influencing the measurements, precise timing for in- and outgoing packets is beneficial. But measurements performed with the sender based method have some drawbacks. At first, the node or path segment under investigation needs to reply on ICMP messages. Some nodes in the Internet do not react on ICMP messages. Secondly, measurements are subject to forward and reverse path effects, i.e. it is not possible to distinguish between characteristics adhering to one path. For example, in case of asymmetric link technologies or disjoint forward and reverse paths this method can not distinguish between these path properties. Furthermore, each estimate comprises errors caused by cross traffic, precision of measurements and delays introduced by entities on the path which makes sender based methods more sensitive.

Receiver based methods (sometimes referred as Sender-Receiver based methods) use the cooperation between two nodes. One node generates data to be evaluated at the receiving host. This method needs access to both hosts and if the receiving host should consider all received packets, privileged access is needed (sniffing). Considering all packets sent and received at the receiving host allows for an evaluation, which respects current network load and packet interaction between the generated packets from the sender and additional traffic. Like for sender based methods precise timing for in- and outgoing packets is beneficial but might require additional access rights at the sender. As receiver based methods only take the forward path between sender and receiver into account, measurements can be expected to be more precise compared to the sender based method.

One additional method, which can also be seen as a sub-category of the receiver based method, is the *receiver-only based* method (cf. [60], [61]). This method only relies on information available at the receiving host. Measurements need a careful analysis as the characteristics form the sending host must be taken into account, e.g. the receiver can not per se differentiate between delays caused by sending host or the network.

3.3.3.2 Passive and Active Methods

Another classification of methods is the distinction between passive and active methods. Passive measurement techniques or applications rely on data that is injected into the network from other nodes or applications. Therefore, it does not perturb or influence the ongoing communication.

Active measurements inject probes into the network or send purposive traffic patterns. One must be aware if this method is used, as it influences other traffic, e.g. if one tries to gather the spare capacity of a network path by flooding, it will swamp other TCP streams out.

3.3.3.3 Intrusive and Non-intrusive Methods

The transition between intrusive and non-intrusive methods is somehow smooth or bound to a specific scenario. Intrusive methods heavily influence the network, e.g. they flood the net. On the other hand non-intrusive methods

- try to avoid influence on existent streams regarding the injected traffic or
- utilize methods to derive parameters indirectly, e.g. they are not flooding the network in order to measure the path capacity, or
- attempt to use only spare capacities and available system resources in order to determine parameters.

3.3.3.4 Explicit and Implicit Methods

Measuring methods can not only be used to gather specific estimates for a link or network path but they can also be used to implicitly control the data communication. One example is TCP’s congestion control that tries to measure the current bandwidth-delay product of the used network path. The estimation of the round trip time and the available bandwidth steer the increase or decrease of the injected stream – they build a closed loop. Explicit methods provide an estimate to the user or an application of the properties of interest.

3.3.4 Packet Dispersion Techniques

One of the first, well known papers describing the packet dispersion effect is [62]. Here, the packet dispersion effect is used to determine the time to inject new packets into the network, which is also known as the self-clocking effect of TCP. The underlying idea is to send packets back to back, i.e. as fast as possible, and analyse the signature when packets leave the network.

Up to now, several papers and tools are using packet dispersion techniques to describe performance measures of IP networks. The following sections present two techniques belonging to this category: packet pair dispersion and packet train dispersion using equal sized packages. Both are used in the scope of capacity estimation in packet switched networks. Producing estimates about network or path capacity is the main challenge for both techniques.

3.3.4.1 Equal Sized Packet Pairs

Packet pair measurements can be used to determine the capacity of a network path. The basic idea is to inject two packets *back to back* into the network, i.e. at the edge capacity or the system throughput of the originating host. In order to describe the measurement results at the destination node, two cases are regarded in the following: The basic case of an unutilized network path and the case of cross traffic, i.e. packets interfering with the packet pair.

In both cases two simplifying assumptions are made:

- The packet processing at store-and-forward entities like routers and bridges are First-Come-First-Served (FCFS), i.e. packet pair measurements affected by packet scheduling or shaping strategies are not regarded.
- The service time to send a packet is determined by the edge capacity and the size of the packet. As a consequence, packet pair measurements influenced by specific link layer characteristics (e.g. TDM, ARQ at link layer) are ignored.

Basic Case

Packet dispersion for packet pairs is sketched in Figure 27. The packet pair between the first two boxes is sent back to back. The edge provides a capacity of C_1 bits per second and the time between the beginning of the first packet and the second packet is $\Delta t_1 = \frac{d}{C_1}$. When the first packet traverses the second link in Figure 27, the service time s to send the packet is $s = \frac{d}{C_2}$. As mentioned, both packets have the same size and the second packet is ready to be sent after Δt_1, but as $C_2 < C_1$, the second packet experiences a delay of $w = s - \Delta t_1$ before the first bit can get on the link. The time between the beginning of the first packet and the second packet on this link is $\Delta t_2 = \frac{d}{C_2}$ and both packets are sent again back-to-back.

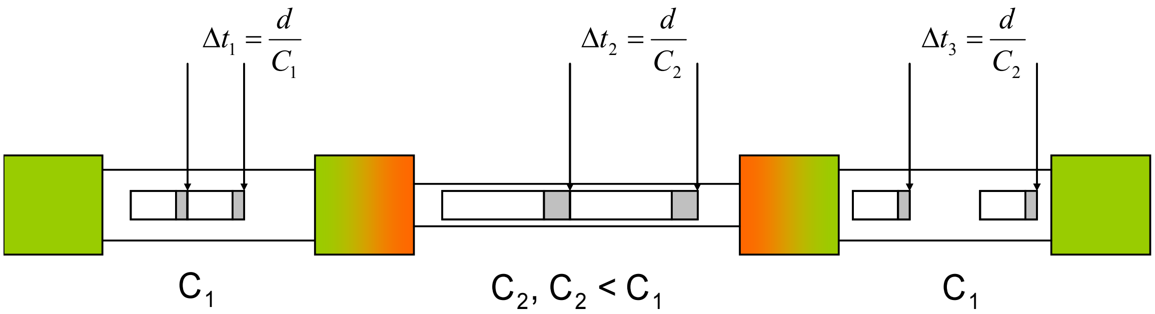

$\Delta t_1 = \frac{d}{C_1}$
$\Delta t_2 = \frac{d}{C_2}$
$\Delta t_3 = \frac{d}{C_2}$
C_1
C_2, $C_2 < C_1$
C_1

Figure 27: Packet Pair Dispersion

After the packet is received on the third hop, the service time s to forward the packet is $s = \frac{d}{C_1}$. As $C_2 < C_1$ the second packet experiences no delay because the service time to forward the packet is lower than the time to receive the second packet, i.e. $s - \Delta t_2 < 0$.

The spacing on the third link is determined by the service time s of the first packet and the time $\Delta t_2 - s$ until the second packet is received. The spacing Δt_3 is given by $\Delta t_3 = s + (\Delta t_2 - s) = \Delta t_2$, i.e. the spacing of the link with the lower capacity remains: The narrow edge of the path leaves its signature.

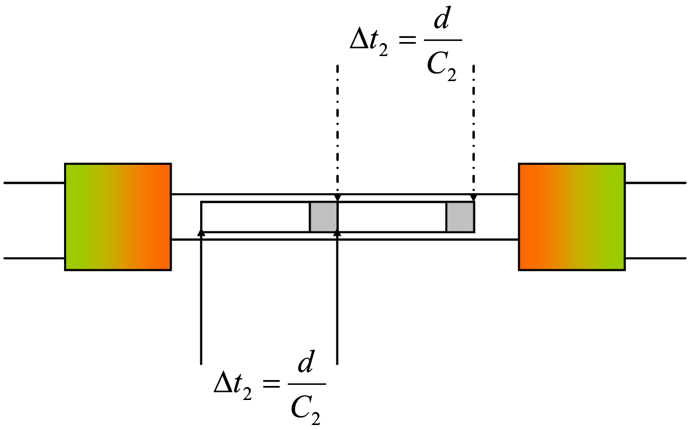


Figure 28: Measuring Packet Dispersion

When the packet dispersion is measured at the end node, the distance between the packets is normally determined by the times of the “last bit” – the timestamp after the packet is received – instead of the time of the first bit. This is sketched in Figure 28.

Cross Traffic Case

The dispersion of packet pairs can be influenced by packets, which is called cross traffic. This interference can lead to misinterpretation of the path capacity measure at the destination. The two cases are shown in Figure 29 and Figure 30.

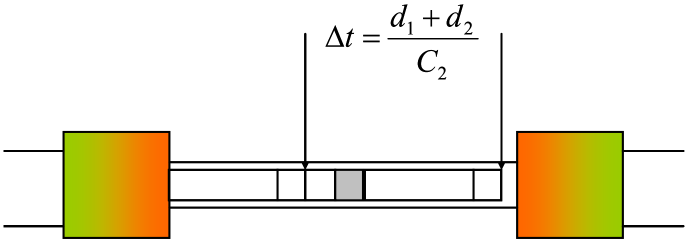


Figure 29: Sub Narrow Capacity

The measured dispersion of a packet can underestimate the capacity of the path, if packets at the narrow edge are inserted between the packet pair. After the reception of the first packet of the packet pair and before the complete reception of the second packet, additional packets can be received during the period $\Delta t' = \frac{d}{C}$, e.g. via other edges. The second packet experiences a delay $w = (s - \Delta t') + t$ at the narrow edge, where s is the service time to process the first packet of the packet pair and t the time to process packets in-between. This results in an underestimation of the bottleneck link. An example is given in Figure 29, where one packet is inserted into the pair, which increases the dispersion of the packet pair and decreases the measured capacity, consequently. It follows from $\Delta t'$ that the probability of packet insertion is dependent on the size of the second packet.

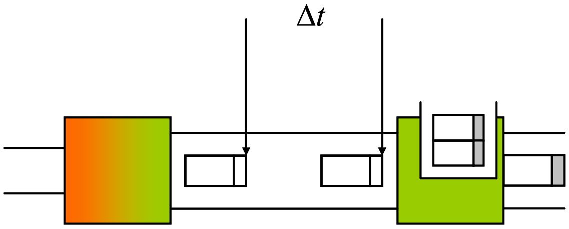


Figure 30: Post Narrow Capacity

The second effect of cross traffic takes place behind the narrow edge of a path. The experienced dispersion of the narrow edge can be decreased or eliminated. This happens when a subsequent node can not process the packet pair immediately, i.e. the packet pair is appended to a queue, as sketched in Figure 30. Therefore, the processing of the first packet is subject to the service time s and a delay w' until the first packet can be serviced. The dispersion $\Delta t'$ behind the narrow edge is $\Delta t' = \max(0, \Delta t - w')$. The decreased dispersion overestimates the measured bandwidth.

3.3.4.2 Equal Sized Packet Trains

The idea of packet pairs is sometimes extended to packet trains, which are multiple packets sent back to back. At the destination the average dispersion $\bar{a}$ is determined by $\bar{a} = \frac{(n-1) \cdot d}{\Delta t}$, where n is the number of packets sent, d is the packet size, and Δt the time between the reception of the first and the last packet.

Although not discussed here, it is worth mentioning that the application of packet train measurements are useful in multi channel scenarios (cf. [63]), where multiple packets can be sent at the same time. In this case a packet pair measurement would underestimate the narrow link capacity. Additionally, if traffic shapers utilize token buckets (e.g. [64]) to allow for burst of packets, packet trains can be adapted to a specific length in order to detect this type of shaping.

If no cross traffic is present, two subsequent packets experience a dispersion $\Delta t'$ at the narrow edge. In this case the average dispersion is $\frac{(n-1)\cdot d}{\Delta t} = \frac{(n-1)\cdot d}{(n-1)\cdot \Delta t'} = \frac{d}{\Delta t'}$, which is the capacity of the narrow edge.

Regarding cross traffic, the average dispersion of a packet train is sometimes used to estimate the available capacity of a path. But as stated in [52], this average is asymptotically lower than the path capacity and does not relate to the available bandwidth.

3.3.5 Application of bandwidth measuring techniques to the OverDRiVE architecture

3.3.5.1 End to end approach MR-HA tunnel

The MR-HA tunnel is bidirectional but not necessarily symmetric. On one hand, most radio technologies provide asymmetric services to accomplish the requirements of web browsing, video streaming and other asymmetric services. On the other hand, the bottleneck link is probably one of the radio links near the MR. This scenario does not demand for different bandwidth measurement technologies at the MR and its HA, but may have an influence on the accuracy of measurements. Therefore the following sections do not differentiate between measuring the tunnel capacity from the HA to the MR or vice versa. However, a deeper insight in error estimation for both directions of the MR-HA tunnel is given in section 4.4.2.

Sender and Receiver based Methods

Section 3.3.3 introduced the distinction between sender and receiver based measuring methods and discussed the advantages and disadvantages of both approaches. Sender based techniques do need neither measurements at nor status reports from the receiver. This would keep the complexity at the receiver low which facilitates mobile devices. However the MR-HA tunnel is bidirectional. This means that the MR would be not only the receiver but also the sender. Thus there is no unloading of mobile devices. In addition the requirements of sender based measurement methods are not met. As already stated in the previous section these techniques cannot distinct the characteristics of a single direction of a bidirectional path. However this is essential because mobile links may be asymmetric and therefore require different traffic shaping parameters at the MR and its HA. In addition the ICMP replies of the receiver induce additional traffic to the reverse path. The increased sensitivity of sender based methods is also a drawback in the OverDRiVE scenario. As a consequence sender based measuring methods should not be applied to the MR-HA tunnel.

Receiver based methods consider only one direction of a bidirectional path. This does not only lead to less sensitivity to errors but makes this approach applicable to an asymmetric MR-HA tunnel. This makes receiver based methods applicable to the OverDRiVE scenario. The receiver is also capable of filtering the measurement results and sending a summarized report back to the sender. This offloads the reverse path from the receiver but reduces the refresh period. A dynamic mechanism may be applied where the receiver notifies the sender when important changes to path capacity, e.g. due to a handover occur. Based on these results the sender can perform traffic shaping (cf. section 3.5.7 of [2])

Receiver only mechanisms may also be applicable to the OverDRiVE scenario in principle but may be not responsive enough to support handovers.

Active and Passive Methods

Using a passive approach means monitoring incoming traffic without creating any additional packet pairs. However, this approach assumes that enough equal sized packet pairs were sent back to back, e.g. TCP packets of the same run, are being sent to the receiver. To generate equally sized packets, padding can be used in combination with encapsulation.

Identifying packets with a high potential bandwidth needs an inspection of the content of those packets. Unfortunately, this cannot be assumed in the OverDRiVE scenario. Multiple tunnels can be nested even if there is only a single MR and a VMN. Combined with encryption, the MR and HA cannot inspect the content of the inner tunnel. Nested MRs even worsen this problem.

Shaping at the sender (the MR or HA), i.e. delaying some packets in order to form packet pairs with high potential bandwidth should not be performed, because it might interfere with end-to-end measurements and rate control mechanisms of the tunnelled traffic.

A dedicated probing stream with marked packet pairs does not have those drawbacks. While being little more intrusive than passive approaches the sender has full control in when to send packet pairs. This makes the approach responsive to changing bandwidth. A separate packet pair probing protocol is possible.

Packet Pair

Packet pairs can either be sent on the same interface that is used to maintain the MR-HA tunnel or on the MR-HA tunnel interface itself. Sending packet pairs on the tunnel interface ensures on the one hand that the probing steam is not treated differently at routers on the network path. If encryption is used these packets cannot be identified by their content, which may help to increase security. On the other hand when using the tunnel interface, packet pairs are subject to the same tunnelling effects, i.e. delays due to encryption and fragmentation, as data traffic. This ensures meaningful measurement results.

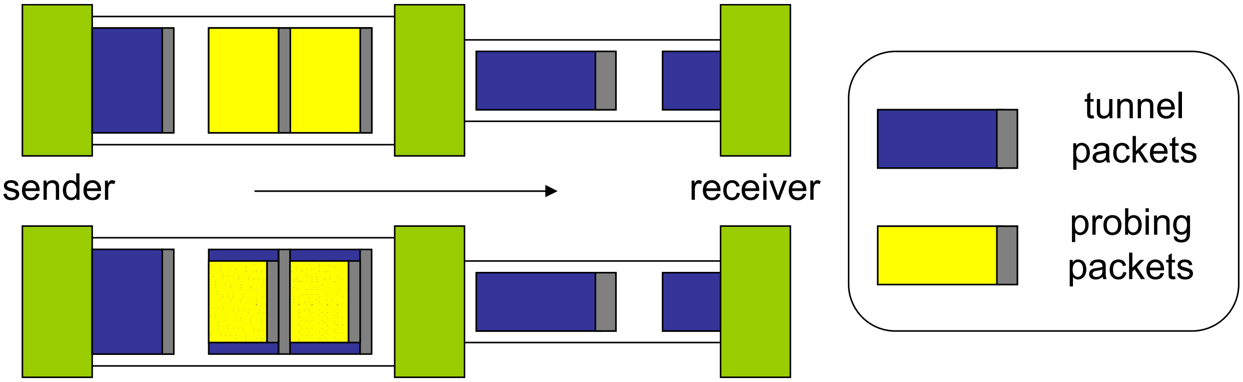


Figure 31: Probing Packets on the same Interface (top) and on the Tunnel Interface (bottom)

Section 4.3.3.1 describes a packet pair generator that was used to generate a dedicated probing stream for evaluation purposes.

3.3.5.2 End to end approach MN-CN

Additional to a MR-HA based approach every application can do measurements. In contrast to the MR-HA tunnelling scenario, passive measurement techniques may well be applied at an application level. The application can inspect the content of data packets received and can apply knowledge of packet sending times to the analysis. In addition not all applications need to report the result of their measurements to the sender.

Section 4.3.3.1 introduces a prototypical implementation of a video streaming software that utilizes the characteristics of a video stream. RTP packets containing fragments of the same video frame are being sent back to back with a high probability. In this case no additional probing packets are required.

4 Validation of Mobile Router and Dynamic IVAN Management

4.1 Introduction

In this chapter we provide detailed descriptions about the implementation of our testbeds and about the evaluation of our concepts with several measurements. First we describe the two implementations of the Mobile Router – Home Agent (MRHA) bidirectional tunnelling mechanism and we present their extensions, which are GPRS, UMTS, vertical handover, fast handover, local fixed node and multicast support.

We present our traffic management testbed, which we used to prove our traffic management concepts with emulation. Traffic management helps to avoid overstressing the mobile router’s radio links. Since UMTS is not yet widely deployed in these experiments we used ADSL to emulate UMTS links.

To understand the handover outage time measurements we performed in our moving network testbeds we introduce an IP handover taxonomy. After the explanation of the different type of IP handovers we present some IP handover and UMTS/WCDMA measurement results.

We also describe an analytical model, that estimates the efficiency of a link for different packet sizes, which we use to verify the packet pair measurements we present in this chapter, as well.

4.2 IETF Draft

The first experiments with moving networks deployed in a vehicle were performed during June-July 2003 in Paris, France. A description of these experiments can be found in [12]

4.3 Testbeds

In this section we present the implementation description of two network mobility testbeds, one based on LIVSIX from Motorola and another one based on MIPL from Helsinki University of Technology (HUT).

4.3.1 Implementation based on LIVSIX

The Open Source LIVSIX IPv6 Stack for Mobility Environments was used on the Mobile Router and on the Home Agent. The detailed implementation description can be found in D14.

4.3.2 Implementation based on MIPL

ETH’s testbed is built in a hierarchical way, using the BRAIN Candidate Mobility Management Protocol (BCMP) for handling node mobility inside a moving network, while the movement of the entire moving network is maintained by using the Mobile Router – Home Agent (MRHA) tunnelling approach (see Figure 32). Having MRHA for mobility management on one side of the network and BCMP on the other requires a special mobile router (MR), which is able to connect these two sides. Thus, ETH’s MR shows MRHA capabilities towards the MRHA part of the network – the Internet – and also has BCMP functionalities towards the infrastructure inside the moving network.

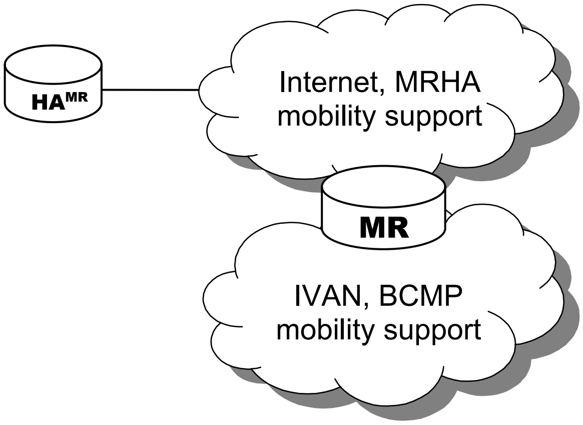


Figure 32: BCMP handles mobility inside the moving network, while mobility of the Mobile Router's network is solved with the MRHA approach

The topology of the testbed can be seen in Figure 33. Both the MRHA and the BCMP part of the network consist of three separate entities. The MRHA part has a home agent (HA) and two access routers (AR1 and AR2), while the BCMP part includes two other access routers (AR3 and AR4) with BCMP capabilities and one equipment (MR) with mobile router, BCMP anchor point and BCMP user registry functionalities. Besides these elements there are also a visiting mobile node (VMN), a local fixed node (LFN), a correspondent node (CN), a virtual GGSN (VGGSN) and a UMTS frontbox equipped with Windows XP present.

Except the mobile node and the correspondent node, all the testbed entities are desktop PCs with AMD Athlon XP 1800+ processors, 512 MB RAM and 40 GB HDD. The network connection is provided via Intel Ethernet cards and, where wireless connection is used, via Avaya 802.11 PCMCIA wireless network adapters. The mobile node and the correspondent node are ASUS laptops with Intel 1.8GHz processors. The mobile node also has an Avaya 802.11 PCMCIA wireless adapter. All the entities run Debian Linux Sarge (test version) operating systems compiled with 2.4.20 kernel and patched with (0.9.5.1) MIPL (Mobile IPv6 for Linux) patch in order to gain Mobile IPv6 support. The basic kernel configurations for all nodes are almost identical, except for the home agent. Before compiling the kernel with the MIPL patch, there is an option to be selected according to the role of the given entity in a moving network. MIPL supports both home agent and mobile node functionalities, but at a time only one of them can be selected. Thus, all entities in the testbed are compiled with mobile node functionality, except for the home agent, which is configured with a home agent role support. The 802.11 adapters are not installed by means of the kernel's PCMCIA basic package but with the package *pcmcia-cs (Card Services)* to gain Demo Ad-Hoc functionality for the wireless cards.

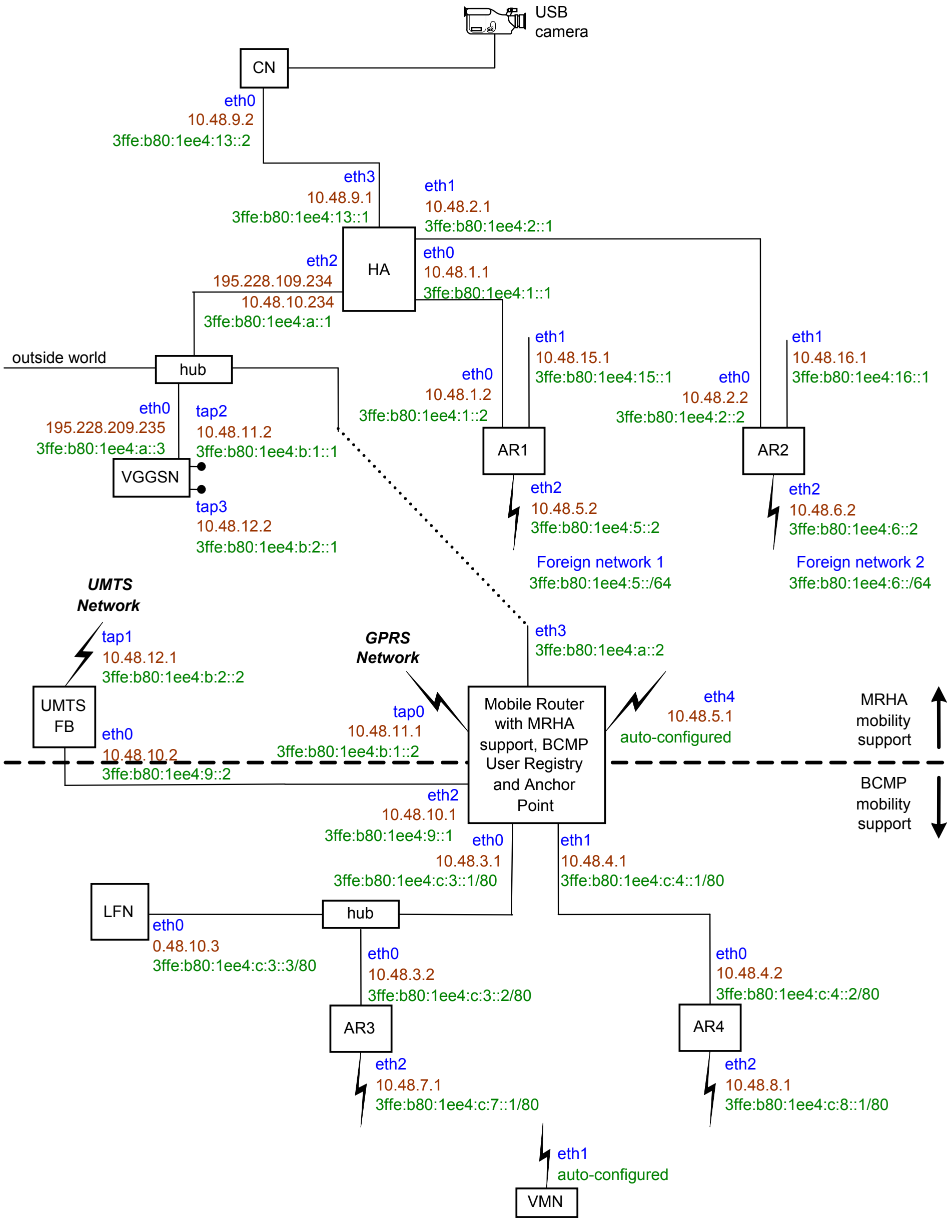


Figure 33: Traffic Lab’s OverDRiVE testbed

Hence the MRHA solution is based on the Mobile IPv6 mobility draft [13], throughout the design and implementation IPv6 addresses, signalling and tools are used.

4.3.2.1 MRHA tunnelling solution part of the testbed

The initial setting of the environment is that the mobile router (MR) is attached to its home network, thus, located in the same network together with its home agent (HA). The home agent and the mobile router have been connected by means of a cross UTP (Unshielded Twisted Pair) cable. The home network has been given IPv6 subnet mask `3ffe:b80:1ee4:a::/64`, the home agent has been provided with the address `3ffe:b80:1ee4:a::1`, while the mobile router has the address `3ffe:b80:1ee4:a::2`. As it can be seen, the "`3ffe:b80:1ee4:a`" parts of the addresses are used in the home network.

