To appear in Proc. IEEE Computer Communications Workshop (CCW), Laguna Niguel, CA, October 2003

Prototype Implementation and Performance Evaluation of a QoS-Conditionalized Handoff Scheme for Mobile IPv6 Networks*

Axel Neumann
Telecommunication Networks Group
Technical University Berlin
neumann@ft.ee.tu-berlin.de

Xiaoming Fu
Telematics Group
University of Göttingen
fu@cs.uni-goettingen.de

Holger Karl
Telecommunication Networks Group
Technical University Berlin
karl@ee.tu-berlin.de

Abstract— **Future internetworks will include large numbers of portable devices moving among small, wireless cells. In order to support real-time applications, users demand seamless mobility and Quality-of-Service (QoS) provisioning. One approach towards a more flexible, customizable and scalable mobility architecture that also reduces signaling load and handoff latency results from the introduction of micro-mobility. Furthermore, by coupling QoS signaling and mobility management, QoS requirements can be negotiated without incurring significant additional signaling latency.**

This paper presents the prototype implementation and performance evaluation of such a QoS-enabled micro-mobility scheme, which is called "QoS-conditionalized handoff". We extended the Mobile IPv6 for Linux implementation to support the basic mode of Hierarchical Mobile IPv6 as the underlying micro-mobility mechanism. One problem that appeared during the implementation was the rather complex event handling in the mobile node; to enable a simple and generic way of event handling, a priority-based execution structure has been developed that can be easily adapted to various policies.

Our experimental results show that by this QoS-conditionalized handoff scheme, QoS-enabled handoffs can be achieved with a small amount of introduced latency compared to Hierarchical Mobile IPv6, which is much less than that of Mobile IPv6. It is further observed that fewer packets were lost and registration latency could be much more decreased when mobility management in the mobile node takes advantage of a movement detection mechanism to expedite the QoS-conditionalized handoff procedure.

I. INTRODUCTION

As a mobile node (MN) travels between wireless cells, data transfer between the MN and the correspondent node (CN) will be typically changed from an old to a new access router (AR). In most mobility solutions, this process involves changes of routing entries in the MN and the CN, in addition to some designated mobility agents (home agent and/or foreign agent), and is called a handoff. It must ensure that end-to-end connectivity is maintained in a seamless way despite the changed path.

The Mobile IPv6 (MIPv6) [1] protocol provides the necessary extensions to enable IPv6 with true and transparent mobility support for the applications. Upon every movement of the MN, however, MIPv6 typically introduces a rather long registration latency, since the MN may be distant from the CN and the Home Agent (HA), where routing entries need to be registered with a new Care-of-Address (CoA) by the MN's Binding Update (BU) messages. By micro-mobility, the handoff operations regarding latency and signaling overhead can be optimized, essentially by moving the point of re-routing from the HA/CN closer to the MN. Currently, the Hierarchical Mobile IPv6 (HMIPv6) protocol [2] is the IETF proposal for micro-mobility support in MIPv6. In HMIPv6, a new entity, the Mobility Anchor Point (MAP) is introduced that allows an MN to send only one BU to the MAP to register its new local CoA after a movement. Mobility can be completely hidden from all nodes outside the access network (AN) and the MAP performs like a local HA.

In addition to seamless handoff support, it is desirable to support QoS in mobile environments, more specifically, to (re-)establish the QoS treatment in the changed path in an efficient way. To meet this requirement, Shen *et al.* [3], [4], Talukdar *et al.* [5], Manner and Raatikainen [6] presented various schemes for extending RSVP [7] to support Mobile IP(v6). However, the (re-)establishment of QoS treatment in the new path in these schemes typically needs separate end-to-end RSVP signaling message exchanges, in addition to mobility management message exchanges, and results in a relatively long latency for the nodes along the new path to get appropriate QoS treatment and the difficulty of releasing the resource along the old path [8].

To reduce the latency of QoS-treatment in the new path, following existing in-band signaling approaches [9], [10], Festag *et al.* [11], [12] proposed a QoS-conditionalized handoff ("QCH" for short in this paper) scheme. This scheme allows to perform QoS (re-)establishment during the handoff process by way of piggybacking an IPv6 hop-by-hop option ("QoS-option", which conveys traffic specification and QoS requirements regarding the MN's flow) in the Binding Update (BU) and Binding Acknowledge (BA) messages. Unlike previous approaches, QCH not only uses this QoS option to trigger QoS control entities to (re-)establish appropriate QoS treatment

---

[*] Corresponding author: X. Fu, Institute for Informatics, Lotzestrasse 16-18, 37083 Göttingen, Germany. This work was supported in part by a research project sponsored by Information Communication Mobile, Siemens AG.

along the path, but also performs a handoff only when the QoS requirements can be met in all nodes along the new path. The design of the QCH scheme takes advantage of HMIPv6, where the MAP is selected as the convergence point of the old and new paths. When an MN moves into a cell covered by several ARs (including the old AR) in the same MAP domain, it can use such piggybacked messages to establish QoS treatment along the new path, meanwhile the real-time data is still transmitted along the old path, until a QoS-satisfying path is found upon a successful handoff. The QCH scheme also allows the MN to specify a QoS range (desired QoS and acceptable QoS) for the MN's flow.