An access router (AR1) has been installed to the network, representing the access point for mobile nodes to a foreign network. The access router has been directly connected with the home agent by means of a cross UTP cable via interfaces `3ffe:b80:1ee4:1::2` and `3ffe:b80:1ee4:1::1`, respectively. Although the two entities are, for simplicity, physically connected, this wire can be substituted by (and also represents) any arbitrary path on the Internet. The foreign network has been given the subnet mask `3ffe:b80:1ee4:5::/64` and the access router's interface towards the foreign network has been provided with the address `3ffe:b80:1ee4:5::2`. This latter interface is wireless, in order to grant connection for mobile entities to the foreign network.

In order to analyse the behaviour of moving networks while changing connections to the Internet, an additional foreign network has been introduced to the testbed, in a quite similar way as described previously. The point of attachment to this other foreign network is also an access router (AR2). The only difference between AR1 and AR2 are the addresses assigned to the interfaces involved. AR2 is also directly connected to the home agent by means of a cross UTP cable via interfaces `3ffe:b80:1ee4:2::1` and `3ffe:b80:1ee4:2::2`, respectively. This second foreign network has been given the subnet mask `3ffe:b80:1ee4:6::/64` and AR2's wireless interface towards this foreign network has been provided the address `3ffe:b80:1ee4:6::2`.

The home agent has to advertise its services on the home network so that other hosts on the network can make use of its capabilities. The program called *radvd* can be used to advertise certain pieces of information on a network, thus, it is ideal for the purpose. This program is particularly configurable by means of the file */etc/radvd.conf* (as example see D.1).

The program *radvd* runs with this aforementioned configuration file and it sends periodically router advertisement messages on the home network telling that on interface eth2 with address `3ffe:b80:1ee4:a::1` a home agent can be found. Note that there are several other optional parameters of *radvd* (eg. message sending period). Leaving them out from */etc/radvd.conf* the program runs with the default values.

When the mobile router leaves the home network and wants to connect to a foreign one, it finds the point of attachment also by means of router advertisement messages. Thus, the access points in the foreign networks also have to run the program *radvd*. This time however, the parameters must be set differently, since the purpose of the advertisement here is not broadcasting home agent capabilities, but to share the information with the mobile router, and to advertise that there is a foreign network nearby with a certain network prefix (as example see D.2).

Finally, there are parameters configured also on the mobile router in order to provide the desired relationship between itself and its home agent (as example see D.2).

As defined in the configuration file, the "mobile node" functionality is chosen. The reason of this choice is that in the plain Mobile IPv6 protocol there is no such functionality as "mobile router", thus, among the available options (mobile node, correspondent node, home agent), the most similar functionality is the most reasonable choice.

By running *radvd* on both the home agent and the first access router with the appropriate settings and disconnecting the mobile router from its home network the following actions are performed: the mobile router notices the presence of a nearby foreign network, and thus its wireless interface gains a new care-of-address, namely `3ffe:b80:1ee4:5:202:2dff:fe42:d569`. This address is automatically generated by combining the received foreign network prefix and the PCMCIA wireless network adapter's hardware address. After the care-of-address is assigned, the mobile router automatically sends a binding update message to its home agent and builds a tunnel between itself and the home agent. The tunnel can be represented as an interface (with name *ip6tnl1*) and it also shows up when using the *ifconfig* command and in the routing table as the default gateway (see *route -A inet6*). (As example see D.4.)

Upon the receipt of the binding update message, the home agent creates an entry in its binding cache assigning the mobile router's home address with its care-of-address and creates a tunnel towards it. This reverse tunnel, similarly to the tunnel created by the mobile router, can be represented by an interface (also with name *ip6tnl1*), and it shows up in the routing table as the interface to be used for packets destined for the mobile router. (As example see D.5.)

The mobile router's disconnection from a foreign network can be triggered by several reasons. One of them is that the mobile router loses contact with its access router, thus, its point of attachment to the foreign network. If there is another foreign network's access router nearby, handoff to that foreign network can be performed. Otherwise, the mobile router – and thus its moving network – loses contact with the Internet. By starting *radvd* on the second access router with advertised prefix `3ffe:b80:1ee4:6::/64` and pulling down the first access router's wireless interface the mobile router – upon receipt of the new router advertisement messages – gains a new care-of-address (`3ffe:b80:1ee4:6:202:2dff:fe42:d569`) with the appropriate network prefix. The steps followed by the mobile router and the home agent are similar to the case when the mobile router joined the first foreign network: MRHA and reverse MRHA tunnels are reconfigured, the binding update list, the binding cache and the routing tables are automatically refreshed according to the new address information.

Providing routing capabilities for the mobile router

The forwarding behaviour of network elements can be changed by setting the *forwarding*[6] value to 1 or 0 whether allowing IPv6 packets to be forwarded between interfaces or not, respectively. For mobile nodes this value is set to 0 by default. However, in the case of a mobile router it is crucial to set this variable to 1 since setting it to 0 prevents the mobile router to forward packages and thus, to fulfil its functionality. After the appropriate setting it is expected that the mobile router is able to change foreign networks and is also able to pass IPv6 packets between its interfaces. However, in the source code of the kernel in file *linux/net/ipv6/ndisc.c* the following sample of code can be found (see D.6).

According to the source code sample if the variable *forwarding* is set to 1 the function *ndisc_router_discovery()* returns before the mobile router can be provided with a new care-of-address with the foreign network's prefix. The reason of this solution in the kernel source code might have been inspired by the assumption that routers do not move and thus, they do not need to accept routing advertisement messages. To avoid this pitfall, the kernel source code is to be modified by removing the step where the value of the variable *forwarding* is examined. After recompiling the modified kernel the problem of discarding the router advertisement messages can be solved.

Preserving the moving network prefix

Since the mobile router's task is to ensure connection to the Internet for its mobile nodes, it is necessary to create a routing entry in the mobile router's routing table pointing towards the nodes in order to ensure the delivery of the packages destined to the nodes within the moving network. In the testbed the moving network has been given the prefix `3ffe:b80:1ee4:c::/64`. By registering this address to the routing table manually the forwarding of the appropriate packages towards the moving network is ensured. However, when launching the Mobile IPv6 extension by running the command */etc/init.d/mobile-ip6 start* all the IPv6 routes set previously become lost, and the routing table is rebuilt according to the Mobile IPv6 local settings and information packages (eg. routing advertisements). The same happens at all of the mobile router's handoff and at every

[6] It can be found and modified at */proc/sys/net/ipv6/conf/all/forwarding*

mobility event. Thus, the moving network prefix must be preserved when the routing table is cleaned up.

First, the desired network address must be introduced to the kernel. When the Mobile IPv6 protocol is started, the program called *mipdiag* is run and it passes variables and settings to the kernel, which can be manually set in the file named */etc/network-mip6.conf*. Thus, a new entry is added to this file, namely “`NETWORKADDRESS=3ffe:b80:1ee4:c::/64`”. In the next step the source code *mipdiag.c* must be extended. Extra commands also must be added in order to read this parameter out from the configuration file. The extension of function *rtn_set_mn_info()* is also required so that it does not only pass the home agent’s address and the home address of the mobile router to the kernel but also the network prefix given. The code sample of the required modifications is the following (see D.7). Previous declaration and definition of the flags ”`IFA_NPREFIX`” and “`IFA_NADDRESS`” in files *include/linux/rtnetlink.h* and *mipv6-0.9.5-v2.4.20/src/mipdiag.c* is required.

After the parameters, stored in the configuration file, are read and sent to the kernel by means of the function *rtnl_talk()*, the information must be sent to its appropriate location. The variables given by *mipdiag* are then read by the function *inet6_rtm_newaddr()* in file *addrconf.c*. There are modifications needed also in this function to read the OverDRiVE extension variables as well (see D.8).

The function *addrconf_set_mipv6_mn_network_address(na, nprefix)* is declared in the file *mipglue.h* which is responsible for the mobility integration into the plain IPv6 protocol. The modifications needed in this file can be found at D.9.

Finally, the network prefix can be read by the file *mn.c*, which configures the initial settings of a mobile node, including the new parameters (see D.10).

As mentioned before, at every mobility event the Mobile IPv6 protocol cleans up the whole IPv6 routing table and reconfigures it. With a network prefix associated with the mobile router, the protocol can be informed about the network that lies behind the mobile router and thus, the cleanup of the routes pointing towards this moving network can be prevented. The mobility detection of the protocol is the responsibility of the file *mdetect.c*. In this source code a function can be found that is responsible for the cleanup of the routing table. The function does not delete every route, it first performs a check to each and every route in the routing table, whether it can be deleted or not. Without code modifications the routes pointing towards the manually set network prefix pass the test and as a consequence they become deleted. Thus, an extension to the test must be applied based on a prefix check – with this addition routes with a specific prefix can be preserved from deletion. The modification of *mdetect.c* can be seen in the code sample in D.11.

It can be seen that if the current address under examination has the same prefix as the manually given network address, then the test returns value 0, thus, the route to this address will not be deleted. In the original code the *mipv6_prefix_compare16()* function was not implemented, it is actually a modification of the kernel’s *mipv6_prefix_compare()* function. The reason for writing an additional prefix checker function was that the built-in one checks prefixes for every 32 bits, not for every 16 bits, as needed in this case. The source code of this new prefix checker function can be found in D.12.

With all these modifications described above, it is possible to set up a permanent route in the mobile router towards an arbitrary network.

Making global routing available

When the mobile router (and thus its network) is in motion, the routes towards the access routers are automatically set up with UGDA flags, as in this following example:

```
MR# route -A inet6
Destination  Next Hop                 Flags ... Iface
::/0         ::                       U     ... ip6tnl1
...
::/0         fe80::202:2dff:fe42:d578 UGDA  ... eth3
```

The flags seen in the routing table mean:

- U - Route is up
- G - Route is a gateway
- D - Route was dynamically installed by daemon or redirect
- A - Route was installed by addrconf

According to tests the mobile router set with default gateway parameters D and A showed anomalous behaviour. The function that is responsible for creating default routes (*rt6_add_dflt_router()* in file *route.c*) added all flags to the routes instead of adding only those flags that the user indicated. Modifying this function only the given flags are added to the routes (see D.13).

Maintaining routes on the home agent

While designing an MRHA tunnelling scenario, besides the extensions of the mobile router, a minor modification to the functionality of the home agent must be made. Additions must be applied in the home agent's routing table maintenance towards the moving network's prefix. Initially, when the mobile router is located in its home network, a routing entry can be added to the home agent (pointing towards the moving network) determining the mobile router's home address as the next hop. However, when the mobile router leaves its home network and joins a foreign one, the routing entry containing the mobile router's home address becomes invalid. The mobility event itself does not trigger the correction of this route automatically – on the contrary to the correction of the route towards the mobile router itself which is triggered by the binding update message sent by the mobile router. This implies that the state of the mobile router must be checked regularly and the routing table of the home agent must be maintained accordingly. The two events when changes must be made are when the mobile router leaves the home network and when it returns to it. Initially the routing table of the home agent shows the following:

```
HA# route -A inet6
Destination          Next Hop           Flags ... Iface
...
3ffe:b80:1ee4:c::/64 3ffe:b80:1ee4:a::2 UG    ... eth2
```

While the mobile router moves away from its home network, the address determined as the next hop (`3ffe:b80:1ee4:a::2`) becomes invalid and sending the packet via interface eth2 also becomes void. Thus, the routing entry towards the prefix `3ffe:b80:1ee4:c::/64` must be changed, and since the mobile router is currently reachable via the tunnel interface denoted as *ip6tnl1*, the appropriate commands must be applied:

```
HA# route -A inet6 del 3ffe:b80:1ee4:c::/64 \
    gw 3ffe:b80:1ee4:a::2
HA# route -A inet6 add 3ffe:b80:1ee4:c::/64 \
    dev ip6tnl1
HA# route -A inet6
Destination           Next Hop   Flags ... Iface
...
```

```
3ffe:b80:1ee4:c::/64  ::         U     ... ip6tnl1
```

The routing table now has an up-to-date and valid entry towards the moving network. Similar steps must be applied when the mobile router returns to the home network – the route via the tunnel must be removed and the gateway entry to the home address of the mobile router must be restored. These mobility events – leaving and returning to the home network – can be tracked by regularly checking the state of the tunnel interface. Change in the tunnel preferences means that a mobility event occurred. Examples for the states of the tunnel interface can be found in D.14.

With regular checking of the tunnel's state the routing table can be maintained by deleting the invalid entries and adding up-to-date ones. With a simple script the check of the tunnel's state can be put in a while loop introducing an execution delay of, e.g., 1 millisecond and thus the removal of the old route and addition of the new one can be performed almost instantly. An example for this program in TCL/TK can be found in D.15.

4.3.2.2 BCMP part of the testbed

BCMP maintains the mobility management inside the moving network part of ETH's scenario. In Traffic Lab's OverDRiVE testbed the BCMP network consists of a user registry, an anchor point, two access routers and a mobile node. The mobile router is the node that has both MRHA tunnelling and BCMP capabilities in order to interconnect the two network parts.

The different BCMP nodes are theoretically separate entities with different tasks and responsibilities. However, since the entities are actually programs also, that run in user space, it is possible to integrate some of them in a way that they run on the same machine at the same time. In our testbed the mobile router, the user registry and the anchor point functionalities are integrated into one computer. The access routers are still separate devices since they are quite identical – both in their roles and configuration – and, even if the system is considered to be a real network scenario, they must be located far from each other to provide accessibility for mobile nodes in as wide area as possible. Thus, the mobile router entity shows MRHA capabilities towards the MRHA part of the network – the Internet – and seems to be functioning as a BCMP anchor point and user registry in the BCMP access routers' point of view.

The user registry and the anchor point modules were installed on the mobile router. Following the addressing convention used during the implementation of the mobile router's route preserving functionality, the user registry was configured to advertise the `3ffe:b80:1ee4:c::/64` address space. The main parameters that can be manually configured and are used by both the user registry and the anchor point module can be found in D.16.

After the similar setting of these parameters on each node, the programs for the user registry and the anchor point are started on the mobile router, while the BCMP access routers start the program written for BCMP access routers. In the testbed the mobile node arrives to the moving network with its air interface unconfigured.

Tunnelling interface collision

When a mobile node is in the BCMP network, it is under supervision of a BCMP access router. Naturally, in case of movement, the mobile node can change BCMP access routers within the network at any time. The current BCMP access router information for each mobile node is stored in the anchor point module, since it is responsible for directing the incoming traffic towards the current BCMP access router in order to reach the mobile node. The anchor point forwards the traffic by means of tunnels pointing from the anchor point to the BCMP access router. However, since in this case the mobile router is the node that has anchor point capabilities, the tunnel's endpoints are the mobile router and one of the two BCMP access routers.

In the MRHA tunnelling approach, if the mobile router is not located in its home network, a bidirectional tunnel is built between the mobile router and its home agent. To be more specific, on the mobile router's side a tunnel is created, for example:

```
MR# ipv6tunnel show ip6tnl1
ip6tnl1: IPv6/IPv6 \
         remote 3ffe:b80:1ee4:a::1 \
         local 3ffe:b80:1ee4:5:202:2dff:fe42:d569 \
         hoplimit 255 flags ELKM
```

It can be seen that the MRHA implementation automatically configures the tunnel interface *ip6tnl1*. However, the BCMP anchor point also starts to create tunnels towards the moving network starting from *ip6tnl1*. If the mobile router was installed with BCMP support without MRHA capabilities, it would have a BCMP tunnel configuration like this:

```
MR# ipv6tunnel show ip6tnl1
ip6tnl1: IPv6/IPv6 \
         remote 3ffe:b80:1ee4:c:7::1 \
         local 3ffe:b80:1ee4:c:3::1\
         hoplimit 255 flags EL
```

However, with both MRHA tunnelling and BCMP support turned on, this concludes in an interface reserving collision, and the system can not operate properly. The reason of this reserving convention is that originally BCMP was written for a system that has BCMP capable devices separately, that is, user registry, anchor point, access router programs are all running on separate computers. In this case however, the anchor point and the mobile router are located in the same device, thus, their independency also in their used resources must be assured. This problem can be solved by modifying the BCMP's tunnel reservation algorithm by specifying a tunnel interface sequence number big enough to avoid reservation collision with high probability. By determining an integer for the first tunnel interface to be used by BCMP, tunnels only with equal to or higher sequence numbers will be reserved towards the moving network's BCMP access routers. Since MRHA only uses interface *ip6tnl1*, this sequence number can be set to 2 – representing *ip6tnl2* – but to avoid any possible future collisions it is suggested to choose a higher integer. In our OverDRiVE testbed sequence number 50 was chosen for BCMP tunnel reservations. Thus, the anchor point module will only reserve tunnels *ip6tnl50* and above towards the BCMP access routers. Applying the modifications and enabling both MRHA and BCMP capabilities of the mobile router results a similar tunnel setup if both the MRHA tunnel and a tunnel towards a BCMP access router in the moving network is up (as example see D.17).

Fast Handover

The mobile router's handover between access routers can be triggered in two ways: it can be initiated by the network or the mobile router. In the first case the mobile router automatically performs a handover when it loses the signal of its current access point (access router) and finds another access router nearby. However, this way of changing foreign networks – due to the mechanism of the MIPL Mobile IPv6 stack – is rather slow, the overall process can take from 1.1 seconds up to 6.1 seconds. This is because the stack waits a certain time before it considers the old AR as lost and connects to the new one. This amount of delay can not be accepted when using Mobile IP in real-time applications.

The original MIPL stack does not support mobile router initiated handover. However, with a couple of modifications in the source code it can be implemented. The handover can be initiated by a user space program that sends an *ioctl* call to the kernel and as a result the mobile router switches from its current access router to another one. The source code of the user space program and other necessary modifications can be found at D.18.

With these modifications the handover occurs instantly.

Local Fixed Node support

We extended the testbed to also support local fixed nodes (LFN). A Windows XP laptop played the role of the LFN. To install the IPv6 support on the laptop we did the following steps:

1. we downloaded "Microsoft IPv6 Developer Edition" from www.microsoft.com;
2. at the command prompt we typed `ipv6 install` (`uninstall` to remove);

It is not possible to configure the IPv6 address in the Control Panel -> Network connections -> Local connections -> Properties, thus the following command prompt commands have to be given out:

1. configuring interface: `ipv6 adu 5/3ffe:b80:1ee4:c:3::3` where `5` is the identifier of the interface
2. adding routing entry : `ipv6 rtu 3ffe:b80:1ee4:/48 5/3ffe:b80:1ee4:c:3::1`
3. to delete an interface or routing entry, the same commands should be used but with `life 0` at the end

Since we wanted to reach the IPv4 Internet we had to use a v4/v6 web proxy [67], which was running on the mobile routers home agent. It can be installed by compiling the source code with 'make' and than copying the files www6to4conf and www6to4_forward.conf to the directory /etc. After the successful installation the proxy can be started with the command 'nohup www6to4'.

Although Internet Explorer (IE) has IPv6 support, it can not handle IPv6 addresses in the proxy field. To resolve this problem we put the the IPv6 address of the home agent in the `hosts` file (`C:\WINDOWS\system32\drivers\etc\hosts`) and named it as `odriveproxy`:

```
3ffe:b80:1ee4:a::1    odriveproxy
```

Using the `odriveproxy` name in the proxy field of the IE, the IE running on the LFN found the proxy on the home agent and reached the IPv4 Internet.

Since the Ethernet interface of the LFN was configured to the `3ffe:b80:1ee4:c:3` subnet, there was no need to make any changes in the routing table of the mobile router.

Setting up the GPRS connection

To set up an IPv6 connection between the MR and the VGSSN through the GPRS network first we attached a GPRS enabled mobile phone (Ericsson T68i) to the MR with a serial cable and we built up a serial line connection between the phone and the MR. In the currently available Linux distributions all standard kernels support this type of connection, only the serial interface must be enabled on the MR. In the next step we established a connection between the MR and the Internet through the GPRS network. In our testbed we use the pppd (Point-to-Point Protocol Daemon) user space daemon that establishes the connection and brings up a */dev/pppX* interface on the MR, which is used to send data to and receive data from the Internet. Furthermore the pppd daemon configures an IPv4 address for the */dev/pppX* interface, as well. The configuration of the daemon is stored in the files */etc/ppp/options*, */etc/ppp/chat-gprs* and */etc/ppp/pap-secrets.* The content of the files can be found at D.19.

After having established a connection between the MR and the Internet an IPv4 based tunnel must be created between the MR and the VGGSN that can transmit IPv6 packets. For this purpose we use the *vtund* user space daemon, which runs on the VGGSN and on the MR, as well. The daemon creates a virtual interface *(/dev/tapX*) to transmit data. The configuration information of the daemon is located in the file */etc/vtund.conf* on each device. The content of the files can be seen in D.20.

On the VGGSN the *vtund* daemon has to be started as a server with an –s option, on the MR it has to be run as a client and two parameters have to be given: the IPv4 address of the VGGSN and the name of the session. As it can be seen in the configuration file, the IPv4 and IPv6 addresses of the interfaces are assigned automatically when the interfaces are brought up, that is when the tunnel is successful established. Furthermore the configuration file indicates that the tunnel between the MR and VGGSN is realized with a UDP connection, it was decided to use UDP to avoid TCP’s traffic control mechanisms (e.g., avoiding retransmission).

In the last step of the GPRS connection setup RAs must be sent through the IPv4 based tunnel to the MR. By using these messages, the MR is able to automatically configure its virtual interface that is used to reach the GPRS network. As described earlier, we use the program *radvd* to send the RAs. Its configuration file (*/etc/radvd.conf*) on the VGGSN can be found at D.21.

Setting up an UMTS/WCDMA connection

The UMTS connection of the MR is built up in basically the same way as the GPRS connection; there are only some minor differences. The first one comes by the fact that the UMTS phone we used had a USB interface and we did not have a driver for Linux, but only for Windows XP. Therefore we had to place a frontbox with Windows XP operating system between the mobile phone and the device running the *vtund* daemon. The frontbox’s role was to establish a connection to the Internet using the UMTS phone and to forward IPv4 packets from the moving network to the Internet and backwards.

The second difference comes from the IPv4 routing in the case when GPRS connection is also available. In our testbed the GPRS and the UMTS links are permanently connected, which means that two IPv4 tunnels have to be established between the moving network and the VGGSN at the same time, through GPRS and UMTS respectively. We had to solve that those packets that have to go through the GPRS network go through the GPRS tunnel and those packets that have to go through the UMTS network go through the UMTS tunnel. One solution would be that for each tunnel a different entity of *vtund* (with different port numbers) is started on the MR and on the VGGSN. Thus the MR would route the packets based on the destination address and the port number (NAT like approach). We chose another solution. We started the *vtund* daemons on different computers, which had the default routing entry pointing to the corresponding access network. So we installed another UMTS frontbox (UMTS-FB) with Linux next to the MR, we set up an IPv6 link between the MR and the UMTS-FB and started *vtund* on the machine. The other *vtund* was running on the MR and handled the tunnel through the GPRS connection.

The */etc/vtund.conf* configuration file we used on the UMTS-FB is similar to the GPRS settings (see D.22). On the VGGSN the */etc/vtund.conf* configuration file must be extended with some lines (see D.23).

As by the GPRS case, in the last step of setting up the UMTS connection RAs must be sent from the UMTS-FB to the MR. These RAs are needed in order that the MR can register the sender of those as default router and not because of the automatically address configuration function of the RAs. This way MR can build up the MRHA tunnel through the UMTS-FB and so through the UMTS network. The */etc/radvd.conf* configuration file on the UMTS-FB can be seen in D.24.

UMTS-GPRS-WLAN vertical handover

The vertical handover is done by filtering out the RAs that come from those networks, which we do not want to use at the moment. If we filter out he unwanted RAs, the mobile router automatically sets its default route pointing to the network chosen. Thus switching between access networks needs a modification in the filter rules only.

Multicast integration

We integrated EAB-BUTE's mobile multicast and ETH's moving network testbed. To reach this goal we did the following steps. We put new Ethernet cards into our access routers (AR1 and AR2) and VGGSN. We connected BUTE's three FreeBSD multicast routers to these new interfaces. We installed BUTE's multicast control panel on our mobile router. On this panel we could choose whether we want to receive the multicast streaming content through AR1, AR2 or GPRS. We started a multicast packet forwarder on the mobile router, which program forwarded all multicast packets to AR4. On AR4 a multicast sender was running, which sent out the multicast packets through the radio interface of AR4.

BUTE's multicast mobile node could choose whether it wants to receive the multicast content from AR1, AR2 or through the mobile router from AR4. Thus the multicast mobile node was able to make a seamless multicast handover between AR1 and AR2, and it could roam into the moving network and receive the content through AR4. AR4 received the multicast content from the mobile router, which was able to make a seamless multicast handover between AR1, AR2 and GPRS.

4.3.3 Traffic Management Testbeds

In an OverDRiVE IVAN all wired and wireless nodes share the multi-radio network resources provided by the MR. As stated in D07 [2] these radio links are believed to be the bottleneck for connections between the IVAN and the Internet, i.e. close to the MR. The OverDRiVE traffic management concept aims at not overstressing these links. Two entities are identified to perform traffic management: the MR and its HA. Using other entities – for example the AR – was also considered but this would violate the requirements stated in D03 [1] section 4.1.

The evaluation of the traffic management concept was done by emulation rather than by simulation. In the scope of traffic management, emulation was preferred because of several reasons. Firstly, simulation results strongly depend on the simulation model used. Secondly, the MR-HA tunnel involves both a low capacity link near the mobile router and a sub-path traversing the Internet. Therefore it seems more appropriate to use a path through the Internet affected by real Internet traffic, which is difficult to simulate [36]. Additionally, measurements in realistic environments involve sources of error which are often omitted during modelling, i.e. it might also be difficult to find realistic model parameters without real measurements.

4.3.3.1 Test Applications

In order to measure bandwidth on the basis of packet pairs, two applications were applied. While the first was used for data acquisition and statistical analysis, the second incorporates parts of the traffic management concept in an interactive application.

Packet Pair/Train Generator

A client-server application was developed and implemented in C, which is able to inject UDP packet pairs and packet trains. Special efforts were made to get a fast and flexible implementation. Packets can be sent with a specified inter and intra packet/train send delay. Packet sizes can be set on the basis of IP datagrams. Because IP is a network layer protocol the header of IP packets is not removed on a per hop basis as headers of link layer protocols. This allows for analysis of packet separation of different packet sizes. The client respectively constructs and sends the packet pairs or trains to the server. To increase the accuracy of inter sending times a real-time scheduler was used. The server is an application that logs all incoming traffic received which was send by the client. For this purpose it uses the pcap library [41].

During the development of this application several test were performed via the 6bone [65]. The 6bone connection is a virtual network using IPv6 over IPv4 tunnelling, i.e. it operating over the IPv4-based Internet to support IPv6 transport. It may also involve the use of non state-of the art hardware which can cause unpredictable delay in routing packets which would not occur in a non-testbed like deployment of the IPv6-based Internet. Some effects – like highly increased drop rates – were observed while doing video streaming tests from University of Bonn in Germany to the Motorola Labs in Paris. As consequence the application exclusively uses IPv4 packets to exclude additional effects caused by the 6bone.

Adaptive IPv6 Video Streaming

An adaptive IPv6 video streaming application based on a Apple Darwin Streaming Server 4.1.3 [39] patched to support IPv6 and the mpeg4ip 0.9.9 [40] video streaming client patched to decode mpeg4 videos on an iPAQ 3870 Strong Arm Pocket PC and enhanced to support adaptive video streaming was developed as a proof of concept for the OverDRiVE demonstration at the HyWiN workshop in Turin in December 2003 [66]. The mpg4ip video client measures the path capacity using a passive, receiver based packet pair technique (cf. sections 3.3.3 and 3.3.4) by analysing the video stream generated by the video server and therefore not producing any additional load.

4.3.3.2 Scenarios

While the Adaptive Video Streaming application described was mainly used within the local demonstrator at the University of Bonn and within the testbed at the HyWin workshop in Turin [66] the packet pair/train generator was also used in the domain of an external DSL provider. The scenarios are based on the premises of D03 and D07 and try to reflect the network characteristics affecting OverDRiVE mobile networks.

ADSL and UMTS

The ADSL technology was chosen to substitute UMTS as bottleneck link. UMTS is still in an introductory phase in many countries in the EU and difficult to handle for larger experiments and large amounts of test data. The ADSL technology however is widely deployed and available for extensive experiments. In addition, ADSL has matured since its introduction several years ago.

UMTS and ADSL have several common characteristics that legitimate a substitution of UMTS with ADSL for test purposes as can be seen in Table 2.

UMTS	ADSL
Asymmetric data rates for data service	Asymmetric rates for data service
Data rates up to 2 Mbps	Data rates up to 1.5 Mbit already deployed
Data rates decrease while moving to edge of cell	Data rates decrease with local loop length
ATM in UTRAN and core network (< release 5)	ATM used for access networks

Table 2: UMTS - ADSL Comparison

Both UMTS and ADSL provide asymmetric data services to support multi media applications and Web surfing. While the data rates allow for new services and applications for mobile and home users, they are orders of magnitude lower than standard LAN technologies, which provide data rates from 10 Mbits/s (classical Ethernet) up to 1 Gbits/s (Gigabit Ethernet). WAN technologies provide even higher data rates.

In the UMTS standard (up to release 4) ATM and AAL5 are used in the user plain of the packet switched domain both in the core network and in the UTRAN as depicted in Figure 34.

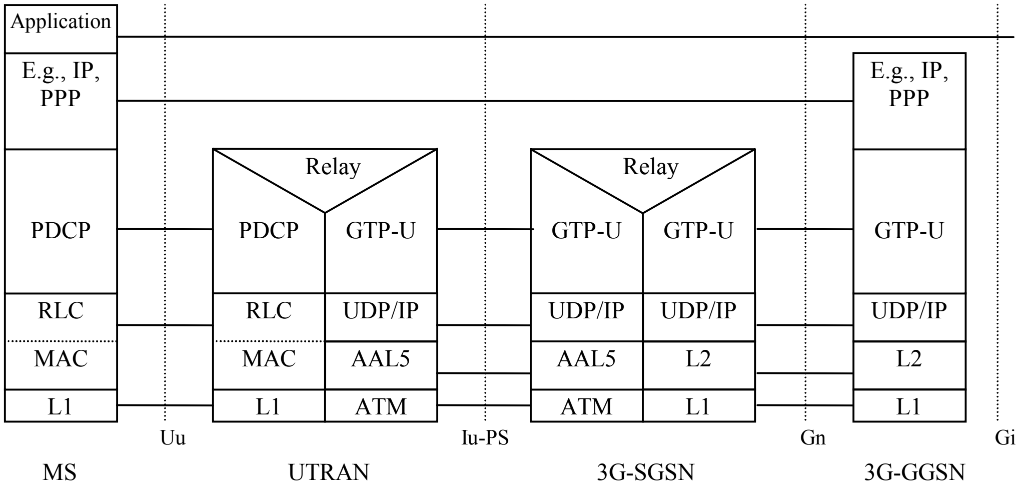


Figure 34: UMTS User Plane [43]

Digital Subscriber Line (DSL) is a service based on the existing copper wire infrastructure of telecommunication companies. DSL is used between the DSL-Modem and the DSLAM while a broadband technology like SONET is used in the access network. ATM may be used for both Access Network and the DSL local loop.

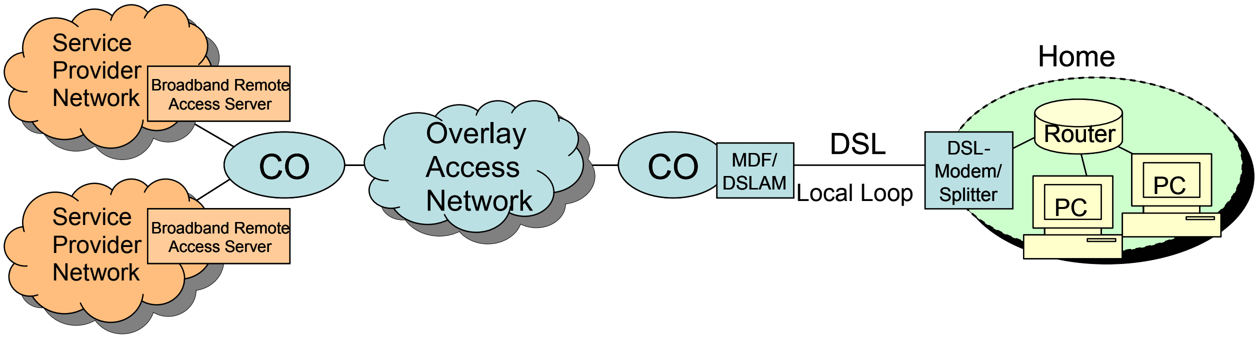


Figure 35: DSL Network Reference Diagram [38]

A DSL network reference diagram is depicted in Figure 35. Here, a Main Distribution Frame (MDF) and a Digital Subscriber Line Access Multiplexer (DSLAM) terminate the DSL connection on the side of a Central Office (CO) which is located within a range of several kilometres from the DSL customer. In a typical ADSL scenario a PC or DSL-Router establishes a PPP connection to a Broadband Remote Access Server. Some encapsulation technologies which can be used with DSL are depicted in Figure 36:

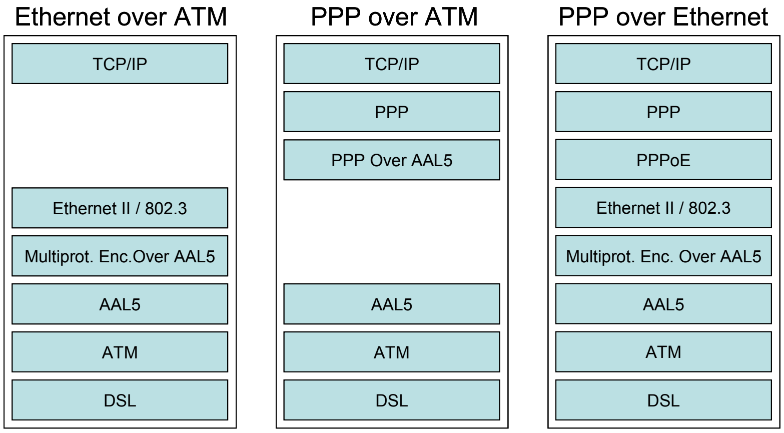


Figure 36: Selected Encapsulation Technologies for IP over DSL [37]

The local loop of DSL is seen as a substitute for the air interface of UMTS while the ADSL access network is similar to the UMTS core network.

In the following test scenarios an ADSL link of Netcologne, a German local telecommunication company and network provider, was used. The provider guaranties 1024kbit/s downlink and 128kbit/s uplink for user data.

The University of Bonn is connected to the network of Netcologne via the DFN in Cologne. The Architecture is depicted in Figure 37.

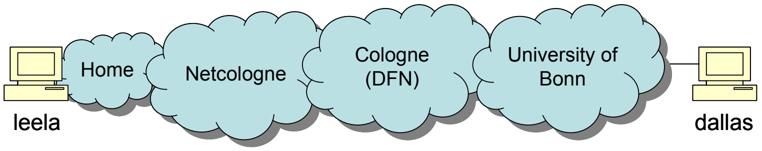


Figure 37: Networks traversed by Packet Pairs

A traceroute plot between `leela located behind` the DSL subnet and `dallas` in the domain of University of Bonn is shown in the following:

```
MR/MN  0  leela                              [192.168.2.171]
MR     1  dsl-router                         [192.168.2.1]
AR     2  erx-maw1.netcologne.de             [195.14.247.95]
       3  swrt-maw1-g34.netcologne.de        [213.196.239.169]
       4  cat6509-pg1-vl200.netcologne.de    [195.14.195.145]
       5  rtint2-g100.netcologne.de          [195.14.247.202]
       6  ir-koeln1-po4-0-0.g-win.dfn.de     [188.1.58.5]
       7  ar-koeln3.g-win.dfn.de             [188.1.84.5]
       8  kr-bonn.rhrz.uni-bonn.de           [131.220.254.2]
       9  sr1-rz-vlan3.rhrz.uni-bonn.de      [131.220.1.249]
      10  cr-rz-po20.rhrz.uni-bonn.de        [131.220.1.57]
      11  ar-vb5-po15.rhrz.uni-bonn.de       [131.220.1.33]
      12  rhenus-router.cs.uni-bonn.de       [131.220.6.3]
HA/CN 13  dallas                             [131.220.6.184]
```

Host `leela` corresponds to a nested MR or a node inside the IVAN and the host `dallas` corresponds to a HA or a corresponding node.

Video Streaming Scenario

The adaptive video on demand application demonstrates the use of a unicast service over different, abrupt changing link characteristics. In the context of the OverDRiVE Project an application has to deal with a heterogeneous multiradio environment with potentially hidden radio links. The user is able to change the quality of a video without restarting the video transmission and therefore to produce different amounts of traffic, according to the available bandwidth. As a further enhancement the client application was able to make its own decision about which quality level was appropriate, adapting to the changing network characteristics.

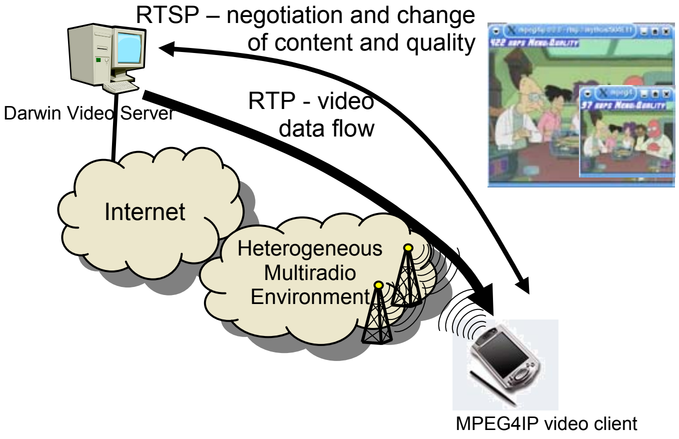


Figure 38: Adaptive Video Streaming

Figure 38 shows an overview of the video streaming setup at the HyWin workshop. A detailed description of this setup is included in section 4 of D14 [3]. The setup represents a small scale OverDRiVE mobile network that moves between locally available wireless LAN hotspots, GPRS and UMTS. This real world scenario was used to demonstrate and validate the applicability of packet pair measurements.

In the context of validation of traffic management, the video streaming solution is also used on a link local basis, avoiding drawbacks caused by the 6bone, to demonstrate measurement accuracy for different link speeds while sending a video stream. The video server and client were separated only via switches. No other IP instances were present. However, also sub-IP layer store and forward boxes may have an influence on capacity and packet pair measurements as described in section 3.3.2.2.

4.4 Measurements

We performed some handover measurements with our testbeds. To understand the results of the IP handover measurements we introduce a taxonomy about mobile node initiated IP handovers.

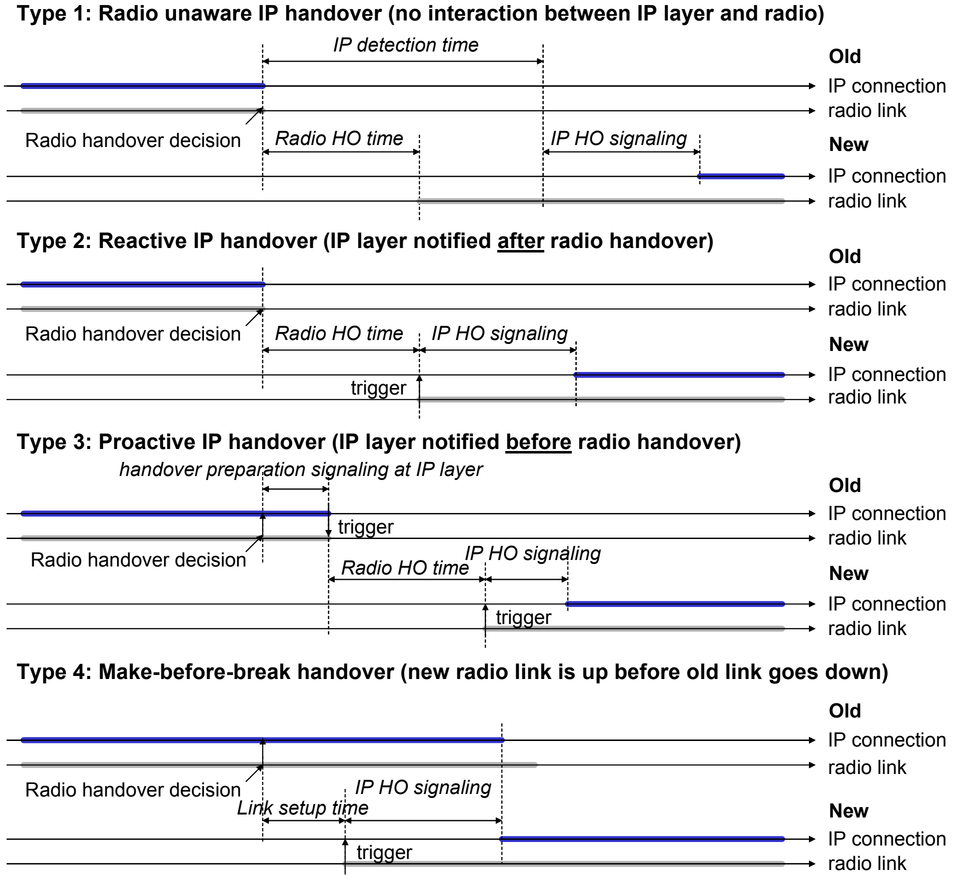


Figure 39: IP handover taxonomy

In case of radio unaware IP handover (type 1) no interaction takes place between the radio and the IP layer. Performing this handover needs a lot of time, because the IP layer has to detect that the old connection is lost and the new one is up without any help from the radio. The radio unaware IP handover can be regarded as a handover with the original layer 3 movement detection, it has a very poor performance and it can be used for device portability only.

In case of reactive IP handover (type 2) the IP layer is notified with a trigger after the radio handover. Due of this trigger the IP layer can perform the handover instantly after the new radio link is up. Such handovers are much faster than radio unaware handovers. This type of handover is called "unplanned handover" in BCMP.

In case of proactive IP handover (type 3) the IP layer is notified about the radio handover beforehand and can perform preparations before the radio link goes down. After the IP layer is ready with the preparations it notifies the radio that the handover can be performed. After the new radio link is up the radio triggers the IP layer telling that it can continue the IP handover through the new radio link. This type of handover could be performed with less disruption in user sessions than the previous one. Type 3 handover is called "planned handover" in BCMP. At the IETF the reactive IP handover is the typical assumption for basic IP mobility protocols and the proactive IP handover for "fast handover protocols".

The make-before-break IP handover (type 4) needs special requirements from the radio technology. It is necessary that the node that wants to perform a handover between two access points hears both access points at the same time. In this case IP layer handover is performed parallel with the radio handover thus resulting in a very fast switch between the old and the new link. The make-before-break IP handover has very good performance, but puts special requirements on radio.

4.4.1 Mobile Router

We performed measurements in the two implementations. In the followings we present the results of our experiments.

4.4.1.1 Measurements in the implementation based on Livsix

During the HyWiN Workshop [66] at Rai Crit in Turin, Italy, December 2003, several demonstrations of the OverDRiVE concepts were performed to a large audience of researchers, engineers and decision makers from Industry, Academia and the European Commission. A common integrated demonstrator involving a DaimlerChrysler research vehicle, a RAI fixed network infrastructure connected to the IPv4/IPv6 Internet and several wireless access systems (GPRS, UMTS, DVB-T) were used.

As part of the common demonstrator, several measurements were performed on-site in order to materialize and evaluate the concepts (software and protocols) developed within the project. In the next section we briefly introduce the main architecture of the common OverDRiVE demonstrator, measurements relevant to the benefits of Route Optimization and, finally, measurements of the behaviour of GPRS and UMTS wireless access systems from an IP communication standpoint.

Architecture of the Overall Demonstrator

The figure below depicts a schematic overview of the common demonstrator; a complete description of the entire demonstrator can be found in Deliverable D14 [3]. We present here only the entities necessary to introduce the RO-related measurements and the IP-over-UMTS/GPRS measurements.

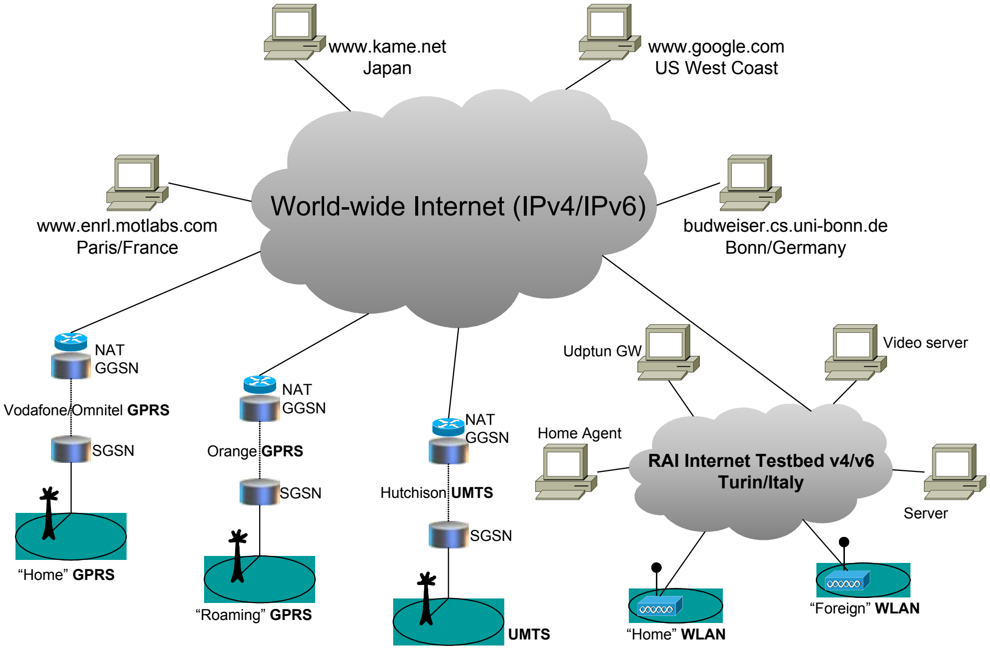


Figure 40: Overall Architecture of the Common Demonstrator

In short, the demonstrator includes the RAI Internet Testbed (bottom right of the figure) supporting both IPv4 and IPv6 which is connected to the IPv4/IPv6 Internet; this Testbed includes several infrastructure servers, such as a Home Agent, a Video Server and the Udptun Gateway. The research vehicle was (that supports an IPv6 moving network) is connecting to five wireless access systems: Vodafone GPRS, Orange GPRS, Hutchison UMTS, Home WLAN and Foreign WLAN (wireless access systems are pictured at the bottom of the figure).

All wireless access systems were used at the same physical location (Rai Crit, Turin, Italy). The "Home" GPRS Access System is used with a Vodafone/Omnitel SIM card that has a local (Italian) subscription. The "Roaming" GPRS Access System is used with a Orange SIM card that has a remote subscription (France); the Orange SIM card is thus used in Italy as "roaming". The difference between the two GPRS Access Systems can be noted in terms of the IP distance between the Udptun GW (placed in the RAI Test-bed) and the exit points of these Access Systems; for example, the distance Udptun GW to Vodafone GGSN is 9, while the distance between Udptun GW and Orange GGSN is 20.

The main demonstration was performed by having the clients within the vehicular moving network to access various services in the RAI Test-bed infrastructure (video streams, vehicular software update) and clients within the RAI Test-bed to access servers deployed within the vehicular moving network. The main mobility aspect of the demonstrator was that the successive attachments of the moving network to the wireless access systems were performed in such a manner that applications were not interrupted. Additional applications performed within the common demonstrator were the "normal" Internet applications like web browsing (google, kame, enrl.motlabs.com). The vehicular moving network used IPv6 exclusively, so, when browsing the web, we used an http v6-to-v4 proxy deployed at budweiser.cs.uni-bonn.de. Finally, since the GPRS and UMTS wireless access systems only offer NAT private IPv4 addresses, we used "FrontBoxes" between the vehicular Mobile Router and the respective Base Station (FrontBoxes are not pictured but see D14 for description of the FrontBoxes); FrontBoxes maintain IPv4 UDP tunnels through the wireless Access System up to the Udptun GW deployed within the RAI Test-bed Infrastructure (IPv6 packets are encapsulated within these UDPv4 tunnels).

The in-vehicular Mobile Router and the Home Agent placed in the RAI infrastructure are running the Open Source LIVSIX IPv6 stack for mobility environments. The FrontBoxes and the Udptun Gateway are running Motorola Labs proprietary software for automatic v6-in-UDPv4 tunnelling. In addition, ppp software and chat scripts developed by RAI and Motorola Labs are used on the dedicated UMTS FrontBox.

Measuring the RO Benefits

In order to evaluate the benefits of using Route Optimization techniques we identified the IP path lengths between various mobility entities. In summary, the following path lengths were obtained (measurements using the traceroute command, see section B).

The measurement files can be found in the appendices.

Intuitively, the Route Optimization techniques benefit the LFN-CN communication only if the following holds:

Dist(CN, LFN) << Dist(LFN, HA) + Dist(CN, HA)

Noting Dist(CN, LFN) as D1, Dist(LFN, HA) as D2 and Dist(CN, HA) as D3, the above statement can be formulated as: RO techniques are useful only if the value V=D1/(D2+D3) tends to 0. V can have any value between 0 and 1 (it will never be either 0 or 1). When V tends to 1, using RO techniques does not bring a useful reduction in the communication path. The more V approaches 0, the more useful RO is.

Let us particularize the V equation for the case when MR is in the foreign WLAN network and LFN communicates to the Video Server; V=3/5.

Let us particularize the V equation for the case when MR is in the foreign WLAN network and LFN communicates to the KAME server; V=14/16.

Let us particularize the V equation for the case when MR is using the UMTS wireless access system and LFN communicates to the Video Server; V=5/7.

Let us particularize the V equation for the case when MR is using the UMTS wireless access system and LFN communicates to www.kame.net; V=14/16.

None of the above cases justifies the use of RO techniques.

Let us now assume that the HA is placed in the lab in Paris (enrl.motlabs.com) and that the MR is connected on the UMTS wireless access system and LFN communicates to the Video Server; V=5/14.

Let us now assume that HA is placed in the enrl domain and that the MR is connected on the foreign WLAN network and LFN communicates to the Video Server; V=3/11.

Intuitively, the benefits of using Route Optimization techniques become more apparent when larger distances are involved.

One simplifying assumption was to consider that the IPv6 distance between a FrontBox and the Udptun GW is 1; this is, in fact, a value that involves much more IPv4 hops and that can be approximated by the respective IPv4 distance; for example the distance IPv6 between the UMTS FrontBox (used as 1) can be better approximated by the respective IPv4 distance (19). Some computations of the V value may thus change.

Measuring the GPRS and UMTS IP Charachteristics

We have used the ping tool to measure various characteristics of the GPRS and UMTS wireless access systems. All result files are copied in the ending sections of this document.

Main conclusions of these measurements can be summarized as:

- Both GPRS and UMTS links are asymmetric links with a higher MTU size on the downlink channel; GPRS has a smaller MTU size than UMTS; these links are adapted to the classical use of Internet such as browsing: user sends HTTP requests on the uplink and receives HTML pages on the downlink; in this operation it is often the case than no IP fragmentation occurs. The asymmetric links are not adapted for the case when a user on the Internet makes HTTP requests to obtain HTML vehicular information data (vehicle connected to the Internet via GPRS or UMTS) since much fragmentation would occur on the uplink channel.
- Obviously, the bandwidth of UMTS link is higher than on the GPRS link.
- The roundtrip time of IP packets on the first IP hop (which includes, but is not limited to, the wireless link between the mobile and the Base Station) depends on the IP packet size, even when the IP MTU size is not involved. On GPRS, the shortest roundtrip time is on the order of 700ms while on UMTS it is on the order of 30ms; the smaller the packets, the shorter the roundtrip time.
- The MTU sizes (Maximum Transmission Unit) for GPRS and UMTS are different and have been measured as being on the order of 1424 (Orange) and 1472 (Omnitel); sending packets larger than these sizes involves triggering of IP Fragmentation and Reassembly. In IPv4, fragmentation and reassembly is performed by the intermediary routers, while in IPv6 this is performed by the end nodes; thus it is possible that there exists a need for an IPv6 end node on the Internet to have knowledge of the fact that it is communicating with a node connected via GPRS or UMTS link.
- Neither GPRS (Orange or Omnitel/Vodafone) nor UMTS (H3G) offer native IPv6 access; these wireless access systems offer private addresses behind NAT.

Orange, Omnitel/Vodafone and H3G connect to the world-wide IPv4 Internet by various firewalling techniques; some ports/protocols are allowed on Omnitel/Vodafone but not on the others. In order to be able to use the UDPv4 tunneling of IPv6 packets we performed port scanning on all three networks.

During the HyWiN Workshop in Turin, Italy, December 2003, several demonstrations of the OverDRiVE concepts were performed to a large audience of researchers, engineers and decision makers from Industry, Academia and the European Commission. A common integrated demonstrator involving a DaimlerChrysler research vehicle, a RAI fixed network infrastructure connected to the IPv4/IPv6 Internet and several wireless access systems (GPRS, UMTS, DVB-T) was exhibited.