In this paper, we present the prototype implementation and performance evaluation of the QCH scheme based on HMIPv6. The Mobile IPv6 for Linux (MIPL) implementation was extended to support the basic mode of HMIPv6 as well as the QCH scheme. The remainder of the paper is organized as follows. Section II gives an overview of our HMIPv6 and QCH implementations. Section III describes the functional extensions in related entities used to implement HMIPv6 and QCH. Finally, Section IV presents the experimental results and is followed by discussions and conclusions in Section V.

II. IMPLEMENTATION OVERVIEW

Our implementation consists of two parts: an HMIPv6 extension to MIPv6 and a QCH extension based on the HMIPv6 extension. In this section we present an overview of these extensions.

A. A Hierarchical Mobile IPv6 Implementation

Our HMIPv6 extension to MIPv6 is based on the Mobile IPv6 for Linux (MIPL) implementation and the Linux IPv6 router advertisement daemon (radvd). [1] Since no stable release software for the IETF HMIPv6 protocol was available, we extended the MIPL-0.9 for Linux kernel 2.4.7 and the radvd-0.7.1 by introducing HMIPv6 functionalities upon which the QCH scheme was developed. Our HMIPv6 implementation has been made publicly available [13]. We implemented the basic mode of the HMIPv6 protocol [2] by the following ways:

- Extended the MIPL HA function to implement HMIPv6 MAP and reused MIPL HA for HMIPv6 HA;
- Extended intermediate routers (IRs) with functionality to support MAP discovery;
- Extended the MIPL MN function to perform (and distinguish) micro-mobility and macro-mobility.

These extensions are described in detail in Section III-A.

B. A QoS-Enabled Micro-mobility Implementation

The QoS-conditionalized handoff (QCH) scheme is implemented and tested in a prototype system. Our prototype provides a flexible platform to evaluate different handoff schemes: standard MIPv6, HMIPv6, and the QCH scheme. Two types of QoS/mobility signaling messages defined in [11] have been implemented:

- *QoS+BU message:* a message carrying both BU option and QoS option;
- *QoS+BA message:* a message carrying both BA option and QoS option.

As shown in Fig. 1, the core components of the prototype system are the QoS daemons (which intercept and modify QoS signaling messages if necessary), admission control modules (which decide whether to accept a QoS request) and mobility management modules (for processing of binding options) in the MAP, the MN and QoS-enabled intermediate routers (including ARs) in the AN.

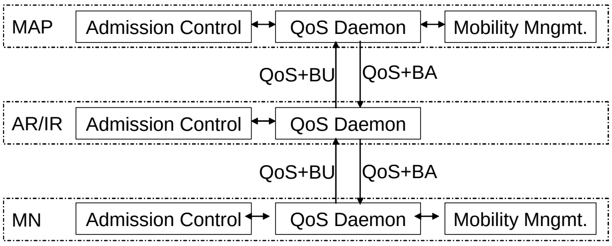

Fig. 1. A QoS-Enabled Micro-mobility Prototype Implementation

Without loss of generality, the prototype implementation of the QCH scheme only considers one important QoS parameter: bandwidth. Consequently, the admission control decision for an MN's flow is simply based on the relationship between the available bandwidth of the connected link and the acceptable and desired bandwidth requirements of the flow. No traffic control mechanism has been implemented, since it has nearly nothing to do with proof and evaluation of the QCH scheme.

III. IMPLEMENTATION DESIGN DETAILS

An existing MIPv6 implementation, MIPL [14], was chosen as basis for the implementation with three important extensions. First, the MAP and the MN are extended with HMIPv6 basic mode micro-mobility support. Second, BU and BA messages are extended to piggyback a QoS option (which become BU+QoS and BA+QoS messages). Finally, MAP, intermediate routers (IRs) and MN are extended with resource data structure and a simple admission control module.

The QoS daemon was developed based on Netfilter [15], to intercept and process the signaling messages in kernel space. The interaction between the QoS daemon, admission control and the mobility management module is achieved via a special shared resource data structure (RDS) which maintains both available and (for the MN' flows) reserved QoS in the MAPs and IRs; the MN's RDS further maintains requested and reserved QoS for each flow. An interface based on the Linux proc file system was introduced to allow access to certain QoS parameters from the user space.

The following subsections describe the implementation design issues of these extensions in all related entities involved with micro-mobility. A more detailed description can be found in [16].

---

[1] MIPL and radvd are open source under GPL and are available at http://www.mipl.mediapoli.com/ and http://v6web.litech.org/radvd/.

A. Hierarchical MIPv6 Implementation Based on MIPL

1) Mobile Node: Several design issues arise for the MN to support HMIPv6 (and QCH) and make it more complex than other entities. Fig. 2 and Fig. 3 show the main data structures and functional extensions to the MIPL implementation used for handling of events.