As part of the common demonstrator, several measurements were performed on-site in order to materialize and evaluate the concepts (software and protocols) developed within the project. In the next section we briefly introduce the main architecture of the common OverDRiVE demonstrator, measurements relevant to the benefits of Route Optimization and, finally, measurements of the behaviour of GPRS and UMTS wireless access systems from an IP communication standpoint.

Architecture of the Overall Demonstrator

The figure below depicts a schematic overview of the common demonstrator; a complete description of the entire demonstrator can be found in Deliverable D14. We present here only the entities necessary to introduce the RO-related measurements and the IP-over-UMTS/GPRS measurements.

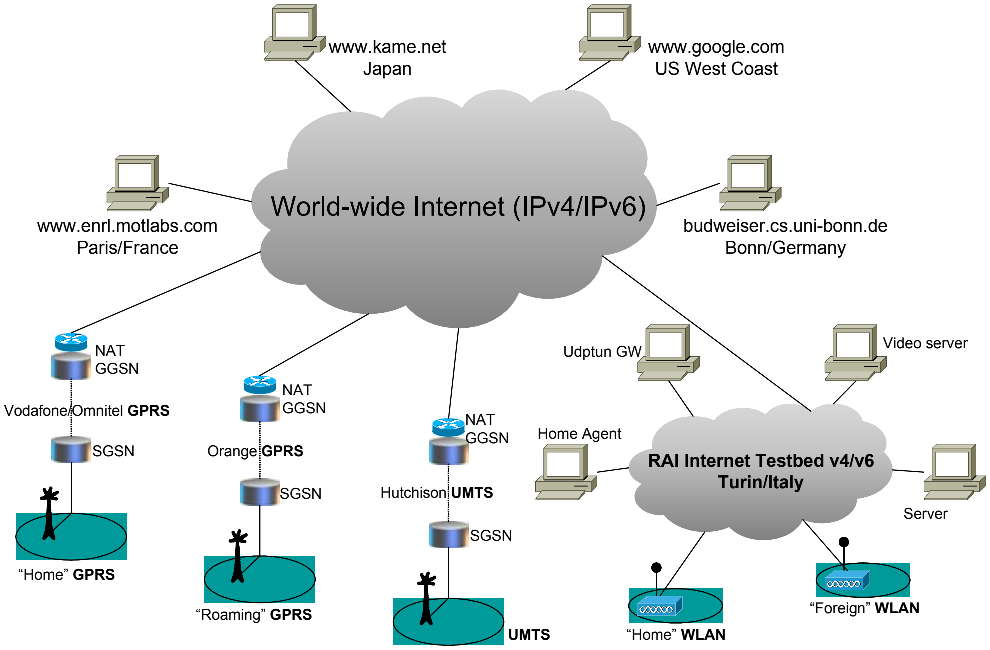


Figure 41: Overall Architecture of the Common Demonstrator

In short, the demonstrator includes the RAI Internet Testbed (bottom right of the figure) supporting both IPv4 and IPv6 which is connected to the IPv4/IPv6 Internet; this Testbed includes several infrastructure servers, such as a Home Agent, a Video Server and the Udptun Gateway. The research vehicle was (that supports an IPv6 moving network) is connecting to five wireless access systems: Vodafone GPRS, Orange GPRS, Hutchison UMTS, Home WLAN and Foreign WLAN (wireless access systems are pictured at the bottom of the figure).

All wireless access systems were used at the same physical location (RAI CRIT, Turin, Italy). The "Home" GPRS Access System is used with a Vodafone/Omnitel SIM card that has a local (Italian) subscription. The "Roaming" GPRS Access System is used with a Orange SIM card that has a remote subscription (France); the Orange SIM card is thus used in Italy as "roaming". The difference between the two GPRS Access Systems can be noted in terms of the IP distance between the Udptun GW (placed in the RAI Test-bed) and the exit points of these Access Systems; for example, the distance Udptun GW to Vodafone GGSN is 9, while the distance between Udptun GW and Orange GGSN is 20.

The main demonstration was performed by having the clients within the vehicular moving network to access various services in the RAI Test-bed infrastructure (video streams, vehicular software update) and clients within the RAI Test-bed to access servers deployed within the vehicular moving network. The main mobility aspect of the demonstrator was that the successive attachments of the moving network to the wireless access systems was performed in such a manner that applications were not interrupted. Additional applications performed within the common demonstrator were the "normal" Internet applications like web browsing (google, kame, enrl.motlabs.com). The vehicular moving network used IPv6 exclusively, so, when browsing the web, we used an http v6-to-v4 proxy deployed at budweiser.cs.uni-bonn.de. Finally, since the GPRS and UMTS wireless access systems only offer NAT private IPv4 addresses, we used "FrontBoxes" between the vehicular Mobile Router and the respective Base Station (FrontBoxes are not pictured but see D14 for description of the FrontBoxes); FrontBoxes maintain IPv4 UDP tunnels through the wireless Access System up to the Udptun GW deployed within the RAI Test-bed Infrastructure (IPv6 packets are encapsulated within these UDPv4 tunnels).

The in-vehicular Mobile Router and the Home Agent placed in the RAI infrastructure are running the Open Source LIVSIX IPv6 stack for mobility environments. The FrontBoxes and the Udptun Gateway are running Motorola Labs proprietary software for automatic v6-in-UDPv4 tunnelling. In addition, ppp software and chat scripts developed by RAI and Motorola Labs are used on the dedicated UMTS FrontBox.

Measuring the RO Benefits

In order to evaluate the benefits of using Route Optimization techniques we identified the IP path lengths between various mobility entities. In summary, the following path lengths were obtained (measurements using the traceroute command, see section xx):

IPv4 distance between GPRS FrontBox (Vodafone) and www.google.com: 23

IPv4 distance between GPRS FrontBox (Vodafone) and Udptun GW: 17

IPv4 distance between Udptun GW and H3G GGSN: 16

IPv4 distance between Udptun GW and Vodafone GGSN: 9

IPv4 distance between Udptun GW and Orange GGSN: 20

IPv6 distance between Udptun GW and Budweiser: 6

IPv6 distance between Udptun GW and www.kame.net: 11

IPv6 distance between Udptun GW and www.enrl.motlabs.com: 5

IPv4 distance between UMTS FrontBox (H3G) and www.google.com: 16

IPv4 distance between UMTS FrontBox (H3G) and Udptun GW: 19

IPv4 distance between www.enrl.motlabs.com and Budweiser: 16

IPv6 distance between www.enrl.motlabs.com and Budweiser: 24

In addition, the following distances were known by topology design (see topology of Common Demonstrator Setup in D14):

IPv6 distance between HA and Video Server (RAI): 2

IPv6 distance between LFN and MR: 1

IPv6 distance between MR and HA (when MR at home): 1

IPv6 distance between MR and Video Server (when MR in "Foreign" WLAN): 2

IPv6 distance between MR and HA (when MR in "Foreign" WLAN): 2

IPv6 distance between HA and Udptun GW: 2

IPv6 distance between MR and Udptun GW when MR in foreign WLAN: 2

The measurement files can be found in the appendices.

Intuitively, the Route Optimization techniques benefit the LFN-CN communication only if the following holds:

Dist(CN, LFN) << Dist(LFN, HA) + Dist(CN, HA)

Noting Dist(CN, LFN) as D1, Dist(LFN, HA) as D2 and Dist(CN, HA) as D3, the above statement can be formulated as: RO techniques are useful only if the value V=D1/(D2+D3) tends to 0. V can have any value between 0 and 1 (it will never be either 0 or 1). When V tends to 1, using RO techniques does not bring a useful reduction in the communication path. The more V approaches 0, RO is more useful.

Let us particularize the V equation for the case when MR is in the foreign WLAN network and LFN communicates to the Video Server; V=3/5.

Let us particularize the V equation for the case when MR is in the foreign WLAN network and LFN communicates to the KAME server; V=14/16.

Let us particularize the V equation for the case when MR is using the UMTS wireless access system and LFN communicates to the Video Server; V=5/7.

Let us particularize the V equation for the case when MR is using the UMTS wireless access system and LFN communicates to www.kame.net; V=14/16.

None of the above cases justifies the use of RO techniques.

Let us now assume that the HA is placed in the lab in Paris (enrl.motlabs.com) and that the MR is connected on the UMTS wireless access system and LFN communicates to the Video Server; V=5/14.

Let us now assume that HA is placed in the enrl domain and that the MR is connected on the foreign WLAN network and LFN communicates to the Video Server; V=3/11.

Intuitively, the benefits of using Route Optimization techniques become more apparent when larger distances are involved.

One simplifying assumption was to consider that the IPv6 distance between a FrontBox and the Udptun GW is 1; this is, in fact, a value that involves much more IPv4 hops and that can be approximated by the respective IPv4 distance; for example the distance IPv6 between the UMTS FrontBox (used as 1) can be better approximated by the respective IPv4 distance (19). Some computations of the V value may thus change.

Measuring the GPRS and UMTS IP Characteristics

We have used the ping tool to measure various characteristics of the GPRS and UMTS wireless access systems. All result files are copied in the Appendices.

Main conclusions of these measurements can be summarized as:

- Both GPRS and UMTS links are asymmetric links with a higher MTU size on the downlink channel; GPRS has a smaller MTU size than UMTS; these links are adapted to the classical use of Internet such as browsing: user sends HTTP requests on the uplink and receives HTML pages on the downlink; in this operation it is often the case than no IP fragmentation occurs. The asymmetric links are not adapted for the case when a user on the Internet makes HTTP requests to obtain HTML vehicular information data (vehicle connected to the Internet via GPRS or UMTS) since much fragmentation will occur on the uplink channel.
- Obviously, the bandwidth of UMTS link is higher than on the GPRS link.

- The roundtrip time of IP packets on the first IP hop (which includes, but is not limited to, the wireless link between the mobile and the Base Station) depends on the IP packet size, even when the IP MTU size is not involved. On GPRS, the shortest roundtrip time is on the order of 700ms while on UMTS it is on the order of 30ms; the smaller the packets, the shorter the roundtrip time.
- The MTU sizes (Maximum Transmission Unit) for GPRS and UMTS are different and have been measured as being x and y respectively; sending packets larger than these sizes involves triggering of IP Fragmentation and Reassembly. In IPv4, fragmentation and reassembly is performed by the intermediary routers, while in IPv6 this is performed by the end nodes; thus it is possible that there exist a need for an IPv6 end node on the Internet to have knowledge of the fact that it is communicating with a node connected via GPRS or UMTS link.
- Neither GPRS (Orange or Omnitel/Vodafone) nor UMTS (H3G) offer native IPv6 access; these wireless access systems offer private addresses behind NAT.

Orange, Omnitel/Vodafone and H3G connect to the world-wide IPv4 Internet by various firewalling techniques; some ports/protocols are allowed on Omnitel/Vodafone but not on the others. In order to be able to use the UDPv4 tunnelling of IPv6 packets we performed port scanning on all three networks.

4.4.1.2 Measurements in the implementation based on MIPL

We performed type 1 and type 4 IP handover and 3G measurements in our moving network testbed. We will show that the radio unaware IP handover (type 1) performs really poor and the make-before-break IP handover (type 4) performs very well with respect on handover outage time. Our experiments also showed that 3G provides the necessary bandwidth for real time video applications.

Specification of the testbed and the software used for the measurements

The testbed consists of the following equipment: the correspondent node, the local fixed node and the visiting mobile node are P4 1.8 GHz laptops, the VGGSN is a P1 133 MHz PC and all other network elements (HA, AR1, AR2, BAR1, BAR2 and MR) are AMD Athlon XP 1800+ PCs. The network cards in the PCs are Intel EtherExpress Pro 10/100 Ethernet cards and the wireless cards are AVAYA Orinoco Silver 802.11b PCMCIA WLAN cards. All of the computers were equipped with the same Linux distribution (Debian Sarge) and kernel (2.4.20). For the implementation of our mobile router prototype we took version 0.9.5.1 of the MIPL stack.

During our experiment we applied similar programs and methods for the measurements as described in [18]: we used *tcpdump* and an own test environment. The environment consists of a packet sender and a packet receiver program. The packet sender, which is placed on the correspondent node, emits 65-byte-long UDP packets at a pre-defined rate. We chose the UDP transport protocol instead of TCP so that we can avoid the effect of TCP's traffic control mechanisms. The receiver, which is running on the local fixed node behind the mobile router, logs the packet loss and whether the sequence of the packets swapped. With *tcpdump* we logged at the LFN the inter-arrival times of the incoming UDP packets. In the experiments we tuned the radio interfaces of the access routers and the mobile router to the same channel.

IP handover measurements

First we measured the outage time of the mobile router's handover, when the access router, which the mobile router was attached to, suddenly disappears. We sent UDP packets in every 10 milliseconds and by pulling down the air interface of the access router, which the MR was connected to, in every 25 seconds we forced the MR to perform a handover. The histogram of the handover outage times can be seen in Figure 42. In the figure on the X axis we can see the intervals of the handover outage times and on the Y axis we can see how many handovers were performed in the given outage time interval during the experiment. The mean value of these measurements was 4080 ms, which means that after the old access router disappeared the mobile router needed 4.08 seconds in average to find and connect to the other WLAN access router. The histogram shows that 77% of the handover outage times were between 3100 and 5500 ms and 57% were between 4000 and 5500 ms. The smallest value we measured for the handover outage time was above 1.1 second, which is also a quite long delay. We can observe that performing handovers without any radio information (trigger) leads to very poor handover performance and causes a long disruption in the user sessions.

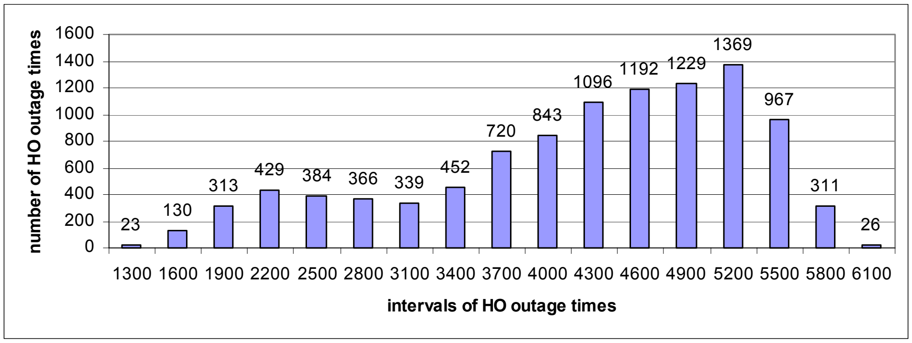


Figure 42: Histogram of handover outage time when AR disappears (radio unaware - type 1 - IP handover)

In the next experiment we turned on the radio interfaces of both access routers. This way the mobile router could hear both access routers at the same time, thus it could perform a make-before-break (type 4) handover. Unfortunately, it was not possible to use the same method for handover measurements as in the previous experiment, this time we measured the packet loss during handovers. We sent UDP packets in every 2 milliseconds and we performed a handover with the MR in every 2 seconds. We measured 0.7% packet loss in average, thus we could say that the outage time was about 1.4 ms (in every 2 seconds 0.7% of the time is spent with the handover). In this experiment packet loss can occur if a packet arrives at the same time when the routing tables are being updated and because of this the packet cannot reach the mobile router. As we can see this type of handover can be performed very fast. During this handover users cannot perceive any kind of disturbance, for example, in their streaming video application.

The comparison of the different IP handover types can be seen in Figure 43. The measurements of the reactive (type 2) and the proactive (type 3) IP handover outage times were taken from [18], where the measurements included the handover preparation as well (this is why the proactive handover needs more time).

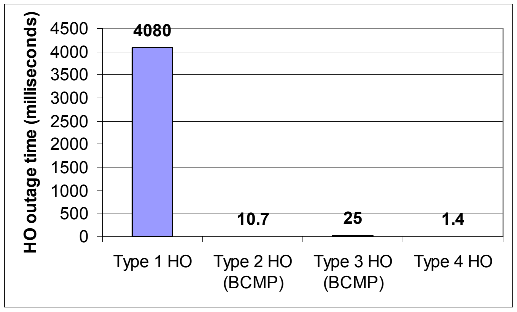


Figure 43: Comparison of IP handover outage times (mean values)

The measurements showed that to perform a switch between two access routers the radio unaware (type 1) IP handover needs much more time (4080 milliseconds) than all other IP handover types (reactive, proactive and make-before-break) that are someway aware of the radio handover. The make-before-break IP handover caused only a short break in the user’s session (1.4 milliseconds). The reactive (type 2) and the proactive (type 3) handovers would result somewhere between type 1 and type 4, but much more closer to type 4. According to the measurements with BCMP reactive and proactive handovers could be performed around 10.7 ms and 25 ms respectively. This shows that it is crucial to take some information from the radio (e.g., triggers) into account, because it helps making IP handovers much faster.

UMTS/WCDMA measurements

We performed some measurements through a publicly available UMTS/WCDMA network. We investigated the delay by sending ping packets from a local fixed node located inside the moving network to the home agent of the mobile router and the throughput by downloading a file from the Internet to a local fixed node. According to our measurement the average delay through the UMTS/WCDMA network was 220 ms and the average bandwidth was 310.31 kbit/s, which can be regarded as very good.

We also tested the UMTS/WCDMA network with a video application. We sent real time streaming video through the 3G network from the correspondent node to the mobile node behind the mobile router and we could see that there was no (or only rarely and negligible) disturbance in the picture of the video at the receiver mobile node.

4.4.2 Traffic Management

The accuracy of packet dispersion techniques is influenced by many different factors. In the first section an analytical model is presented that estimates the efficiency of a link for different packet sizes. This model is then used to verify the packet pair measurements presented in the following subsections. Different kinds of influences on packet pair measurements are discussed and examined in the context of the scenarios presented in section 4.3.3.

4.4.2.1 Packet Size – Bandwidth Dependence: Analytical Model

The size of probing packets may have both impact on the influence of cross traffic to the packet pairs [42]as well as on the link capacity. Smaller packets have a lower efficiency than bigger packets if the header size is constant. In this subsection a mathematical model is presented which makes it possible to calculate the capacity of an edge (i.e. link - see section 3.3.2.2) on different layers (i.e. for different protocol stacks). This is the basis for calculating path capacity.

Let $H_P(x)$ be the size in bytes of a protocol data unit (PDU) of protocol P as a function of the service data unit (SDU) size x in bytes. The normalised efficiency $E : N \to [0;1]$ introduced by protocol header information up to layer M can then be expressed as ratio of the size of the service data unit and the concatenation of $H_P(x) : N \to N$:

$$E_{L_M} = x \Big/ H_1 \circ H_2 \circ \ldots \circ H_M(x)$$

Figure 45 depicts the link layer overhead of different transmission and encapsulation. While the lower layer protocols used in the video streaming scenario are known (only Ethernet II), the ADSL local loop and the access network are in the domain of the network provider and the exact protocol stacks used are not known previous to the experiments. However candidate protocols are introduced.

A MAC layer Ethernet II frame consists of 18 bytes of header information and 46 bytes – 1500 bytes of user data (preamble and inter frame gap are considered to belong to the physical layer). Some other protocols might be of interest when examining the ADSL scenario are PPPoE, PPP, AAL5, ATM and RFC1483. A PPPoE (RFC2516) add 6 bytes and the PPP header another 2 bytes. The preamble and the inter frame spacing of Ethernet is not considered here, because it is not used with link layer encapsulation. For small MTU sizes additional padding further abates efficiency.

As ADSL might also use ATM, ATM cell are also considered. They consist of 5 bytes of header information and 48 bytes of user data. Larger blocks of data have to be spitted into several ATM cells. Padding to full 48 bytes adds additional overhead. When using ATM Adaptation Layer 5 (AAL5) which emulates a random access broadcast medium 8 bytes are added to the end of each payload. To support encapsulation of Ethernet frames in AAL5, Multiprotocol Encapsulation over AAL5 (RFC1483) may be used which adds another 6 bytes of header information.

$H_{\text{Protocol}}(x)$ = Header [bytes] + Padding [bytes] + SDU [bytes]			
Protocol	**Header [bytes]**	**Padding [bytes]**	**SDU [bytes]**
Ethernet II	18	$\delta_{\lvert 46-x \rvert, 46-x} \cdot (46-x)$	x
PPPoE	6	0	x
PPP	2	0	x
ATM	$\lceil x/48 \rceil \cdot 5$	$48 - x \bmod 48$	$\lceil x/48 \rceil \cdot 48$
AAL5	8	0	x
RFC1483	6	0	x

Table 3: PDU Size as Function of SDU

Note that overhead due to retransmissions and separate protocol control packets is not considered due to simplicity reasons. All functions describe the PDU size correctly up to the MTU of the protocol. However, $H_{\text{ATM}}(x)$ also models fragmentation. Therefore, all functions model protocol behaviour for payload sizes between 1 and 1500 bytes. $\delta_{k,l} = \begin{cases} 1 & for \quad k = l \\ 0 & for \quad k \neq l \end{cases}$ is the Kronecker symbol and models, if padding has to be applied or not.

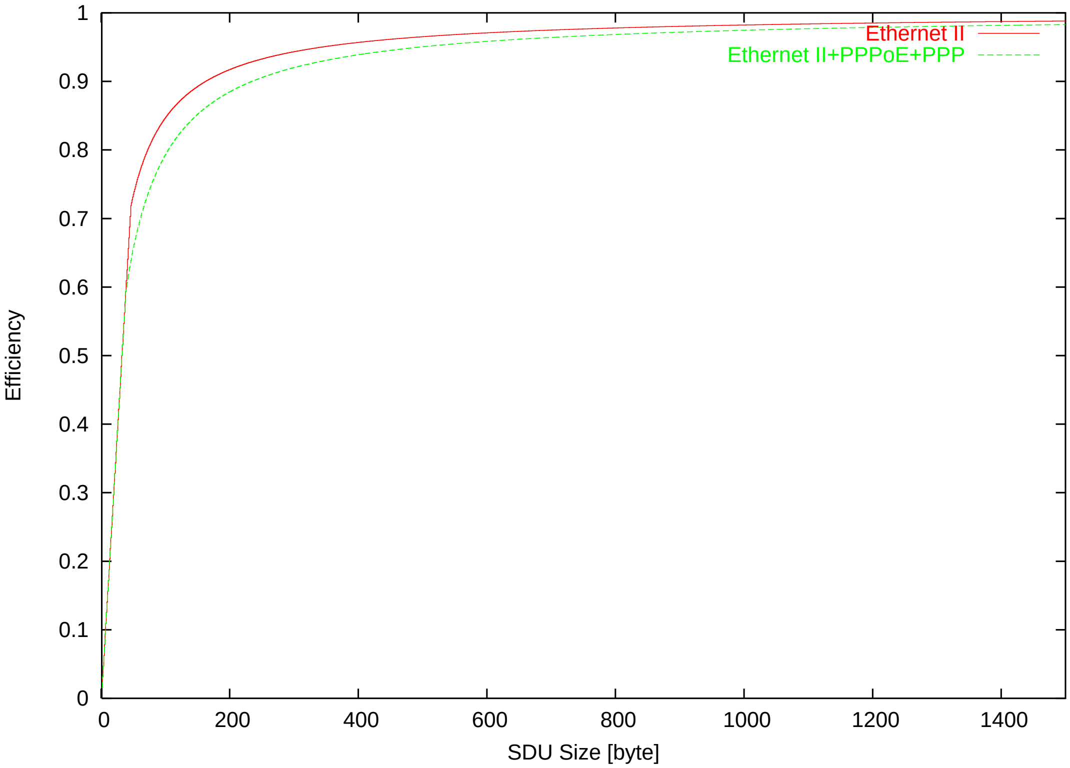


Figure 44: Efficiency of Link Layer Header of Ethernet II with and without encapsulation

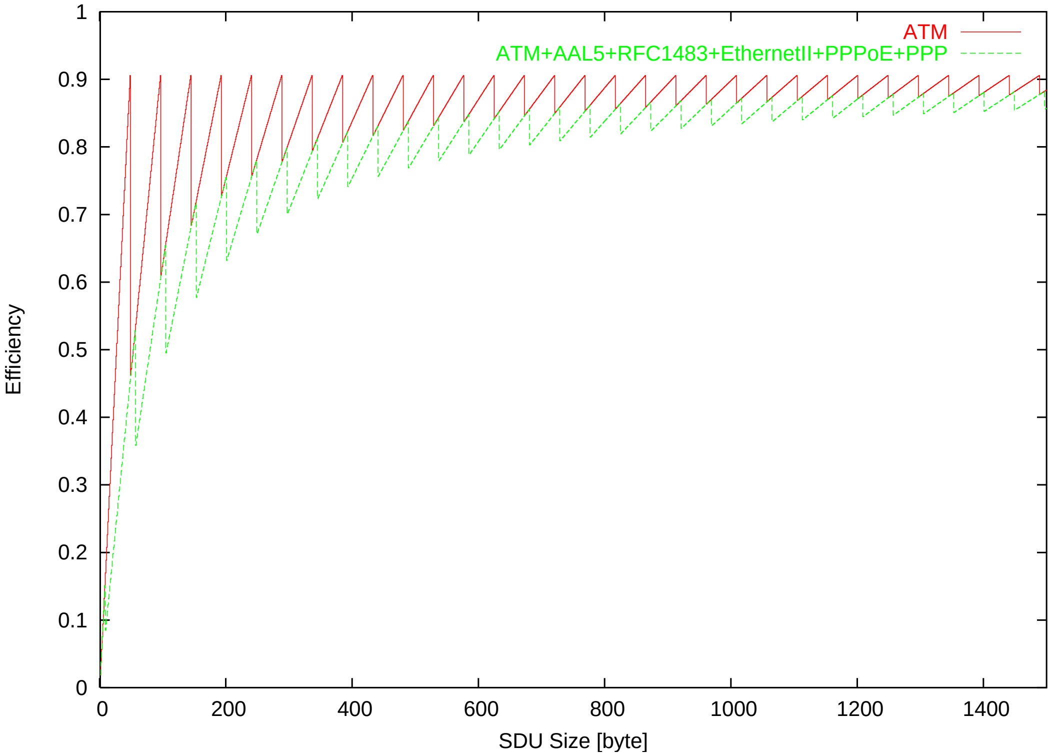


Figure 45: Efficiency of Link Layer Header of ATM with and without encapsulation

Figure 44 and Figure 45 visualise the efficiency E_{L_M} for different protocol stacks in the interval $[1:1500]$ bytes which may be considered as a typical range of IP packet sizes. Due to padding, the efficiency of Ethernet II in conjunction with PPPoE and PPP increases linearly up to 38 bytes and equals the efficiency of Ethernet II while the efficiency of Ethernet II alone stays linear up to 48 bytes payload. Further increasing packet size the slope of the curves decrease. Ethernet II has a maximum efficiency of about 0.99.

The efficiency of ATM alone and ATM in conjunction with AAL5, RFC1483, Ethernet II, PPPoE and PPP is depicted on the right side of Figure 45. The saw tooth characteristic of both curves is due to fragmentation of the SDU onto several ATM cells. AAL5, RFC1483, Ethernet II, PPPoE and PPP all add a constant amount of bytes per SDU in the specified payload interval. Using ATM the maximum of efficiency is about 0.91.

4.4.2.2 Packet Size – Bandwidth Dependence: Packet Pair Measurements

The scenario for packet pair measurements has been described in section 4.3.3.2. The DSL downlink used had a bandwidth of 1024 kbit/s and an uplink of 128 kbit/s guaranteed by the provider. From `leela` to `dallas` and backwards 50 packet pairs for different IP packet sizes were sent. The packet sizes ranged from 60 byte to 1460 bytes. A selection of packet pair separation measurements at the host leela is depicted in Figure 46 using Box and Whisker Plots. The horizontal line within a box is the statistical median of measurements. The box itself comprises the inner quartiles. The whiskers extend to the farthest measurements which are not outliners, i.e. lie outside a range of ¾ times the inner quartile range from the end of a box. Nearer outlines are marked with x, outliners even farther away are marked with o.

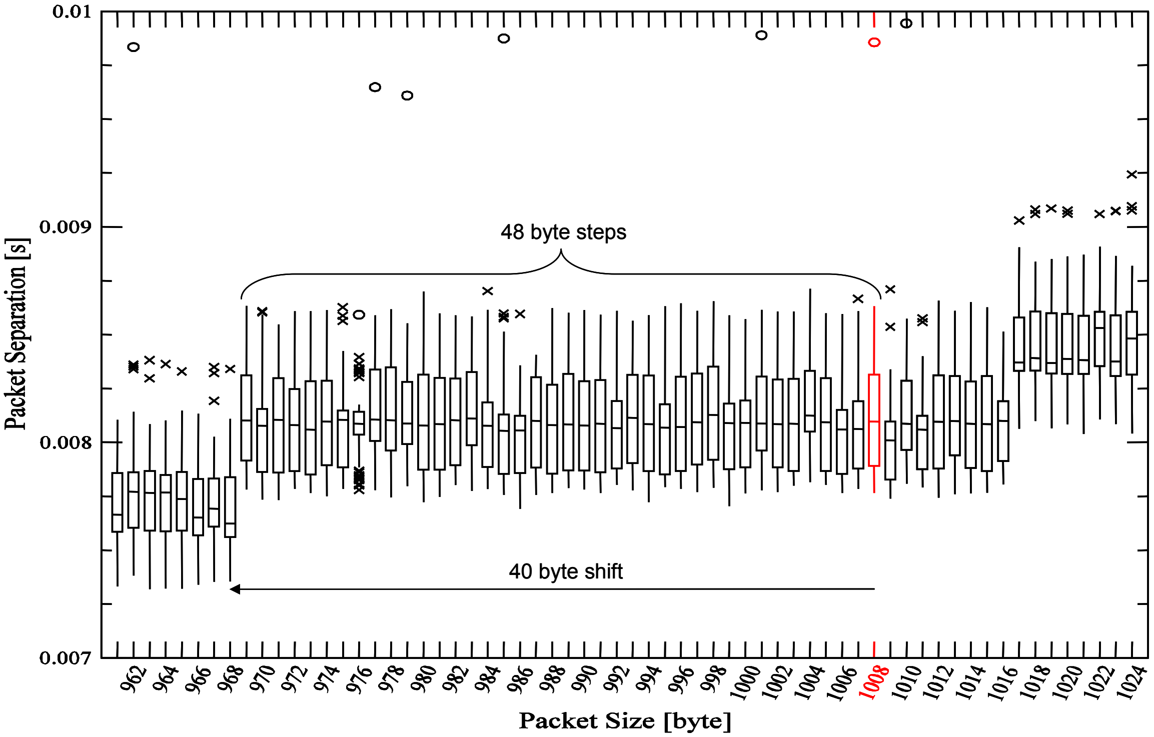


Figure 46: Boxplot of Packet Pair Separation if different Packet Sizes on DSL-Downstream

The stepping of packet separation for different IP packet sizes continues over the whole spectrum of packet sizes as well for the downlink as for the uplink. Every step has a width of 48 bytes which matches exactly the ATM cell's SDU. The largest packet size of a step corresponds to a complete exploitation of all ATM cells used, needing no padding. However, every step is shifted for 40 bytes, i.e. packet sizes 39 bytes smaller than the corresponding size dividable by 48 bytes already show a larger packet separation and therefore belong to the next step. This means that in addition to the IP packet 40 additional bytes are transported in the ATM cells.

Taking into account the different protocol stacks to be used with DSL (some were already depicted in Figure 36 in section 4.3.3.2), the protocol stack architecture shown in Figure 47 has likely been used.

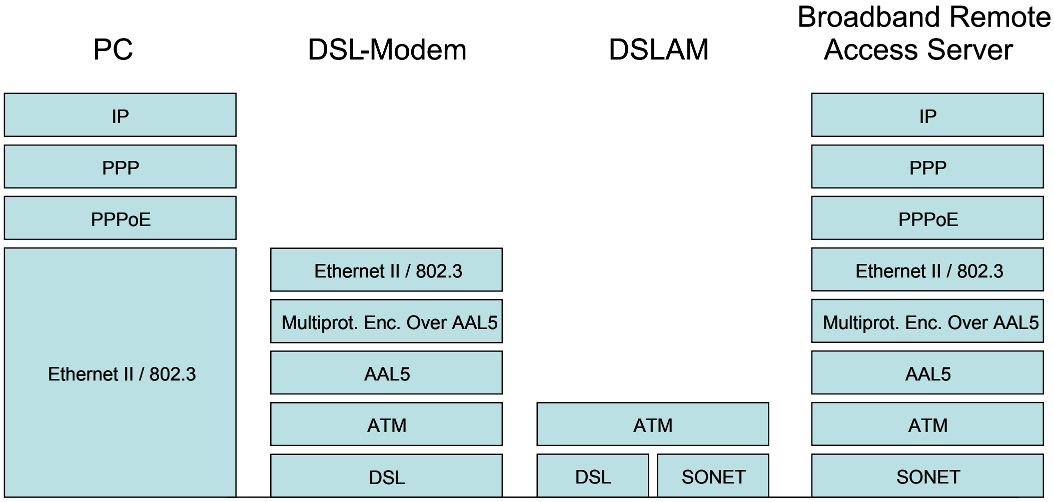


Figure 47: Protocol Stack probably used by DSL-Provider

Adding all protocol header overhead between ATM and IP layer produces exactly 40 bytes of header overhead.

The information gathered about the protocol stack of DSL can be used as an input for the analytical model described in the previous subsection to endorse the results of the packet pair measurements.

4.4.2.3 Comparison of Analytical Model and Packet Pair Measurements

In the previous subsections 4.4.2.1 and 4.4.2.2 packet size – bandwidth dependencies were discussed both from an arithmetical point of view and based on measurements taken from a DSL-1024 downlink. This subsection is dedicated to comparing the results of both subsections and thus evaluating both approaches. Note that outliners of the box plots in this subsection are omitted.

Figure 48 depicts bottleneck measurements and the results of the analytic model. From the analytic point of view the achievable bottleneck bandwidth for IP packets is determined by the efficiency E of the protocol stack and the nominal bandwidth of the bottleneck link. As founded in section 4.4.2.2, a possible protocol stack is ATM-AAL5-Multiprotocol Encapsulation over AAL5-EthernetII-PPPoE-PPP which leads to an efficiency E depicted in Figure 45 on the right side. This efficiency has to be multiplied with the nominal link bandwidth. The DSL-modem was configured with a nominal downlink bandwidth of 1152 kbit/s.

Packet Pair measurements consisted of 50 Packet Pairs for every second local maximum and minimum of the analytic curve.

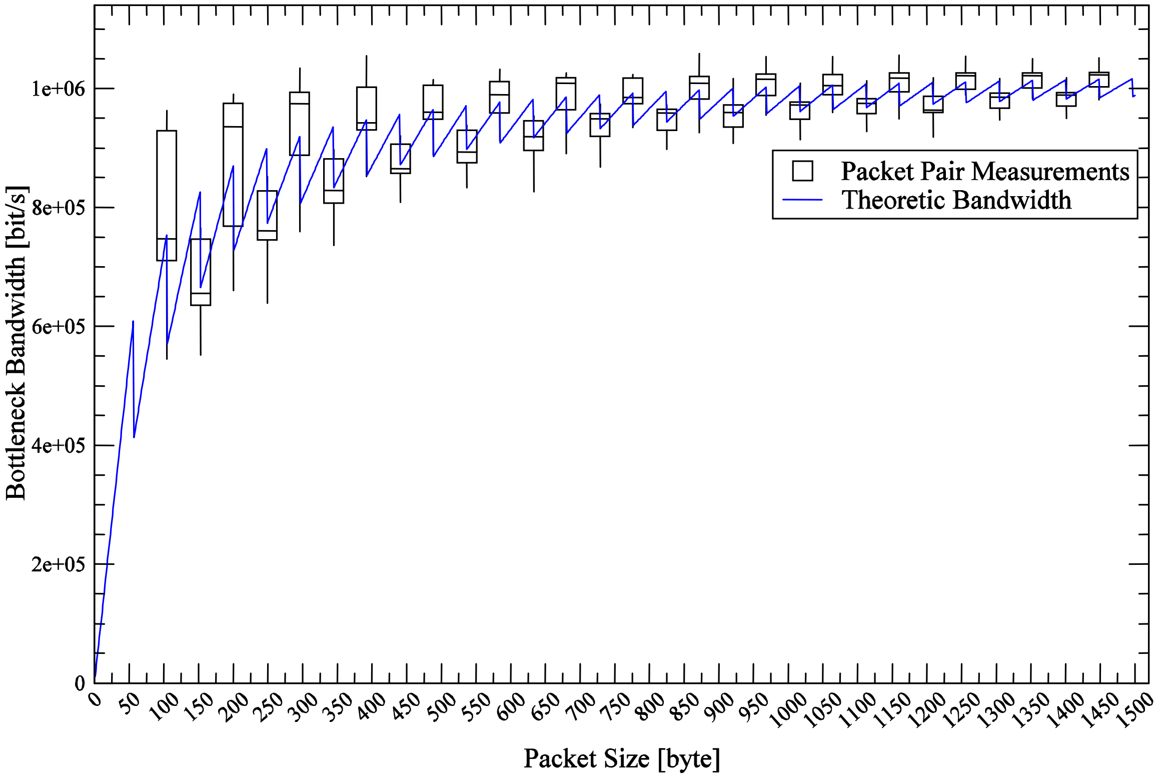


Figure 48: DSL 1024 kbit/s Downlink - Comparison of Analytic and Packet Pair Approach

As the reader can see the theoretic bandwidth from the analytic model and the packet pair measurements both share the saw tooth characteristic. In addition both approaches lead to the same result. All analytical values lie in the box of the corresponding packet pair measurement and therefore in the inner two quartiles. In most cases even the median of the packet pair measurement and the analytic value are nearly identical.

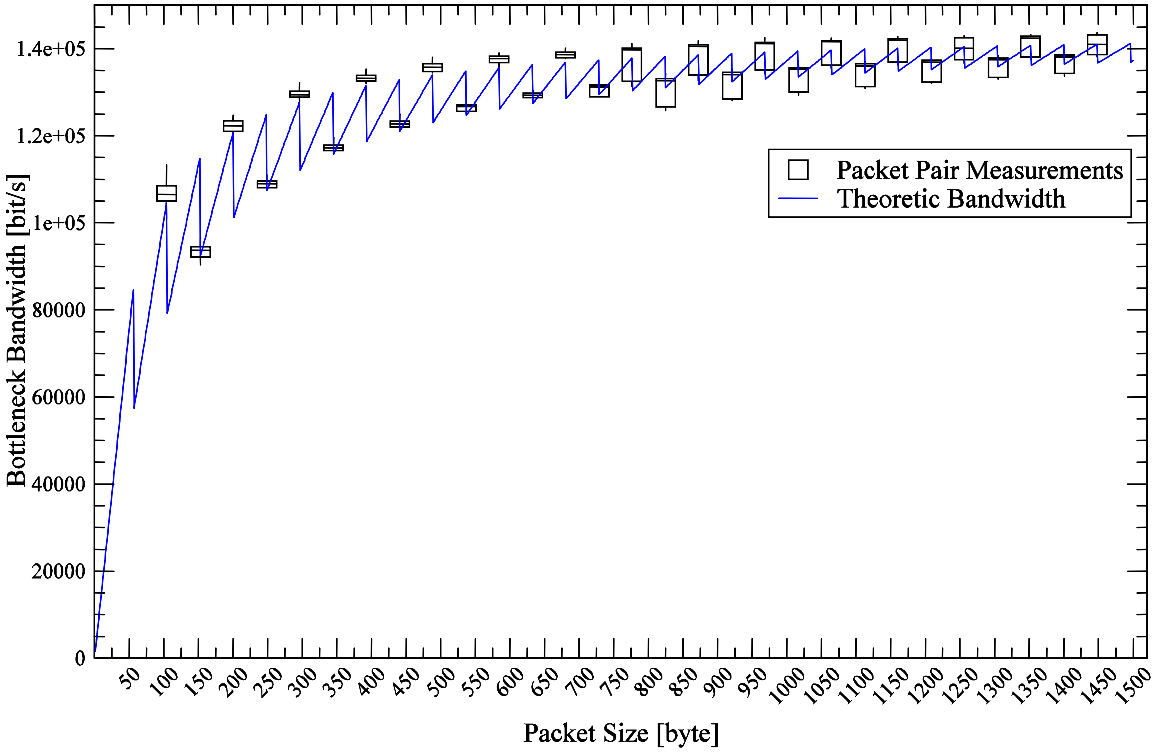


Figure 49: DSL 128 kbit/s Uplink - Comparison of Analytic and Packet Pair Approach

For the uplink the DSL-modem was configured with 160 kbit/s nominal bandwidth. **Figure 49** includes both the theoretic values and the packet pair results. Again the analytic model matches the results of the packet pair measurements. For large packets the bandwidth is higher than 128 kbit/s. ADSL links can only be adjusted in 32 kbit/s steps [35] because of the Reed-Solomon forward error correction used. The next smaller nominal link bandwidth 128 kbit/s would be to low. Note that Figure 53 and Figure 55 have different scale.

These results strongly endorse both the interpretation of packet pair measurements and the analytic model.

4.4.2.4 Cross traffic Influences on packet pair measurements

Packet pair measurements are influenced by many factors; among them cross traffic before the bottleneck link, cross traffic after the bottleneck link and the accuracy of the measurement process which is directly related to packet size and bottleneck bandwidth.

Regarding the first two influences, the location of the bottleneck link may be of importance. D03 [1]stated that the bottleneck link is supposed to be the radio links of one of the mobile routers and therefore is probably located near the sender or the receiver of packet pairs respectively. Cross traffic before the bottleneck link will only affect measurements if it is delaying a second packet more than the bottleneck link will do. These effects result in an underestimation of the bottleneck bandwidth by some measurements. This part of the distribution of measurements is referred to as Sub-Capacity Dispersion Range (SCDR) in [52]. Delaying the first packet may only influence packet pairs after the bottleneck link, masking their bottleneck separation and leading to Post-Narrow Capacity Modes (PNCMs) [52]. As a consequence packet pair measurements on the HA or CN may suffer from the SCDR more severe than measurements on the MR or MN.

The DSL scenario described in section 4.3.3.2 was supposed to verify this theory. Figure 50 and Figure 51 show some results of measurements in this scenario.

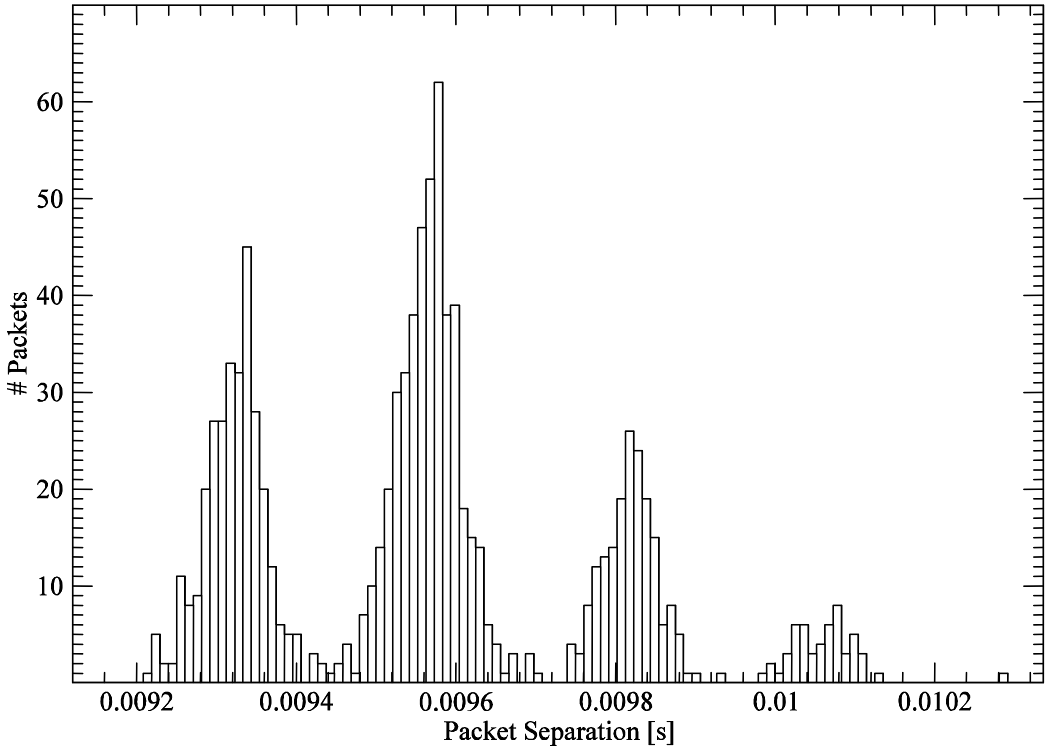


Figure 50: Histogram of Packet Pair Separation for a DSL-1024 kbit/s downlink (IP packet size: 1200 bytes)

Figure 50 shows the packet separation measured at the host `leela` which corresponds to the MR or MN. Packet pair measurements host `leela` have four distinct modes. These modes are separated by 250 μs which match the size of a DSL frame. The total range of packet separation is about 9%. This results in a deviation of measured bottleneck bandwidth of about 80 kbit/s at `leela`.

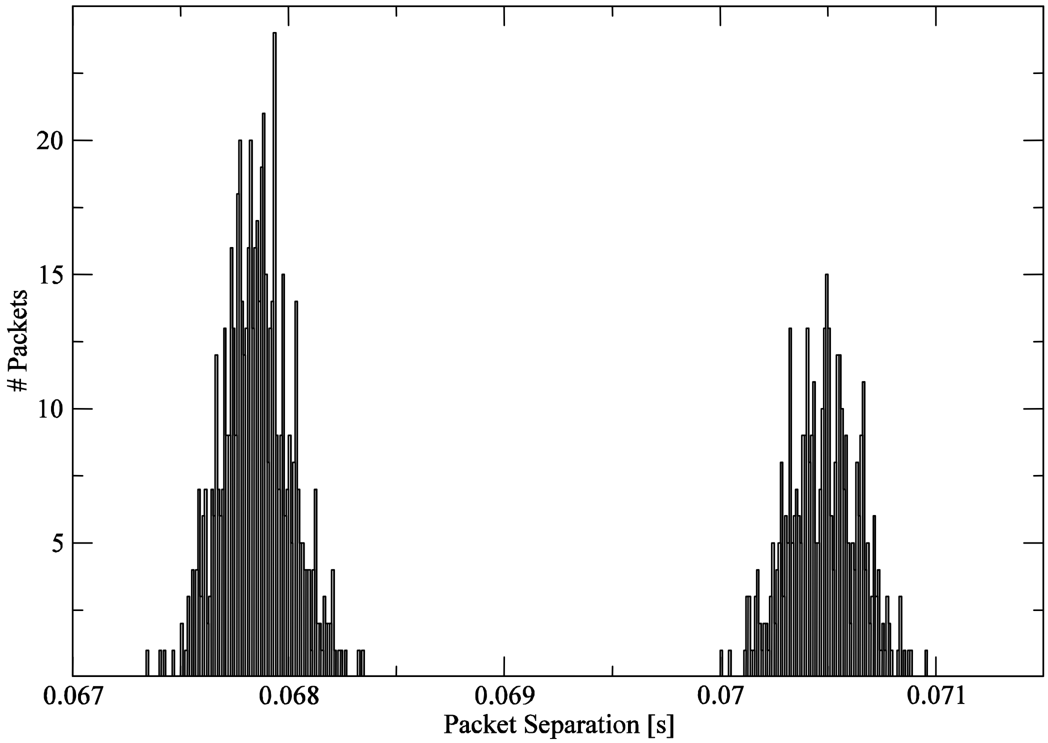


Figure 51: Histogram of Packet Pair Separation for a DSL-128 kbit/s uplink (IP packet size: 1200 bytes)

Measurements on the host `dallas` which corresponded to the HA or CN are depicted Figure 51. Packet pair separation via this uplink shows only two modes separated by about 2500 μs. However this is a range of about 4 % and results in a bandwidth deviation of about 8 kbit/s.

Both results suggest that the effects of cross traffic on high capacity links do not heavily influence the bottleneck measurements if the bottleneck bandwidth is magnitudes lower. In this cases link layer effects, e.g. of DSL may outweigh the effects of cross traffic. Therefore in such scenarios the location of the bottleneck link may not have much influence on packet pair measurements.

4.4.2.5 Influence of Packet Sizes on Measurement Accuracy

Figure 52, Figure 53, Figure 54 and Figure 55 depict both packet separation and bottleneck bandwidth measurements from the HA/CN to the MR/MN (Figure 52 and Figure 53) and vice versa (Figure 54 and Figure 55), for different packet sizes. For the reason of clarity only every second IP packet size that results in an optimal filling of ATM cells without padding is shown. Both packet separation graphs show that the error in packet pair separation is mainly independent from packet size.

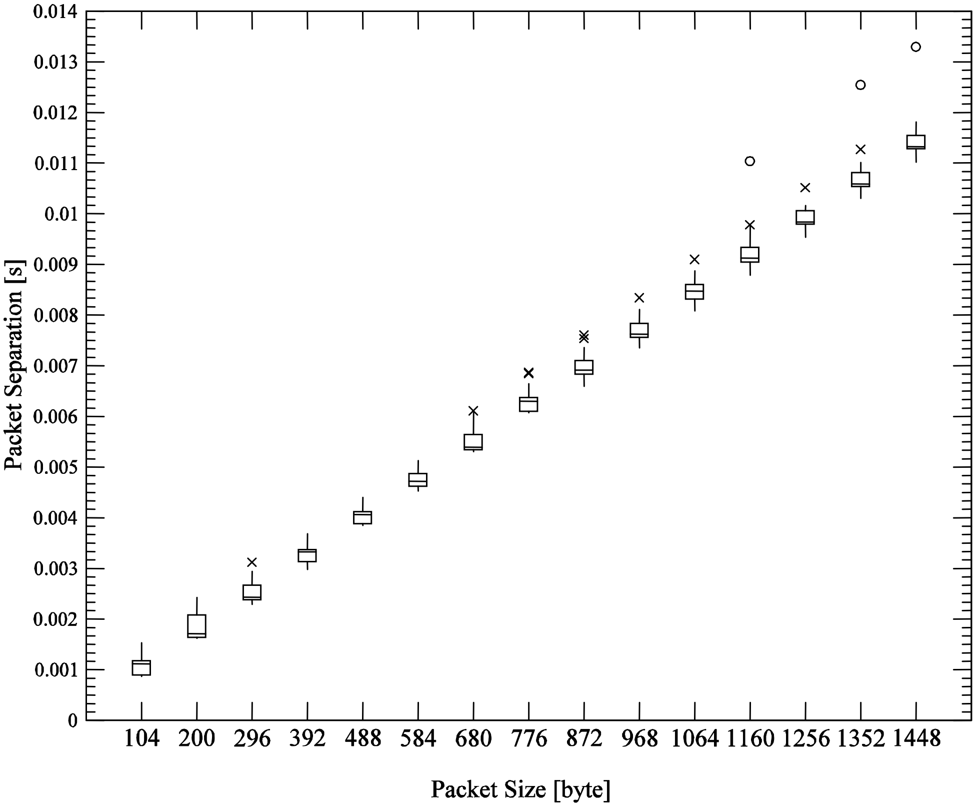


Figure 52: DSL-1024 kbit/s Downlink Packet Separation

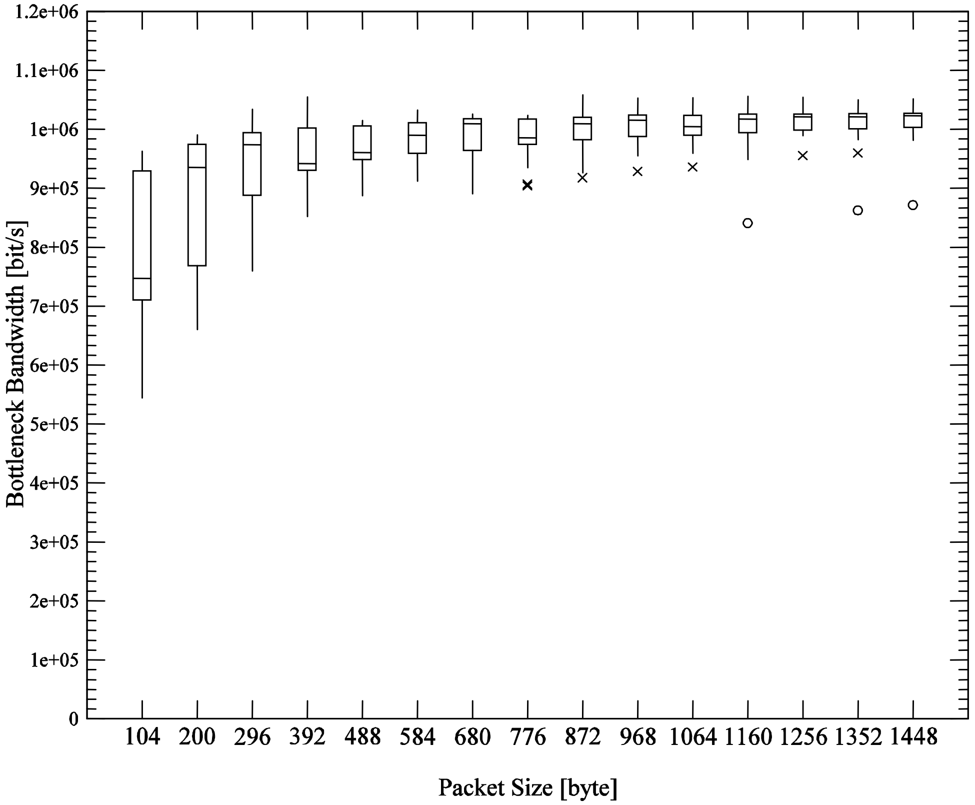


Figure 53: DSL-1024 kbit/s Downlink Bottleneck Bandwidth

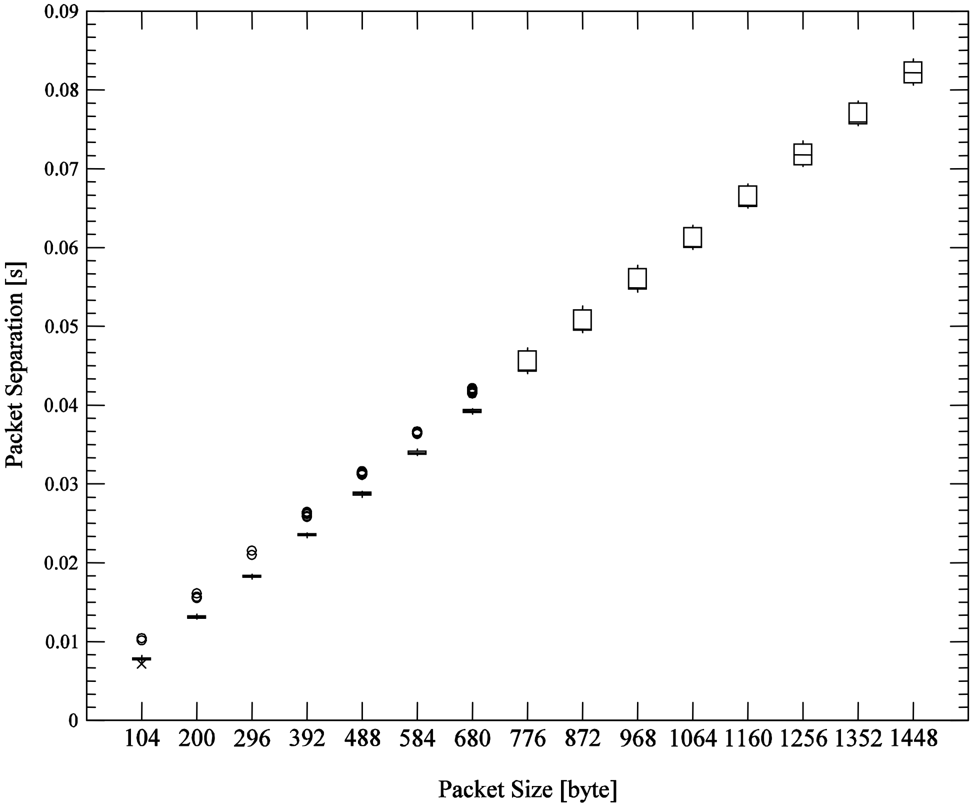


Figure 54: DSL-128 kbit/s Uplink Packet Separation

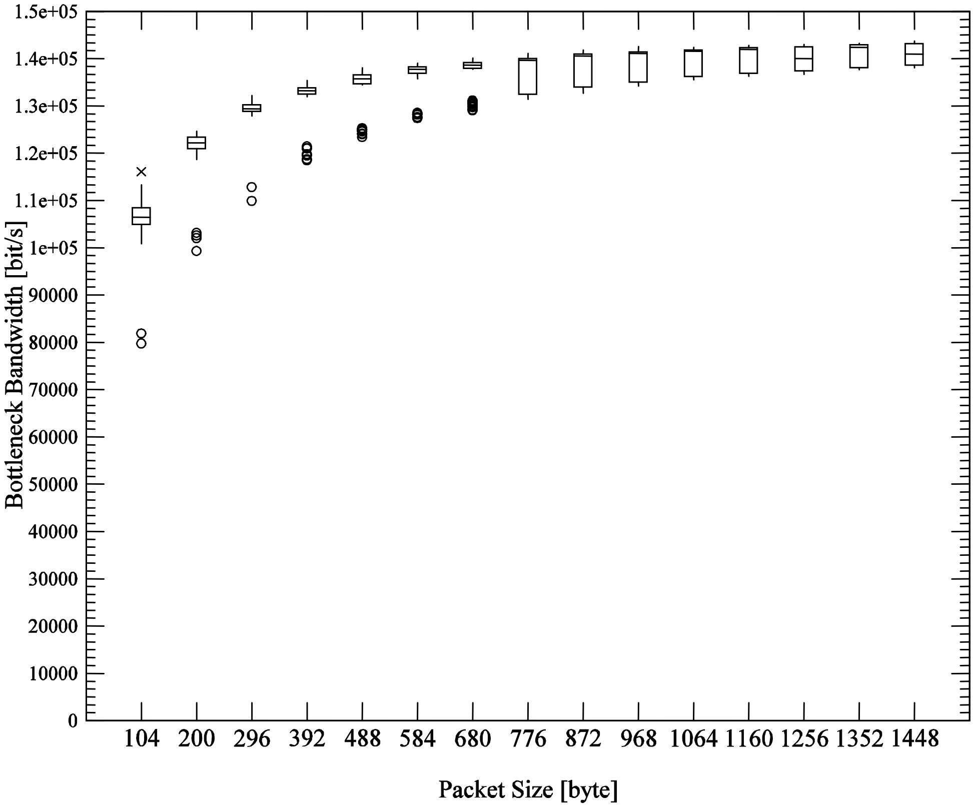


Figure 55: DSL-128 kbit/s Uplink Bottleneck Bandwith

To analyse the propagation of error from packet separation measurements to bandwidth the error of bandwidth measurements is derived. Bandwidth or capacity C in bit/s is calculated from packet separation by $C = \frac{8d}{t}$ where d is the packet size in bytes and t is the packet separation in seconds. t is the only variable that encounters error Therefore we consider C to be a function of t while fixing d. Applying a Taylor expansion we get

$$C(x) = C(t_0) + \frac{8d}{{t_0}^2} * (x - t_0) + \int_{t_0}^{t} C''(t)(x-t)dt .$$

Omitting the remainder of the Taylor expansion the error of C is approximately: $\Delta C \approx \frac{d}{{t_0}^2} \cdot \Delta t$.

When expanding about a value t_0 which is calculated by the mathematic model developed in section 4.4.2.1 we get

$$\Delta C \approx \frac{8d}{t^2} * \Delta t = \frac{8d}{\left(\frac{8d}{E * C_0}\right)^2} * \Delta t = \frac{E^2 {C_0}^2}{8d} * \Delta t .$$

Figure 56 (left) depicts the estimated error in bandwidth calculation for a protocol overhead of 40 bytes per packet. As a basis for both graphs, the error in packet separation is approximated with 500 μs and the nominal link speed C_0 is 1152000 bit/s, corresponding to the DSL-1024 bottleneck downlink. The figure shows a strong increase in bandwidth error when decreasing the packet size to about 50 bytes. The decrease for packets smaller than 50 bytes is due to omitting the remainder of the Taylor expansion in error estimation. However, smaller packets are not of practical importance because the minimum transfer unit for many link layer technologies is about 50 bytes.

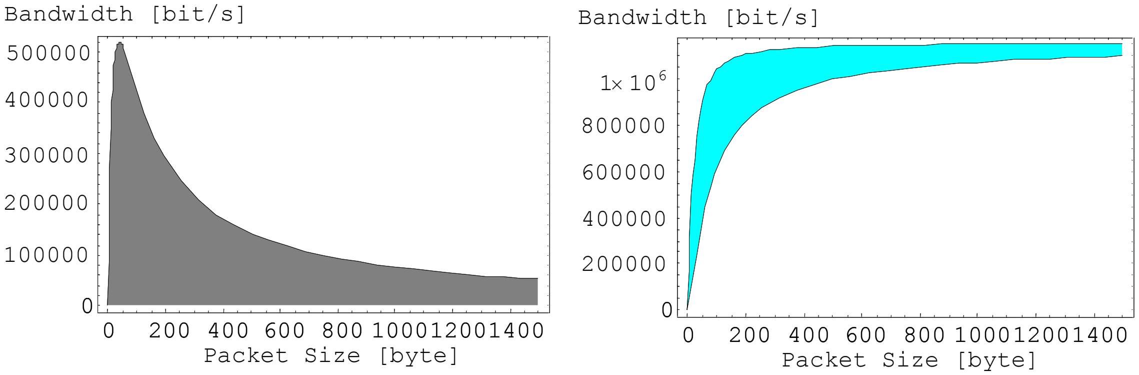


Figure 56: Bandwidth Error [bit/s] as Function of Packet Size (left) and Bandwidth Error Corridor [bit/s] (right)

Figure 56 (right) depicts a symmetric error area around the expansion point, which is itself a function of packet size. As already stated in previous sections the area of error might be asymmetric in practice. Nevertheless this asymmetry does not change the error estimate as we have only considered the linear addend of the Taylor expansion.

4.4.2.6 Influence of Link Speed and Host Capabilities on Packet Pair Measurements

The second test application described in section 4.3.3.1 was adaptive video streaming over IPv6. Packet pair measurements were non-intrusively integrated into the video stream. Figure 57 depicts the results of streaming a video with an average bitrate of 512 kbit/s and an MTU of about 1200 bytes on a 100 Mbit/s and 10 Mbit/s link. The incursion about measurement 1450 on both the 100 Mbit/s and 10 Mbit/s link is due to the video server not being able to send packet pairs fast enough.

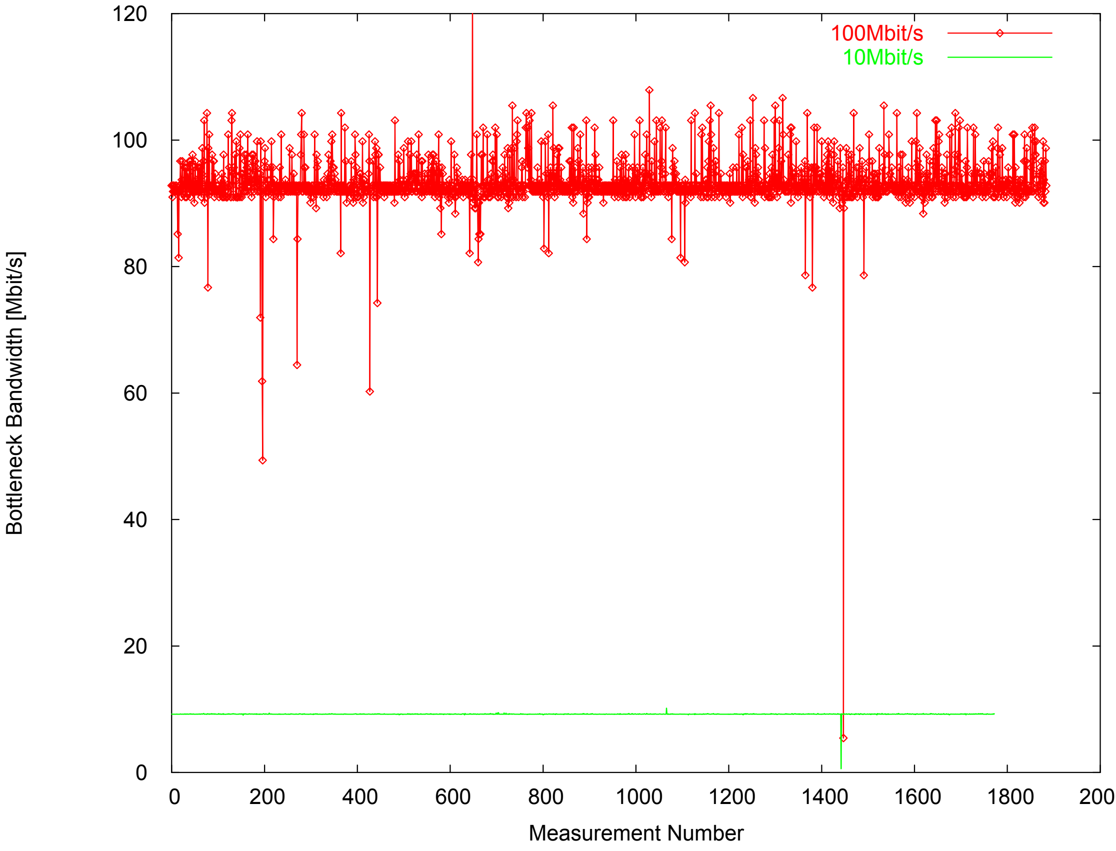


Figure 57: Packet Pair Measurements Integrated into a Video Application

Another important observation is the high variance of the measurements on the 100 Mbit/s link compared to the 10 Mbit/s link. These errors mainly result from small errors in the accuracy of time measurements on the host systems hardware and operating system. When using Linux this error strongly depends from the kernel version. Operating systems which have better real time behaviour result in more accurate measurements. However, faster network adapters like Gigabit Ethernet may also send a single interrupt for several data packets received to offload the host. This feature is called interrupt coalescing. Moving packet arrival time measurements form the operating system to the network adapter, which means setting timestamps for received packets on the network adapter itself may overcome some of these problems.

All host specific errors have more effect on fast links than on slow. This is because the absolute packet pair separation is smaller for faster links than for slower while the absolute measurement error caused by the hosts operating system and hardware is independent from link speed.

Another factor is the minimum time interval between the packets of a packet pair the sender can supply its network adapter. The resulting bandwidth might be lower than the link speed. Especially when it comes to measuring high speed links. In the adaptive video scenario the video server was able to hand over packets of the same video frame to the TCP/IP protocol stack with a peek data rate of about 500 Mbit/s with a high variance. This would also exclude the correct detection of for example a Gigabit Ethernet bottleneck link.

For link speeds used in wireless scenarios however the problem of measurement inaccuracies seems to be negligible. However handheld devices with very low computational power and no hardware support for setting timestamps for packet pairs might have problems monitoring broadband wireless links.

5 Conclusion

The OverDRiVE project successfully investigated and developed mechanisms to support mobile/moving networks based on Mobile IPv6 solutions. Such mobile networks might be found in future vehicles such as trains, cars, ships, etc. This deliverable investigated advanced topics of the problem space extending the work done as described in deliverable D07 [2]. Basically three main topics were covered in the document:

- Route optimizations in the scope of moving networks,
- Traffic Management, and
- Validation activities.

The optimization work covers route optimization in order to use a path of optimal length between communication entities, i.e. avoid multi-angular routing overhead via mobility agents. Focus within the work was put on route optimization within the vehicle to save air time, reduce latencies and allow for disconnected operation. For route optimization between the Mobile Router and Correspondent Nodes a threat analysis for the basic mobility mode paves the way for further solutions in the field of wide area networks. The traffic management section studies in detail the effect of Binding Update storms by means of simulations. Binding Update storms can occur if all mobile nodes handle their mobility on their own. This study points to the usefulness of a Mobile Router depending on the envisioned context. Beside that, traffic management approaches where developed to lay out the basis for traffic shaping and fair bandwidth allocation for nodes within an IVAN. A special focus on end-to-end methods avoids requirements of signalling and relies on bandwidth measurements (e.g. packet dispersion techniques). In the validation and measurement section the interworking of micro- and macro-mobility (BCMP and Mobile IPv6) approaches is described. The measurements show the advantages of such a combined usage especially for large vehicles such as trains. The work had significant impact on the scientific community by actively contributing to IETF NEMO working group [15] and publications by all project partners on various conferences and workshops (cf. [4],[5],[6],[7],[8],[9],[10],[11],[12],[18],[29]). The demonstration as described in deliverable D14 [3] further increased to awareness of the scientific community and provided valuable feedback for validation. With this work a basic solution incorporating certain optimizations (e.g. route optimization, mico-mobility, etc.) is described which could be used for deployment in real products.

6 References

[1] OverDRiVE Deliverable D03, "OverDRiVE Scenarios, Services, and Requirements"; http://www.ist-OverDRiVE.org, September 2002

[2] OverDRiVE Deliverable D07, "Concept of Mobile Router and Dynamic IVAN Management"; http://www.ist-OverDRiVE.org, March 2003

[3] OverDRiVE Deliverable D14, "Description of Demonstrator for Mobile Multicast and the Vehicular Router"; http://www.ist-OverDRiVE.org, March 2004

[4] R. Tönjes, K. Mößner, T. Lohmar, M. Wolf: "OverDRiVE - Spectrum Efficient Multicast Services to Vehicles", IST Mobile Summit, Thessaloniki, 16-19.June, 2002

[5] W. Hansmann, M. Frank, M. Wolf "Performance Analysis of TCP Handover in a Wireless/Mobile Multi-Radio Environment" Proc. of the 27th Annual Conference on Local Computer Networks, LCN'02, Tampa, FL, November 2002

[6] M. Wolf, "Evaluation of Mobility Management Approaches for IPv6 based Mobile Car Networks", KiVS 2003, Leipzig, 25-28th February 2003

[7] H.-Y. Lach, C. Janneteau, A. Petrescu, "Network Mobility in Beyond-3G System", IEEE Communications Magazine, Vol. 41, Issue 7, July 2003

[8] Miklós Aurél Rónai, Ralf Tönjes, Michael Wolf, Alexandru Petrescu: "Mobility Issues in OverDRiVE Mobile Networks", in proceedings of the IST Mobile & Wireless Communications Summit 2003, Aveiro, Portugal, 15-18 June 2003, pp. 287-291

[9] W. Hansmann, M. Frank, "On Things to happen during a TCP Handover", 28th Annual Conference on Local Computer Networks, LCN'03, Königswinter, Germany, October 2003

[10] A. Petrescu, "OverDRiVE Moving Networks: Achievements and Perspectives",3rd Workshop on Internet Vehicles, Nara Institute of Science andTechnology, Nara, Japan, March 8th, 2004.

[11] H.-Y. Lach, C. Janneteau, A. Petrescu, "MR-HA Bidirectional Tunneling for Network Mobility", B3G Report, A Publication by Systems Beyond 3G Cluster, Information Society Technologies, October 31st, 2003

[12] H.-Y. Lach, C. Janneteau, A. Olivereau, A. Petrescu, T. Leinmüller, M. Wolf, M. Pilz, "Laboratory and Field Experiments with IPv6 Mobile Networks in Vehicular Environments", draft-lach-nemo-experiments-overdrive-01 (work in progress), October 2003

[13] D. Johnson, C. Perkins, J. Arkko, Mobility Support in IPv6, draft-ietf-mobileip-ipv6-24 (work in progress), June 2003

[14] IETF Working Group mipshop, http://www.ietf.org/html.charters/mipshop-charter.html

[15] IETF Working Group nemo, http://www.ietf.org/html.charters/nemo-charter.html

[16] Network simulator ns-2, http://www.isi.edu/nsnam/ns/index.html

[17] Thierry Ernst, "MobiWan: NS-2 extensions to study mobility in Wide-Area IPv6 Networks", http://www.inrialpes.fr/planete/mobiwan, May 2002

[18] Gergely Biczók, Kristóf Fodor, Balázs Kovács, "Handover Latencies in BCMP Networks", Komunikácie/Communications journal, Zilina, Slovakia

[19] Vijay Devarapalli, Ryuji Wakikawa, Alexandru Petrescu, and Pascal Thubert, "Nemo basic support protocol," Internet Draft draft-ietfnemo-basic-support-02.txt (Work in Progress), IETF, Dec. 2003.

[20] Miklós Aurél Rónai, Ralf Tönjes, Michael Wolf, and Alexandru Petrescu, "Mobility issues in overdrive mobile networks," in Proceedings of the IST Mobile & Wireless Communications Summit 2003, Aveiro, Portugal, June 2003.

[21] S. Deering and R. Hinden, "Internet protocol, version 6 (ipv6) specification," RFC (Standards Track) 2460, IETF, Dec. 1998.

[22] P. Thubert, M. Molteni, C. Ng, H. Ohnishi, and E. Paik, "Taxonomy of route optimization models in the nemo context," Internet Draft draft-thubert-nemo-ro-taxonomy-02 (Work in Progress), IETF, Feb. 2004.

[23] Kyeong-Jin Lee, Jae-Hoon Jeong, Jung-Soo Park, and Hyoung-Jun Kim, "Route optimization for mobile nodes in mobile network based on prefix delegation," Internet Draft draft-leekj-nemo-ro-pd- 02.txt (Work in Progress), IETF, Feb. 2004.

[24] Jaehoon Paul Jeong, Kyeongjin Lee, Jungsoo Park, and Hyoungjun Kim, "Nd-proxy based route and dns optimizations for mobile nodes in mobile network," Internet Draft draft-jeong-nemo-rondproxy-02.txt (Work in Progress), IETF, Feb. 2004.

[25] H. Soliman et. al., "Hierarchical mipv6 mobility management (hmipv6)," Internet Draft draft-ietf-mobileip-hmipv6-06.txt (Work in Progress), IETF, July 2002.

[26] H. Ohnishi, K. Sakitani, and Y. Takagi, "HMIP based route optimization method in a mobile network," Internet Draft draft-ohnishinemo-ro-hmip-00.txt (Work in Progress), IETF, Oct. 2003.

[27] Thierry Ernst, Alexis Olivereau, Ludovic Bellier, Claude Castelluccia, and Hong-Yon Lach, "Mobile networks support in mobile ipv6 (prefix scope binding updates)," Internet Draft draft-ernst-mobileipv6-network-03.txt (Work in Progress), IETF, Mar. 2002.

[28] T. Narten, E. Nordmark, and W. Simpson, "Neighbor discovery for ip version 6 (ipv6)," RFC (Standards Track) 2461, IETF, Dec. 1998.

[29] C. Barz, M. Frank, H.-Y. Lach, A. Petrescu, M. Pilz, M. Wolf, and L. Zombik, "Network access control in overdrive mobile networks," in *Proceedings of the IST Mobile & Wireless Communications Summit 2003*, Aveiro, Portugal, June 2003.

[30] A. Petrescu, M. Catalina-Gallego, C. Janneteau, H.-Y. Lach, and A. Olivereau, "Issues in designing mobile ipv6 network mobility with the mr-ha bidirectional tunnel (mrha)," Internet Draft draftpetrescu- nemo-mrha-02.txt (Work in Progress), IETF, Oct. 2003.

[31] Thierry Ernst, Alexis Olivereau, Ludovic Bellier, Claude Castelluccia, and Hong-Yon Lach, "Mobile networks support in mobile ipv6 (prefix scope binding updates)," Internet Draft draft-ernst-mobileipv6-network-03.txt (Work in Progress), IETF, Mar. 2002.

[32] T. Narten, E. Nordmark, and W. Simpson, "Neighbor discovery for ip version 6 (ipv6)," RFC (Standards Track) 2461, IETF, Dec. 1998.

[33] C. Barz, M. Frank, H.-Y. Lach, A. Petrescu, M. Pilz, M. Wolf, and L. Zombik, "Network access control in overdrive mobile networks," in *Proceedings of the IST Mobile & Wireless Communications Summit 2003*, Aveiro, Portugal, June 2003.

[34] A. Petrescu, M. Catalina-Gallego, C. Janneteau, H.-Y. Lach, and A. Olivereau, "Issues in designing mobile ipv6 network mobility with the mr-ha bidirectional tunnel (mrha)," Internet Draft draftpetrescu- nemo-mrha-02.txt (Work in Progress), IETF, Oct. 2003.

[35] John A. C. Bingham, "ADSL, VDSL, and Multicarrier Modulation", John Wiley & Sons Inc., 2000

[36] Sally Floyd, Vern Paxson, "Difficulties in Simulating the Internet", IEEE/ACM Transactions on Networking, vol. 9, no. 4, pp. 392–403, Aug. 2001.

[37] "PING Testing over xDSL End-to-End Service Verification", Sunrise Telecom Incorporated, 2001

[38] "The DSL Sourcebook", Paradyne Corporation, 2000

[39] Apple Darwin Streaming Server, http://developer.apple.com/darwin/projects/streaming/]

[40] MPEG4IP, http://mpeg4ip.net

[41] The libpcap project, http://sourceforge.net/projects/libpcap/

[42] Attila Pasztor and Darryl Veitch, "The packet size dependence of packetpair like methods", in Proc. of IWQoS'2002

[43] 3GPP TS 23.060 V4.9.0 (2003-12)

[44] V. Jacobson, "Pathchar: A Tool to Infer Characteristics of Internet Paths", ftp://ftp.ee.lbl.gov/pathchar, April 1997

[45] A.B. Downey, "Using Pathchar to Estimate Internet Link Characteristics", Proc. ACM SIGCOMM, Sept. 1999, pp. 222-223

[46] A. Petrescu, A. Olivereau, C. Janneteau and H-Y. Lach, "Threats for Basic Network Mobility Support", Internet Draft, draft-petrescu-nemo-threats-01.txt (Work in Progress), IETF, January 2004.

[47] S. Jung, F. Zhao, S. Felix Wu and H. Kim, „Threat Analysis on NEMO Basic Operations", Internet Draft, draft-jung-nemo-threat-analysis-02, (Work in Progress), IETF, February 2004.

[48] J. Arkko, V. Devarapalli and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents", Internet Draft, draft-ietf-mobileip-mipv6-ha-ipsec-06.txt, (Work in Progress), IETF, June 2003.

[49] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing", IETF, BCP 38, RFC 2827, May 2000.

[50] E. Rescorla, "A Survey of Authentication Mechanisms", Internet Draft, draft-rescorla-auth-mech-02.txt, (Work in Progress), IETF, October 2003.

[51] Prasad, Dovrolis, Murray, claffy, "Bandwidth Estimation: Metrics, Measurement Techniques, and Tools", IEEE Network, Nov/Dec 2003

[52] Dovrolis, Ramanathan, Moore, "What do packet dispersion techniques measure?", IEEE Infocom, April 2001

[53] M. Allman, "Measuring End-to-End Bulk Transfer Capacity", ACM SIGCOMM Internet Measurement Workshop, November 2001

[54] M. Mathis, M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", RFC 3148, July 2003

[55] M. Allman, V. Paxson, W. R. Stevens, "TCP Congestion Control", RFC 2581, April 1999

[56] Multi Router Traffic Grapher, http://people.ee.ethz.ch/~oetiker/webtools/mrtg

[57] Waldbusser, "Remote Network Monitoring Management Information Base", RFC 2819, May 2000

[58] V. Jacobsen, "Pathchar: A Tool to infer Characteristics of Internet Paths", ftp://ftp.ee.lbl.gov/pathchar, April 1997

[59] Bellovin, "A Best-Case Network Performance Model", February 1992

[60] K. Lai, M. Baker, "Measuring Bandwidth", in Proceedings of IEEE INFOCOM, March 1999

[61] K. Lai ,"Measuring the Bandwidth of Packet Switched Networks", Ph.D. Thesis, Stanford University, October 2002

[62] V. Jacobsen, "Congestion Avoidance and Control", ACM SIGCOMM, September 1988

[63] Paxson, "End-to-End Internet Packet Dynamics", IEEE/ACM Transaction on Networking, vol. 7, no. 3, pp. 277-292, June 1999

[64] HTB (Hierachical Token Bucket) Home, http://luxik.cdi.cz/~devik/qos/htb

[65] 6bone, "testbed for deployment of IPv6", http://www.6bone.net

[66] International Workshop on Hybrid Wireless Networks, http://www.ist-overdrive.org/HyWiN2003

[67] V4/v6 web proxy, F.W. Dillema, University of Tromsø, Norway in 2001

[68] A. Conta, S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998

Annex A IPsec Protection of NEMO Binding Messages

Unprotected BU and BAck when MR in foreign network

```
   Base Header                        Base Header
     src: CoA                   (*)     src: Home Agent address    (*)
     dst: Home Agent address    (*)     dst: CoA                   (*)
   Destination Options                Routing Header
     Home Address: Home Address (*)     Routing Type: 2            (*)
   Mobility Header                      Segments Left: 1           (*)
     Header Len                        Home Address: Home Address (*)
     MH Type:                        Mobility Header
     Reserved:                         Header Len:
     Checksum:                         MH Type:
     Message Data                      Reserved:
       Seq #                           Checksum:
       AHLKR                    (*)     Message Data
       Lifetime:                         Status                   (*)
       Mobility Options                         K:
         Alternate CoA: CoA              Reserved
         Mobile Net Prefix Option         Seq #:
           Prefix Len:          (*)       Lifetime:
           Mobile Net Prefix:   (*)       Mobility Options
                                           Binding Refresh Advice
                                             Refresh Interval:
```

Unprotected BU and BAck when MR returns home

```
   Base Header                        Base Header
     src: Home Address          (*)     src: Home Agent address    (*)
     dst: Home Agent address    (*)     dst: Home Address          (*)
   Mobility Header                   Mobility Header
     Header Len                        Header Len:
     MH Type:                          MH Type:
     Reserved:                         Reserved:
     Checksum:                         Checksum:
     Message Data                      Message Data
       Seq #                             Status                   (*)
       AHLKR                    (*)       K:
       Lifetime:                         Reserved
       Mobility Options                         Seq #:
         Mobile Net Prefix Option         Lifetime:
           Prefix Len:          (*)       Mobility Options
           Mobile Net Prefix:   (*)         Binding Refresh Advice
                                           Refresh Interval:
```

AH protected BU and BAck when MR in foreign network

```
  Base Header                         Base Header
    src: CoA                   (*x)     src: Home Agent address   (*x)
    dst: Home Agent address    (*x)     dst: CoA                  (*x)
  Destination Options                Routing Header
    Home Address: Home Address (*x)     Routing Type: 2           (*x)
  Authentication Header                       Segments Left: 1
(*x)
```

```
  SPI:                               Home Address: Home Address(*x)
  Seq No:                          Authentication Header
  ICV:                               SPI:
Mobility Header                      Seq No:
  Header Len                         ICV:
  MH Type:                         Mobility Header
  Reserved:                          Header Len:
  Checksum:                          MH Type:
  Message Data                       Reserved:
    Seq #                            Checksum:
    AHLKR                    (*x)     Message Data
    Lifetime:                          Status                  (*x)
    Mobility Options                   K:
      Alternate CoA: CoA               Reserved
      Mobile Net Prefix Option          Seq #:
        Prefix Len:          (*x)       Lifetime:
        Mobile Net Prefix:   (*x)       Mobility Options
                                         Binding Refresh Advice
                                           Refresh Interval:
```

AH protected BU and BAck when MR returns home

```
Base Header                         Base Header
  src: Home Address          (*x)     src: Home Agent address   (*x)
  dst: Home Agent address    (*x)     dst: Home Address          (*x)
Authentication Header                     Authentication Header
  SPI:                               SPI:
  Seq No:                            Seq No:
  ICV:                               ICV:
Mobility Header                    Mobility Header
  Header Len                         Header Len:
  MH Type:                           MH Type:
  Reserved:                          Reserved:
  Checksum:                          Checksum:
  Message Data                       Message Data
    Seq #                              Status                  (*x)
    AHLKR                    (*x)       K:
    Lifetime:                          Reserved
    Mobility Options                   Seq #:
      Mobile Net Prefix Option          Lifetime:
        Prefix Len:          (*x)       Mobility Options
        Mobile Net Prefix:   (*x)          Binding Refresh Advice
                                          Refresh Interval:
```

ESP NULL auth algo for BU and BAck when MR in foreign network

```
Base Header                         Base Header
  src: CoA                    (*)     src: Home Agent address    (*)
  dst: Home Agent address     (*)     dst: CoA                   (*)
Destination Options                Routing Header
  Home Address: Home Address  (*)     Routing Type: 2            (*)
ESP Header                           Segments Left: 1          (*)
  SPI:                               Home Address: Home Address (*)
  Seq No:                          ESP Header
Mobility Header                      SPI:
  Header Len                         Seq No:
  MH Type:                         Mobility Header
  Reserved:                          Header Len:
  Checksum:                          MH Type:
  Message Data                       Reserved:
    Seq #                            Checksum:
    AHLKR                     (*y)    Message Data
    Lifetime:                          Status                  (*y)
    Mobility Options                   K:
      Alternate CoA: CoA               Reserved
      Mobile Net Prefix Option          Seq #:
        Prefix Len:          (*y)       Lifetime:
        Mobile Net Prefix:   (*y)       Mobility Options
ESP Trailer                              Binding Refresh Advice
                                           Refresh Interval:
                                   ESP Trailer
```

ESP NULL auth algo for BU and BAck when MR returns home

```
Base Header                         Base Header
  src: Home Address           (*)     src: Home Agent address    (*)
  dst: Home Agent address     (*)     dst: Home Address          (*)
ESP Header                         ESP Header
  SPI:                               SPI:
  Seq No:                            Seq No:
Mobility Header                    Mobility Header
  Header Len                         Header Len:
  MH Type:                           MH Type:
  Reserved:                          Reserved:
  Checksum:                          Checksum:
  Message Data                       Message Data
    Seq #                              Status                  (*y)
    AHLKR                     (*y)      K:
    Lifetime:                          Reserved
    Mobility Options                   Seq #:
      Mobile Net Prefix Option          Lifetime:
        Prefix Len:          (*y)       Mobility Options
        Mobile Net Prefix:   (*y)         Binding Refresh Advice
ESP Trailer                                Refresh Interval:
                                   ESP Trailer
```

ESP non-NULL auth algo for BU and BAck when MR in foreign network

```
Base Header                         Base Header
  src: CoA                    (*)     src: Home Agent address    (*)
  dst: Home Agent address     (*)     dst: CoA                   (*)
Destination Options                       Routing Header
  Home Address: Home Address  (*)     Routing Type: 2            (*)
ESP Header                           Segments Left: 1          (*)
  SPI:                               Home Address: Home Address (*)
  Seq No:                          ESP Header
Mobility Header                      SPI:
  Header Len                         Seq No:
  MH Type:                         Mobility Header
  Reserved:                          Header Len:
  Checksum:                          MH Type:
  Message Data                       Reserved:
    Seq #                            Checksum:
    AHLKR                    (*yz)    Message Data
```

```
      Lifetime:                          Status                 (*yz)
      Mobility Options                   K:
        Alternate CoA: CoA                      Reserved
        Mobile Net Prefix Option          Seq #:
          Prefix Len:         (*yz)       Lifetime:
          Mobile Net Prefix:  (*yz)       Mobility Options
  ESP Trailer                              Binding Refresh Advice
  ESP Auth                                   Refresh Interval:
                                    ESP Trailer
                                    ESP Auth
```

ESP non-NULL auth algo for BU and BAck when MR returns home

```
  Base Header                         Base Header
    src: Home Address           (*)     src: Home Agent address    (*)
    dst: Home Agent address     (*)     dst: Home Address          (*)
  ESP Header                        ESP Header
    SPI:                              SPI:
    Seq No:                           Seq No:
  Mobility Header                   Mobility Header
    Header Len                        Header Len:
    MH Type:                          MH Type:
    Reserved:                         Reserved:
    Checksum:                         Checksum:
    Message Data                      Message Data
      Seq #                             Status                  (*yz)
      AHLKR                   (*yz)       K:
      Lifetime:                         Reserved
      Mobility Options                  Seq #:
        Mobile Net Prefix Option          Lifetime:
          Prefix Len:         (*yz)       Mobility Options
          Mobile Net Prefix:  (*yz)         Binding Refresh Advice
  ESP Trailer                                Refresh Interval:
  ESP Auth                          ESP Trailer
                                    ESP Auth
```

AH and ESP NULL for BU and BAck when MR in foreign network

```
  Base Header                         Base Header
    src: CoA                   (*x)     src: Home Agent address   (*x)
    dst: Home Agent address    (*x)     dst: CoA                  (*x)
  Destination Options               Routing Header
    Home Address: Home Address (*x)     Routing Type: 2           (*x)
  Authentication Header                       Segments Left: 1
(*x)
    SPI:                              Home Address: Home Address(*x)
    Seq No:                         Authentication Header
    ICV:                              SPI:
  ESP Header                          Seq No:
    SPI:                              ICV:
    Seq No:                         ESP Header
  Mobility Header                     SPI:
    Header Len                        Seq No:
    MH Type:                        Mobility Header
    Reserved:                         Header Len:
    Checksum:                         MH Type:
    Message Data                      Reserved:
      Seq #                            Checksum:
      AHLKR                   (*xy)     Message Data
      Lifetime:                          Status                 (*xy)
      Mobility Options                   K:
        Alternate CoA: CoA               Reserved
        Mobile Net Prefix Option          Seq #:
          Prefix Len:         (*xy)       Lifetime:
          Mobile Net Prefix:  (*xy)       Mobility Options
  ESP Trailer                              Binding Refresh Advice
                                             Refresh Interval:
                                     ESP Trailer
```

AH and ESP NULL for BU and BAck when MR returns home

```
  Base Header                         Base Header
    src: Home Address           (*x)    src: Home Agent address   (*x)
    dst: Home Agent address     (*x)    dst: Home Address         (*x)
  Authentication Header                      Authentication Header
    SPI:                               SPI:
    Seq No:                            Seq No:
    ICV:                               ICV:
  ESP Header                         ESP Header
    SPI:                               SPI:
    Seq No:                            Seq No:
  Mobility Header                    Mobility Header
    Header Len                         Header Len:
    MH Type:                           MH Type:
    Reserved:                          Reserved:
    Checksum:                          Checksum:
    Message Data                       Message Data
      Seq #                              Status                 (*xy)
      AHLKR (*xy)                        K:
      Lifetime:                          Reserved
      Mobility Options                   Seq #:
        Mobile Net Prefix Option          Lifetime:
          Prefix Len:         (*xy)       Mobility Options
          Mobile Net Prefix:  (*xy)         Binding Refresh Advice
  ESP Trailer                                  Refresh Interval:
                                     ESP Trailer
```

AH and ESP non-NULL for BU and BAck when MR in foreign network

```
  Base Header                         Base Header
    src: CoA                    (*x)    src: Home Agent address   (*x)
    dst: Home Agent address     (*x)    dst: CoA                  (*x)
  Destination Options                Routing Header
    Home Address: Home Address (*x)    Routing Type: 2           (*x)
  Authentication Header                        Segments Left: 1
(*x)
    SPI:                               Home Address: Home Address(*x)
    Seq No:                          Authentication Header
    ICV:                               SPI:
  ESP Header                           Seq No:
    SPI:                               ICV:
    Seq No:                          ESP Header
  Mobility Header                      SPI:
    Header Len                         Seq No:
```

```
  MH Type:                         Mobility Header
  Reserved:                          Header Len:
  Checksum:                          MH Type:
  Message Data                       Reserved:
    Seq #                            Checksum:
    AHLKR                  (*xyz)     Message Data
    Lifetime:                          Status                (*xyz)
    Mobility Options                   K:
      Alternate CoA: CoA               Reserved
      Mobile Net Prefix Option          Seq #:
        Prefix Len:        (*xyz)       Lifetime:
        Mobile Net Prefix: (*xyz)       Mobility Options
ESP Trailer                              Binding Refresh Advice
ESP Auth                                   Refresh Interval:
                                   ESP Trailer
                                   ESP Auth
```

AH and ESP non-NULL for BU and BAck when MR returns home

```
Base Header                         Base Header
  src: Home Address          (*x)     src: Home Agent address   (*x)
  dst: Home Agent address    (*x)     dst: Home Address         (*x)
Authentication Header                     Authentication Header
  SPI:                               SPI:
  Seq No:                            Seq No:
  ICV:                               ICV:
ESP Header                         ESP Header
  SPI:                               SPI:
  Seq No:                            Seq No:
Mobility Header                    Mobility Header
  Header Len                         Header Len:
  MH Type:                           MH Type:
  Reserved:                          Reserved:
  Checksum:                          Checksum:
  Message Data                       Message Data
    Seq #                              Status                (*xyz)
    AHLKR                  (*xyz)       K:
    Lifetime:                          Reserved
    Mobility Options                   Seq #:
      Mobile Net Prefix Option          Lifetime:
        Prefix Len:        (*xyz)       Mobility Options
        Mobile Net Prefix: (*xyz)         Binding Refresh Advice
ESP Trailer                                Refresh Interval:
ESP Auth                           ESP Trailer
                                   ESP Auth
```

Annex B Traceroute Measurements for RO Study

B.1 Traceroute Omnitel/Vodafone FrontBox to Google

```
 1  10.136.50.3 (10.136.50.3)  679.524 ms  659.901 ms  659.893 ms
 2  10.136.49.1 (10.136.49.1)  659.978 ms  649.945 ms  669.986 ms
 3  10.136.48.161 (10.136.48.161)  690.005 ms  669.877 ms  679.977 ms
 4  10.127.1.22 (10.127.1.22)  659.983 ms  659.956 ms  650.003 ms
 5  10.128.219.60 (10.128.219.60)  669.930 ms  719.947 ms  659.985 ms
 6  10.128.219.188 (10.128.219.188)  679.987 ms  669.959 ms  669.984 ms
 7  10.129.211.11 (10.129.211.11)  679.979 ms  679.969 ms  689.969 ms
 8  194.185.97.2 (194.185.97.2)  650.070 ms  599.929 ms  539.973 ms
 9  212.239.126.25 (212.239.126.25)  689.984 ms  680.379 ms  679.521 ms
10     ge3-1.milano1-gsr0.net.inet.it   (194.185.46.77)      669.977   ms   ge3-0.milano1-
gsr1.net.inet.it    (194.185.46.13)        649.922    ms    ge3-1.milano1-gsr0.net.inet.it
(194.185.46.77)  639.954 ms
11  srp10-0.m1-cr2.net.inet.it (194.185.46.169)  649.952 ms  619.954 ms  660.010 ms
12  t2a1-p11-0-0.it-mil2.concert.net (166.49.142.21)  689.913 ms  649.932 ms  569.993 ms
13  t2c1-ge6-0.it-mil2.concert.net (166.49.180.11)  599.972 ms  649.968 ms  689.948 ms
14  t2c2-p10-3.uk-lon2.concert.net (166.49.164.242)  719.956 ms  669.876 ms  679.975 ms
15  t2c1-ge6-2.uk-lon2.concert.net (166.49.164.169)  749.990 ms  699.968 ms  719.926 ms
16  t2c1-p1-0.nl-ams2.concert.net (166.49.195.225)  719.973 ms  739.933 ms  719.984 ms
17  t2c2-ge6-1.nl-ams2.concert.net (166.49.208.182)  689.984 ms  629.954 ms  659.984 ms
18  t2c2-p2-0.uk-eal.concert.net (166.49.195.33)  719.996 ms  690.001 ms  649.933 ms
19  t2c2-p5-0.us-ash.concert.net (166.49.164.22)  779.989 ms  789.947 ms  809.985 ms
20  eqixva-google-gige.google.com (206.223.115.21)  839.987 ms  779.953 ms  779.981 ms
21  * 64.233.174.134 (64.233.174.134)  900.000 ms  779.956 ms
22  216.239.48.89 (216.239.48.89)  769.991 ms  779.965 ms  1079.989 ms
23  * 216.239.48.94 (216.239.48.94)  830.009 ms  769.902 ms
24  * * *
25  * *
```

B.2 Traceroute FrontBox Omnitel/Vodafone to Udptun GW

```
1  10.136.50.3 (10.136.50.3)  810.390 ms  659.833 ms  569.976 ms
 2  10.136.49.1 (10.136.49.1)  659.996 ms  579.951 ms  689.956 ms
 3  10.136.48.161 (10.136.48.161)  599.927 ms  669.946 ms  629.987 ms
 4  10.127.1.22 (10.127.1.22)  699.981 ms  589.889 ms  579.978 ms
 5  10.128.219.60 (10.128.219.60)  599.996 ms  539.875 ms  599.981 ms
 6  10.128.219.188 (10.128.219.188)  599.988 ms  589.898 ms  649.985 ms
 7  10.129.211.11 (10.129.211.11)  679.987 ms  599.941 ms  659.986 ms
 8  194.185.97.2 (194.185.97.2)  599.989 ms  649.938 ms  609.983 ms
 9  s4-0.milano1-cr3.net.inet.it (194.185.4.245)  610.040 ms  719.921 ms  809.984 ms
```

```
10  ge3-0.milano1-gsr1.net.inet.it (194.185.46.13)  690.024 ms  639.902 ms  689.987 ms
11  garr-mix.mix-it.net (217.29.66.39)  719.993 ms  679.983 ms  579.918 ms
12  rtg1-rtg2.mi.garr.net (193.206.134.17)  609.985 ms *  750.011 ms
13  rt-rtg.mi.garr.net (193.206.134.205)  869.982 ms  579.961 ms  679.941 ms
14  to-mi.garr.net (193.206.134.62)  599.990 ms  579.956 ms  579.973 ms
15  polito-rc.to.garr.net (193.206.132.146)  3309.999 ms  679.941 ms  749.984 ms
16  130.192.230.253 (130.192.230.253)  4469.997 ms  4589.956 ms  3889.982 ms
17  193.204.112.219 (193.204.112.219)  3659.983 ms  3979.946 ms  4559.983 ms
```

B.3 Traceroute Udptun GW to H3G NAT

```
1  193.204.112.253 (193.204.112.253)  1.761 ms  1.761 ms  1.698 ms
 2  130.192.230.254 (130.192.230.254)  30.976 ms  30.755 ms  30.234 ms
 3  rc-polito.to.garr.net (193.206.132.145)  132.762 ms  131.592 ms  145.405 ms
 4  mi-to.garr.net (193.206.134.61)  153.216 ms  162.565 ms  159.771 ms
 5  rtg-rt.mi.garr.net (193.206.134.206)  163.042 ms  149.451 ms  161.726 ms
 6  so-6-0-0.ar2.LIN1.gblx.net (64.214.196.241)  144.958 ms  153.327 ms  157.415 ms
 7  pos6-0-2488M.cr1.LIN1.gblx.net (67.17.65.129)  136.071 ms  155.866 ms  161.862 ms
 8  so0-0-0-2488M.cr2.LON3.gblx.net (67.17.64.38)  164.619 ms  164.728 ms  160.839 ms
 9  so7-0-0-2488M.ar2.LON3.gblx.net (67.17.66.30)  163.957 ms  147.585 ms  160.102 ms
10  sl-bb21-lon-1-3.sprintlink.net (213.206.131.25)  164.319 ms  154.645 ms  178.808 ms
11  sl-bb20-lon-15-0.sprintlink.net (213.206.128.37)  178.682 ms  203.134 ms  205.108 ms
12  sl-bb21-par-14-0.sprintlink.net (213.206.129.70)  204.068 ms  180.282 ms  194.247 ms
13  sl-bb20-mil-12-0.sprintlink.net (213.206.129.79)  225.899 ms  182.617 ms  852.023 ms
14  217.147.128.68 (217.147.128.68)  1086.830 ms  245.676 ms  844.293 ms
15  sle-h3gs-1-0.sprintlink.net (217.147.129.158)  279.605 ms  241.915 ms  224.595 ms
16  62.13.167.34 (62.13.167.34)  258.750 ms  254.163 ms  243.863 ms
17  * * *
18  *
```

B.4 Traceroute Udptun GW to Vodafone/Omnitel NAT

```
1  193.204.112.253 (193.204.112.253)  1.818 ms  1.800 ms  1.733 ms
 2  130.192.230.254 (130.192.230.254)  30.393 ms  30.684 ms  41.631 ms
 3  rc-polito.to.garr.net (193.206.132.145)  169.356 ms  155.696 ms  159.288 ms
 4  mi-to.garr.net (193.206.134.61)  164.223 ms  196.531 ms  194.520 ms
 5  rtg-rt.mi.garr.net (193.206.134.206)  209.569 ms  162.969 ms  131.939 ms
 6  rtg2-rtg1.mi.garr.net (193.206.134.18)  134.872 ms  153.605 ms  154.331 ms
 7  inet-mix.mix-it.net (217.29.66.2)  137.651 ms  160.133 ms  150.455 ms
 8  fe0-0.milano1-cr3.net.inet.it (194.185.46.2)  161.837 ms  192.594 ms  204.624 ms
 9  s1-0.gw-omnitelrete.inet.it (194.185.4.246)  207.180 ms  196.512 ms  194.311 ms
10  * * *
11  * * *
12  * * *
```

```
13  *
```

B.5 Traceroute Udptun GW to Orange NAT

```
1  193.204.112.253 (193.204.112.253)  1.766 ms  1.751 ms  1.726 ms
 2  130.192.230.254 (130.192.230.254)  106.549 ms  493.354 ms  78.485 ms
 3  rc-polito.to.garr.net (193.206.132.145)  155.986 ms  133.258 ms  145.420 ms
 4  mi-to.garr.net (193.206.134.61)  162.992 ms  192.635 ms  177.325 ms
 5  rtg-rt.mi.garr.net (193.206.134.206)  202.835 ms  265.693 ms *
 6  so-6-0-0.ar2.LIN1.gblx.net (64.214.196.241)  233.346 ms  211.655 ms  119.246 ms
 7  pos6-0-2488M.cr1.LIN1.gblx.net (67.17.65.129)  152.255 ms  117.106 ms  168.366 ms
 8  so0-0-0-2488M.cr2.AMS2.gblx.net (67.17.64.94)  247.926 ms  242.009 ms  295.501 ms
 9  so1-0-0-2488M.ar1.AMS1.gblx.net (67.17.65.242)  226.706 ms  175.916 ms  188.417 ms
10  64.215.195.210 (64.215.195.210)  203.088 ms  184.131 ms  227.562 ms
11   P6-0.AMSBB2.Amsterdam.opentransit.net (193.251.242.142)  237.857 ms  195.133 ms
217.377 ms
12  So3-0-0.FFTCR1.Frankfurt.opentransit.net (193.251.154.165)  282.392 ms  156.411 ms
175.113 ms
13  P1-0.AUVCR1.Aubervilliers.opentransit.net (193.251.132.74)  264.705 ms  244.130 ms
297.538 ms
14    pos6-0.ntaub201.Aubervilliers.francetelecom.net (193.251.126.153)  295.475 ms
270.657 ms  295.267 ms
15  pos14-0.ntsta202.Paris.francetelecom.net (193.252.161.29)  223.454 ms  286.470 ms
326.498 ms
16  pos13-2.nrsta104.Paris.francetelecom.net (193.251.126.101)  298.125 ms  260.888 ms
221.496 ms
17  194.51.159.54 (194.51.159.54)  195.977 ms  349.303 ms  371.953 ms
18   POS-12-0.RASR2.Raspail.raei.transitip.francetelecom.net (81.52.10.21)  839.364 ms
2622.246 ms  2587.482 ms
19  POS-4-0-0.RAS6.Raspail.raei.francetelecom.net (81.52.10.10)  2570.783 ms  2570.290 ms
2567.658 ms
20  81.54.100.250 (81.54.100.250)  2664.421 ms  2639.824 ms  2678.139 ms
21  * * *
```

B.6 Traceroute6 Udptun GW to Budweiser

```
1  3ffe:b80:14d1:1000::1 (3ffe:b80:14d1:1000::1)  0.489 ms  0.654 ms  0.397 ms
 2  3ffe:b80:2:f327::1 (3ffe:b80:2:f327::1)  356.099 ms  348.374 ms  313.63 ms
 3  rap.ipv6.viagenie.qc.ca (3ffe:b00:c18:1:290:27ff:fe17:fc0f)  365.126 ms  367.773 ms
1086.58 ms
 4  edt-viagenie.ipv6.edisontel.it (3ffe:8170:1:11::1)  477.475 ms  491.65 ms  741.866 ms
 5  3ffe:401:0:1::19:1 (3ffe:401:0:1::19:1)  622.332 ms  532.37 ms  730.341 ms
 6  budweiser.cs.uni-bonn.de (3ffe:400:450:1070:260:97ff:fe0d:1f61)  560.519 ms  517.014
ms *
```

B.7 Traceroute6 Udptun GW to www.enrl.motlabs.com

```
1  3ffe:b80:14d1:1000::1 (3ffe:b80:14d1:1000::1)  0.651 ms  0.441 ms  0.396 ms
```

```
 2  3ffe:b80:2:f327::1 (3ffe:b80:2:f327::1)  308.88 ms  322.372 ms  337.213 ms

 3  viagenie.tu-0.plalca01.us.b6.verio.net (2001:418:0:4000::26)  324.006 ms  309.089 ms
315.245 ms

 4  3ffe:b00:c18:1017::2 (3ffe:b00:c18:1017::2)  490.529 ms  508.959 ms  438.396 ms

 5  2002:c3d4:6ffd:1:201:2ff:fe6c:eff8 (2002:c3d4:6ffd:1:201:2ff:fe6c:eff8)  573.55 ms
605.956 ms  575.621 ms
```

B.8 Traceroute6 Udptun GW to www.kame.net

```
1  3ffe:b80:14d1:1000::1 (3ffe:b80:14d1:1000::1)  0.485 ms  0.436 ms  0.394 ms

 2  3ffe:b80:2:f327::1 (3ffe:b80:2:f327::1)  306.073 ms  339.124 ms  346.063 ms

 3  Viagenie-gw.int.ipv6.ascc.net (2001:288:3b0::55)  346.146 ms  298.823 ms  344.844 ms

 4  gw-Viagenie.int.ipv6.ascc.net (2001:288:3b0::54)  712.116 ms *  702.479 ms

 5  c7513-gw.int.ipv6.ascc.net (2001:288:3b0::c)  715.851 ms  731.73 ms  710.94 ms

 6  m160-c7513.int.ipv6.ascc.net (2001:288:3b0::1e)  707.76 ms  717.486 ms  720.048 ms

 7  m20jp-m160tw.int.ipv6.ascc.net (2001:288:3b0::1b)  909.745 ms  884.323 ms *

 8  hitachi1.otemachi.wide.ad.jp (2001:200:0:1800::9c4:2)  708.138 ms  715.41 ms  742.667
ms

 9  pc3.yagami.wide.ad.jp (2001:200:0:1c04::1000:2000)  752.648 ms  740.909 ms  719.665
ms

10  gr2000.k2c.wide.ad.jp (2001:200:0:8002::2000:1)  744.351 ms  743.504 ms  689.675 ms

11  orange.kame.net (2001:200:0:8002:203:47ff:fea5:3085)  705.472 ms  749.108 ms  742.944
ms
```

B.9 Traceroute FrontBox UMTS to Google

```
1  216.239.57.99 (216.239.57.99)  371.201 ms  379.886 ms  369.986 ms

 2  * * *

 3  62.13.180.21 (62.13.180.21)  299.989 ms  399.880 ms  389.979 ms

 4  62.13.180.21 (62.13.180.21)  399.983 ms  379.963 ms  399.986 ms

 5  62.13.168.66 (62.13.168.66)  689.990 ms  309.871 ms  329.972 ms

 6  62.13.168.1 (62.13.168.1)  340.001 ms  309.948 ms  339.991 ms

 7  yar1-serial0-0.Milan.cw.net (208.175.149.1)  339.980 ms  309.942 ms  299.990 ms

 8  ycr2-ge-3-2-0-0.Milan.cw.net (208.175.148.130)  349.986 ms  299.966 ms  309.989 ms

 9  ycr2-so-1-0-0-1.Zurichzuh.cw.net (208.175.232.89)  319.987 ms  309.914 ms  309.989 ms

10  ycr1-so-0-3-0.Zurichzuh.cw.net (208.175.232.21)  319.985 ms  309.972 ms  309.984 ms

11  bcr1-so-7-0-0-1.Frankfurt.cw.net (166.63.195.209)  349.988 ms  319.968 ms  309.992 ms

12  dcr1-loopback.SantaClara.cw.net (208.172.146.99)  499.984 ms  479.967 ms  569.988 ms

13  bhr1-pos-0-0.SantaClarasc5.cw.net (208.172.156.74)  479.985 ms  569.963 ms  569.990
ms

14  csr21-ve242.SantaClarasc4.cw.net (216.34.3.18)  579.983 ms  549.968 ms  579.987 ms

15  google-exodus.exodus.net (64.41.147.62)  569.990 ms  549.966 ms  579.985 ms

16  216.239.49.2 (216.239.49.2)  579.988 ms  589.969 ms  589.996 ms

17  * * *

18  * * *

19  * * *
```

```
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

B.10 Traceroute FrontBox UMTS to Udptun GW

```
1  193.204.112.219 (193.204.112.219)  305.805 ms  389.871 ms  409.981 ms
 2  * * *
 3  62.13.180.21 (62.13.180.21)  310.011 ms  379.943 ms  399.988 ms
 4  62.13.180.21 (62.13.180.21)  399.979 ms  769.968 ms  419.979 ms
 5  62.13.168.66 (62.13.168.66)  329.985 ms  299.980 ms  299.987 ms
 6  62.13.168.17 (62.13.168.17)  449.986 ms  469.982 ms  389.987 ms
 7  yar1-serial0-0.Milan.cw.net (208.175.149.1)  569.987 ms  379.942 ms  399.988 ms
 8  ycr1-ge-3-3-0-0.Milan.cw.net (208.175.148.129)  369.994 ms  369.970 ms  399.988 ms
 9  bcr1-so-7-0-0-0.Paris.cw.net (208.172.251.153)  409.988 ms  439.964 ms  389.985 ms
10   cable-and-wireless-peering.Paris.cw.net  (208.172.251.182)   409.992  ms   1609.970  ms
299.984 ms
11  prs-bb2-pos0-3-0.telia.net (213.248.70.9)  299.989 ms  309.975 ms  299.989 ms
12  mno-vcn-i1-pos0-1.telia.net (213.248.77.50)  319.988 ms  319.976 ms  329.986 ms
13  dante-01355-mno-vcn-i1.c.telia.net  (213.248.79.126)  339.989 ms  339.972 ms  329.988
ms
14  mi-rm-g.garr.net (193.206.134.13)  329.990 ms  329.976 ms  339.985 ms
15  rt-rtg.mi.garr.net (193.206.134.205)  339.991 ms  340.024 ms  339.941 ms
16  to-mi.garr.net (193.206.134.62)  349.989 ms  349.977 ms  339.990 ms
17  polito-rc.to.garr.net (193.206.132.146)  389.988 ms  409.976 ms  409.991 ms
18  130.192.230.253 (130.192.230.253)  809.988 ms  969.903 ms  1489.951 ms
19  193.204.112.219 (193.204.112.219)  599.978 ms  549.979 ms  499.984 ms
```

B.11 Traceroute ENRL to Budweiser

```
1  195.212.111.241 (195.212.111.241)  2.013 ms  1.878 ms  2.058 ms
 2  152.158.68.150 (152.158.68.150)  19.084 ms  19.269 ms  18.824 ms
 3  frpari2102cr2--0-1-2-0.pa.fr.ip.att.net (165.87.211.65)  40.020 ms  19.426 ms  20.194
ms
 4  desttg1102cr2--0-1-1-0.st.de.ip.att.net (165.87.211.33)  32.273 ms  39.774 ms  32.224
ms
 5   desttg1101cr2--2-0-0-0.st.de.ip.att.net  (165.87.216.245)    34.345  ms    32.168  ms
33.679 ms
```

```
 6   defrnk1101cr2--4-0-0-0.fr.de.ip.att.net  (165.87.212.201)    35.614  ms    35.383  ms
38.804 ms
 7  defrnk1101er1-0-0.fr.de.ip.att.net (165.87.216.146)  35.478 ms  35.148 ms  35.607 ms
 8  dcix1nap-5-0-0.de.ip.att.net (165.87.212.126)  35.940 ms  35.547 ms  36.064 ms
 9  ir-frankfurt2.g-win.dfn.de (80.81.192.222)  36.516 ms  36.130 ms  36.874 ms
10  cr-frankfurt1-po3-0.g-win.dfn.de (188.1.80.41)  48.393 ms  49.865 ms  49.157 ms
11  cr-koeln1-po4-0.g-win.dfn.de (188.1.18.98)  41.577 ms  40.982 ms  42.544 ms
12  ar-koeln3.g-win.dfn.de (188.1.84.5)  42.030 ms  40.449 ms  41.117 ms
13  kr-bonn.rhrz.uni-bonn.de (131.220.254.2)  42.320 ms  41.576 ms  42.212 ms
14  sr1-rz-vlan3.rhrz.uni-bonn.de (131.220.1.249)  42.631 ms  41.029 ms  41.966 ms
15  cr-rz-po20.rhrz.uni-bonn.de (131.220.1.57)  41.486 ms  45.545 ms  41.333 ms
16  ar-vb5-po15.rhrz.uni-bonn.de (131.220.1.33)  44.145 ms  41.484 ms  44.805 ms
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
```

B.12 Traceroute6 ENRL to Budweiser

```
1   2002:c3d4:6ffd:1:260:b0ff:fe56:cc85  (2002:c3d4:6ffd:1:260:b0ff:fe56:cc85)   0.431  ms
0.222 ms  0.198 ms
 2  6to4.ipv6.funet.fi (2001:708:0:1::624)  77.131 ms  74.457 ms  74.421 ms
 3  helsinki0-g2106-sixpack.ipv6.funet.fi (2001:708:0:1::625)  81.991 ms  77.6 ms  75.701
ms
 4  fi-gw2.nordu.net (2001:948:0:f035::1)  75.469 ms  77.458 ms  90.688 ms
 5  se-kth.nordu.net (2001:948:0:f02b::1)  81.631 ms  84.356 ms  103.102 ms
 6  sw-gw.nordu.net (2001:948:0:f025::1)  81.855 ms  82.116 ms  82.15 ms
 7  6net-gw.nordu.net (2001:948:0:f02a::2)  82.741 ms  88.957 ms  83.67 ms
 8  nordunet.se6.se.6net.org (2001:798:25:200::1)  85.328 ms  90.258 ms  77.59 ms
 9  se.de6.de.6net.org (2001:798:0:8::2)  79.41 ms  78.842 ms  83.9 ms
10  de.nl6.nl.6net.org (2001:798:0:5::2)  79.05 ms  78.877 ms  79.616 ms
11  2001:798:22:200::2 (2001:798:22:200::2)  79.298 ms  79.43 ms  79.486 ms
12   Gi5-1-2.BR2.Amsterdam1.surf.net  (2001:610:16:6036::37)   103.41  ms   90.366  ms   81.58
ms
13  2001:610:16:6052::54 (2001:610:16:6052::54)  186.349 ms  185.139 ms  192.175 ms
14  3ffe:3900:a0::1 (3ffe:3900:a0::1)  175.087 ms  174.774 ms  174.898 ms
15  3ffe:3900:1000:1::2 (3ffe:3900:1000:1::2)  176.521 ms  177.319 ms  176.561 ms
16  2001:400:0:53::1 (2001:400:0:53::1)  189.502 ms  190.574 ms  190.73 ms
17  2001:400:0:77::1 (2001:400:0:77::1)  189.274 ms  190.173 ms  189.906 ms
18  2001:400:0:54::1 (2001:400:0:54::1)  238.153 ms  237.406 ms  237.16 ms
19  2001:400:0:61::2 (2001:400:0:61::2)  238.688 ms  238.566 ms  237.905 ms
20  2001:400:0:80::2 (2001:400:0:80::2)  238.823 ms  295.668 ms  255.818 ms
21  3ffe:1001:1:ffff::2 (3ffe:1001:1:ffff::2)  1021.2 ms  1052.52 ms  1014.22 ms
22  gw1-bk1.ipv6.tilab.com (2001:6b8:0:100::2)  897.896 ms  1035.41 ms  1040.19 ms
```

```
23  3ffe:1001:1:f011::2 (3ffe:1001:1:f011::2)  1036.72 ms  1095.74 ms  1065.74 ms

24    *  budweiser.cs.uni-bonn.de  (3ffe:400:450:1070:260:97ff:fe0d:1f61)    1053.12  ms
1092.54 ms
```

Annex C Ping Measurements over GPRS and UMTS Links

A large number of ping4 and ping6 tests were performed between the GPRS and UMTS FrontBoxes and the respective wireless access systems ("roaming" GPRS: Orange, "home" GPRS: Omnitel/Vodafone, UMTS: H3G). The pings varied the packet size, the number of packets and the interval between packets (flooding or non-flooding pings). The measurements tried to identify the Round Trip Times (RTT) on the wireless links, the IP Maximum Transmission Unit size, the symmetricity of the link, and various other characteristics.

Due to space constraints we will not paste the ping outputs in this deliverable, but in a separate document entitled "Companion to D17: GPRS/UMTS Ping Measurements".

Annex D Source code examples of ETH’s testbed

D.1 The content of the */etc/radvd.conf* on the home agent

```
interface eth2
{
   AdvSendAdvert on;
   AdvHomeAgentFlag on;
   prefix 3ffe:b80:1ee4:a::1/64
   {
        AdvRouterAddr on;
   };
};
```

D.2 The */etc/radvd.conf* file of Access Router 1

```
interface eth2
{
   AdvSendAdvert on;
   prefix 3ffe:b80:1ee4:5::2/64
   {
        AdvRouterAddr off;
   };
};
```

D.3 In the file */etc/network-mip6.conf* the following values have been set

```
FUNCTIONALITY=mn
HOMEADDRESS=3ffe:b80:1ee4:a::2/64
HOMEAGENT=3ffe:b80:1ee4:a::1/64
```

D.4 The status of the MRHA tunnel and the mobile router’s binding update

```
MR# ipv6tunnel show ip6tnl1
ip6tnl1: IPv6/IPv6 \
         remote 3ffe:b80:1ee4:a::1 \
         local 3ffe:b80:1ee4:5:202:2dff:fe42:d569 \
         hoplimit 255 flags ELKM

MR# mipdiag -l
Mobile IPv6 Binding update list
Recipient CN: 3ffe:b80:1ee4:a::1
BINDING home address: 3ffe:b80:1ee4:a::2 \
care-of address: 3ffe:b80:1ee4:5:202:2dff:fe42:d569
expires: 129 sequence: 10 state: 1
delay: 1 max delay 256 callback time: 82
```

D.5 The status of the MRHA reverse tunnel and the home agent’s binding cache

```
HA# ipv6tunnel show ip6tnl1
ip6tnl1: IPv6/IPv6 \
         remote 3ffe:b80:1ee4:5:202:2dff:fe42:d569 \
         local 3ffe:b80:1ee4:a::1 \
         hoplimit 255 flags ELKM

HA# mipdiag -c
Mobile IPv6 Binding cache
```

```
Home Address 3ffe:b80:1ee4:a::2
Care-of Address 3ffe:b80:1ee4:5:202:2dff:fe42:d569
Lifetime 867
Type 2
```

D.6 The source code of the kernel in file *linux/net/ipv6/ndisc.c*

```
static void ndisc_router_discovery(struct sk_buff *skb)
{
...
 if (in6_dev->cnf.forwarding ||
            !in6_dev->cnf.accept_ra) {
                in6_dev_put(in6_dev);
                return;
        }
...
}
```

D.7 The code sample of the required modifications

```
static struct option long_options[] =
    {
        ...
        {"networkaddress", 2, 0, 'N'},
        ...
    };
c = getopt_long (argc, argv,
                 "i:IP:mslc?Vd::t::H::h:N::",
                 long_options, &option_index);

int rtn_set_mn_info(int ifindex,
                    struct in6_ifreq *home,
                    struct in6_ifreq *ha,
                    struct in6_ifreq *na)
    // this last parameter is the network address
{
    ...
    addattr_l(&req.n, sizeof(req), IFA_NPREFIX,
              &na->ifr6_prefixlen,
              sizeof(u_int32_t));
    addattr_l(&req.n, sizeof(req), IFA_NADDRESS,
              &na->ifr6_addr,
              sizeof(struct in6_addr));
    ...
    if (rtnl_talk(&rth, &req.n, 0, 0,
        NULL, NULL, NULL) < 0)
        return -2;
}
```

D.8 Modifications in *addrconf.c*

```
inet6_rtm_newaddr(struct sk_buff *skb,
                  struct nlmsghdr *nlh, void *arg)
{
    ...
    if (rta[IFA_NPREFIX-1])
    {
        if (pfx == NULL ||
            !(ifm->ifa_flags & IFA_F_HOMEADDR))
            return -EINVAL;
        if (RTA_PAYLOAD(rta[IFA_NPREFIX-1]) <
            sizeof(nprefix))
            return -EINVAL;
        nprefix = *(u_int32_t*)
```

```
                  RTA_DATA(rta[IFA_NPREFIX-1]);
    }
    if (rta[IFA_NADDRESS-1])
    {
        struct in6_addr *na;
        if (pfx == NULL ||
            !(ifm->ifa_flags & IFA_F_HOMEADDR))
            return -EINVAL;
        if (RTA_PAYLOAD(rta[IFA_NADDRESS-1]) <
            sizeof(*na))
            return -EINVAL;
        na = RTA_DATA(rta[IFA_NADDRESS-1]);
        addrconf_set_mipv6_mn_network_address(na,
                                              nprefix);
    }
    ...
}
```

D.9 Modifications in *mipglue.h*

```
static inline void
addrconf_set_mipv6_mn_network_address(
                       struct in6_addr *networkaddress,
                       u_int32_t nprefix)
{
    MIPV6_CALLPROC(mipv6_set_network_address)
                             (networkaddress, nprefix);
}
/* pointers to mipv6 callable functions */
struct mipv6_callable_functions
{
    ...
    void (*mipv6_set_network_address)
                           (struct in6_addr *home_addr,
                            u_int32_t n_prefix);
    ...
};

int __init mipv6_mn_init(void)
{
    ...
    MIPV6_SETCALL(mipv6_set_network_address,
                  mipv6_mn_set_network_address);
    ...
}
```

D.10 Modifications in *mn.c*

```
void mipv6_mn_set_network_address(
                                struct in6_addr *naddr,
                                u_int32_t nprefix)
{
    networkaddress=(*naddr);
    networkprefix=nprefix;
}
```

D.11 Modifications in *mdetect.c*

```
if ((type & (IPV6_ADDR_MULTICAST |
             IPV6_ADDR_LINKLOCAL)) ||
     rt->rt6i_dev == &loopback_dev ||
     rtr_is_gw(rtr, rt) ||
     // additional prefix check here
     mipv6_prefix_compare16(&rt->rt6i_dst.addr, na,
                       (int)networkprefix) ||
     is_prefix_route(rtr, rt) ||
     (rt->rt6i_flags & RTF_DEFAULT))
        ret = 0;
```

D.12 Further modifications in *mdetect.c*

```
int mipv6_prefix_compare16(struct in6_addr *addr,
                           struct in6_addr *prefix,
                           unsigned int nprefix)
{
    int i;

    if (nprefix > 128)
        return 0;

    for (i = 0; nprefix > 0; nprefix -= 16, i++) {
        if (nprefix >= 16) {
            if (addr->s6_addr16[i] !=
         prefix->s6_addr16[i])
         return 0;
        } else {
            if (((addr->s6_addr16[i] ^
           prefix->s6_addr16[i]) &
                ((~0) << (16 - nprefix))) != 0)
                return 0;
            return 1;
        }
    }
return 1;
}
```

D.13 Modifications in *route.c*

```
struct rt6_info *rt6_add_dflt_router(
                               struct in6_addr *gwaddr,
                               struct net_device *dev)
{
    struct in6_rtmsg rtmsg;
    memset(&rtmsg, 0, sizeof(struct in6_rtmsg));
    rtmsg.rtmsg_type = RTMSG_NEWROUTE;
    ipv6_addr_copy(&rtmsg.rtmsg_gateway, gwaddr);
    rtmsg.rtmsg_metric = 1024;
// old flags
//  rtmsg.rtmsg_flags = RTF_GATEWAY | RTF_ADDRCONF | \
//                      RTF_DEFAULT | RTF_UP;
// new flags
    rtmsg.rtmsg_flags = RTF_GATEWAY | RTF_UP;
    rtmsg.rtmsg_ifindex = dev->ifindex;
    ip6_route_add(&rtmsg);
    return rt6_get_dflt_router(gwaddr, dev);
}
```

D.14 Examples for the states of the tunnel interface

```
// mobile router being at home
HA# ipv6tunnel show ip6tnl1
ip6tnl1: IPv6/IPv6 \
         remote :: \
         local :: \
         hoplimit 255 flags K
// mobile router connects to a foreign network \
// via an access router
```

```
HA# ipv6tunnel show ip6tnl1
ip6tnl1: IPv6/IPv6 \
         remote 3ffe:b80:1ee4:a::1 \
         local 3ffe:b80:1ee4:5:202:2dff:fe42:d569 \
         hoplimit 255 flags ELKM
```

D.15 Simple script for checking the tunnel

```
#!/bin/sh
# Restart using tclsh \
exec wish8.3 "$0" "$@"

# Mobile Router's home address
set mr_homeaddr "3ffe:b80:1ee4:a::2"

# The prefix of nodes behind the Mobile Router
set networkaddress "3ffe:b80:1ee4:c::/64"

catch { exec date +%T } timestamp
catch { exec ip route get $mr_homeaddr } msg1

while {1} {
 catch { exec ip route get $mr_homeaddr } msg2
 if { [string compare $msg1 $msg2] != 0 } {
  catch { exec date +%T.%N } timestamp
  set timestamp \
                      [ string range $timestamp 0 11]
  catch { exec route -A inet6 \
                        del $networkaddress \
                        gw $mr_homeaddr } routemsg
  puts "$timestamp Route deleted"
  catch { exec route -A inet6 \
                        add $networkaddress \
                        gw $mr_homeaddr } routemsg
  catch { exec date +%T.%N } timestamp
  set timestamp \
                      [ string range $timestamp 0 11]
  puts "$timestamp Route added"
  catch { exec date +%T } timestamp
  puts "$timestamp $msg2"
  puts ""
 }
 set msg1 $msg2
 exec sleep 0.001
}
```

D.16 The main paramteres that should be configure in the UR and ANP

```
user_registry = 3ffe:b80:1ee4:c:3::1

anp_ip = 3ffe:b80:1ee4:c:3::1
anp_pool = 3ffe:b80:1ee4:c::/64

# Access Router 1
ar_ip = 3ffe:b80:1ee4:c:3::2

# Access Router 2
ar_ip = 3ffe:b80:1ee4:c:4::2
```

D.17 Tunnel setup

```
// MRHA bidirectional tunnel
MR# ipv6tunnel show ip6tnl1
ip6tnl1: IPv6/IPv6 \
         remote 3ffe:b80:1ee4:a::1 \
         local 3ffe:b80:1ee4:5:202:2dff:fe42:d569 \
         hoplimit 255 flags ELKM

// BCMP tunnel towards access router 3
MR# ipv6tunnel show ip6tnl50
ip6tnl1: IPv6/IPv6 \
         remote 2002:0:1:7::1 \
         local 2002:0:1:3::1\
         hoplimit 255 flags EL
```

D.18 User space program and other modifications needed for fast handover

```
#include <linux/in6.h>
#include "mipv6_ioctl.h"

int fd;
int main(void) {
        fd = open("/dev/mipv6_dev", 0);
        if (fd < 0) {
                perror("open:");
                printf("error opening file\n");
        }
        ioctl(fd, IOCTL_SET_UNREACHABLE);
        printf("Performing handover\n");
        return 0;
}
```

The *IOCTL_SET_UNREACHABLE* variable must be defined in the file *net/ipv6/mobile_ip6/mipv6_ioctl.h*:

```
#define IOCTL_SET_UNREACHABLE _IOR(MAJOR_NUM,
                                   19, void *)
```

The call then must be handled by the function *mipv6_ioctl()* in file *net/ipv6/mobile_ip6/ioctl_mn.c*:

```
case IOCTL_SET_UNREACHABLE:
        set_unreachable();
        break;
```

This set_unreachable() function is implemented in file net/ipv6/mobile_ip6/mdetect.c:

```
int manual_not_reachable = 0;
...
void set_unreachable(void)
{
        manual_not_reachable = 1;
        timer_handler(1);
}
```

If the function *timer_handler()* is called with the variable *manual_not_reachable* set to 1 the mobile router is immediately forced to send out a router solicitation message and to perform handover to another access router nearby. The modifications needed in the function *timer_handler()* are the following:

```
// if (state == NO_RTR)
if (state == NO_RTR || manual_not_reachable) {
        timeout = rs_send();
}
...
// } else if (oldstate == NO_RTR) {
} else if (oldstate == NO_RTR ||
                       manual_not_reachable) {
```

```
        manual_not_reachable = 0;
        DEBUG(DBG_INFO, "No router or router not
              reachable, switching to new one");
        goto handoff;
}
```

D.19 GPRS connection; PPP setup

- */etc/ppp/options*:

```
/dev/ttyS0 19200
connect '/usr/sbin/chat -v -f /etc/ppp/chat-gprs'
dump
debug
kdebug 4
noauth
crtscts
defaultroute
show-password
#:10.0.0.1
noipdefault
nopcomp
user "USERNAME"
```

- */etc/ppp/chat-gprs*:

```
TIMEOUT        5
ECHO           ON
''             \rAT
TIMEOUT        12
OK             ATE0V1
OK             AT
OK             ATD*99***1#
```

- */etc/ppp/chat-gprs:*

```
"USERNAME"     *       "PASSWORD"     *
```

D.20 Configuring *vtund*

- VGGSN:

```
options {
  port 5000; # Connect to this port.
  timeout 1; # General timeout

# Path to various programs
  ppp /usr/sbin/pppd;
  ifconfig /sbin/ifconfig;
  route /sbin/route;
}

# UDP connection
gprs-udp {
  pass abc; # Password
  type ether;
  proto udp;
  keepalive yes;
  device tap2;
  compress lzo:9;
  up {
     # Connection is Up
     # Assign IP addresses.
     ifconfig "%% 10.48.11.2 netmask 255.255.255.0 mtu 1300";
     ifconfig "%% add 3ffe:b80:1ee4:b:1::1/80";
```

```
            program "/data/overdrive/setroute.sh";
         };
         down {
            # Connection is Down
            ifconfig "%% del 3ffe:b80:1ee4:b:1::1/80";
            ifconfig "%% down";
            program "/data/overdrive/delgprsroute.sh";
         };
       }
```

- MR:

```
       options {
         port 5000; # Connect to this port.
         timeout 1; # General timeout

       # Path to various programs
         ppp /usr/sbin/pppd;
         ifconfig /sbin/ifconfig;
         route /sbin/route;
       }

       # UDP connection
       gprs-udp {
         pass abc; # Password
         persist yes; # Persist mode
         device tap0;
         up {
            # Connection is Up
            # Assign IP addresses.
            ifconfig "%% 10.48.11.1 netmask 255.255.255.0 mtu 1300";
            ifconfig "%% add 3ffe:b80:1ee4:b:1::2/80";
         };
         down {
            # Connection is Down
            ifconfig "%% del 3ffe:b80:1ee4:b:1::2/80";
            ifconfig "%% down";
         };
       }
```

D.21 *radvd.conf* on the VGGSN

```
interface tap2
{
   AdvSendAdvert on;
   prefix 3ffe:b80:1ee4:b::/64
   {
        AdvRouterAddr on;
   };
};
```

D.22 *vtund* configuration on the UMTS-FB

```
options {
  port 5000; # Connect to this port.
  timeout 1; # General timeout

# Path to various programs
  ppp /usr/sbin/pppd;
  ifconfig /sbin/ifconfig;
  route /sbin/route;
}

# UDP connection
umts-udp {
  pass abc; # Password
  persist yes; # Persist mode
  device tap1;
  up {
     # Connection is Up
     # Assign IP addresses.
     ifconfig "%% 10.48.12.1 netmask 255.255.255.0 mtu 1300";
     ifconfig "%% add 3ffe:b80:1ee4:b:2::2/80";
  };
  down {
     # Connection is Down
     ifconfig "%% del 3ffe:b80:1ee4:b:2::2/80";
     ifconfig "%% down";
  };
```

D.23 *vtund* extension on the VGGSN

```
# UDP connection
umts-udp {
  pass abc; # Password
  type ether;
  proto udp;
  keepalive yes;
  device tap3;
  compress lzo:9;
  up {
     # Connection is Up
     # Assign IP addresses.
     ifconfig "%% 10.48.12.2 netmask 255.255.255.0 mtu 1300";
     ifconfig "%% add 3ffe:b80:1ee4:b:2::1/80";
     program "/data/overdrive/setroute.sh";
  };
  down {
     # Connection is Down
     ifconfig "%% del 3ffe:b80:1ee4:b:2::1/80";
     ifconfig "%% down";
     program "/data/overdrive/delumtsroute.sh";
  };
}
```

D.24 *radvd.conf* on the UMTS-FB

```
interface eth0
{
   AdvSendAdvert on;
   prefix 3ffe:b80:1ee4:9::/64
   {
        AdvRouterAddr on;
   };
};
```