We observed that MN's movement event handling is very complex, especially the proper management of movement detection, selection of a suitable new default router, and the execution of related registration operations proved to be a difficult task. A new method, based on the concept of a so-called "context field", has been designed to overcome this challenge. Since the overall and quite complex connection state of the MN is distributed in a number of pointers, address and lifetime fields, which are necessary for maintaining the status of each known AR and its related path, a context field per AR is introduced to summarize these volatile properties and rank their priorities in one single field for each AR. Each known property for an AR is represented by a certain flag in the "context field". Additionally, these flags are arranged in an order (depending on the policy) such that the AR with the largest value in the context field is regarded as the currently most promising router. This selection method allows to easily apply different movement policies (this can be achieved by changing the order in which properties are reflected in the context field or by extending it with new flags to consider new information). Fig. 4 gives an overview of the context field and its related functions and properties.

Furthermore, registration operations are executed through the change_router() function which disseminates functionality into further procedures, again exploiting the clear state representation given through related bits in the context field to identify outstanding registration operations.

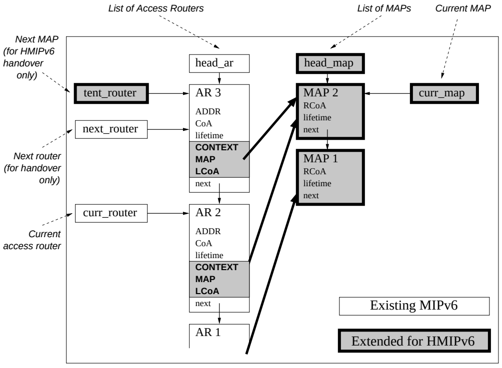

Fig. 2. The Mobile Node Data Structure for Mobility Management

2) Mobility Anchor Point and Intermediate Routers: In HMIPv6 a MAP in basic mode works essentially like an HA in such a way that it intercepts packets destined for a registered MN's RCoA and tunnels them to the MN's current location as identified by its LCoA. Therefore, an HA implementation can be used for a MAP but requires extensions for MAP discovery, to inform MNs about the presence of the MAP.

Since MAP discovery is an extension to the Neighbor Discovery protocol [17], and in particular to the part responsible for router discovery, extensions to the code providing this service is sufficient. In the Linux operating system, sending Router Advertisements (RtAdvs) is performed by the Router Advertisement Daemon [18]. This program runs in user space of a Linux IPv6 router and can be configured to initiate router discovery via its interfaces. It has been extended for the MAP discovery service which includes advertising of a MAP's own MAP option as well as the receiving and propagation of options from other MAPs. Therefore, the described functionality in this section fits also for intermediate routers by assuring the propagation of received MAP options. The desired behavior of such an entity eventually depends only on the configuration.

B. QoS-Conditionalized Handoff Implementation

1) Mobile Node: The MN was designed to be able to initiate QoS/mobility signaling messages through the data path in the visited AN. In the QCH implementation, a path is represented by its leaf node, the corresponding AR. The maintenance of requested and successfully reserved resources necessitates an extension to the aforementioned context field. Additionally, the MN's proc file system has been extended to dynamically pass flow and resource request parameters from user space to the related functionalities in the corresponding

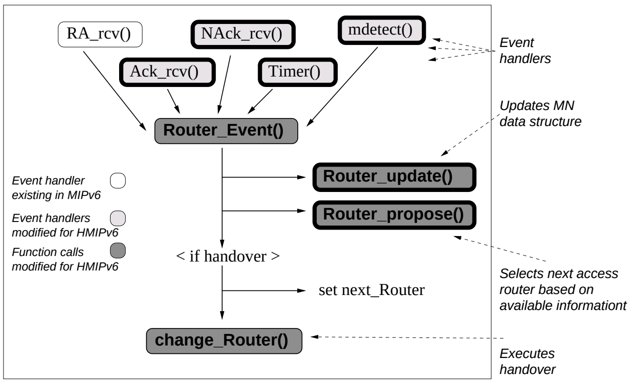

Fig. 3. The Mobile Node Event Handling

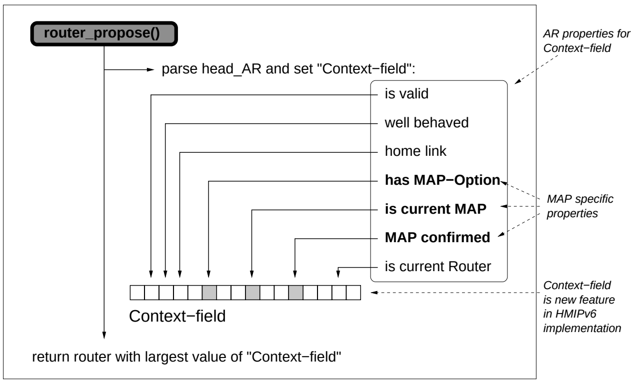

Fig. 4. The Mobile Node Context Field

kernel module.

2) Mobility Anchor Point and Intermediate Routers: A new resource structure and a simple admission control module were introduced in MAP and intermediate routers concerned with QoS control. Each structure element holds flow identification parameters, the affiliated QoS requirements and the lifetime status as deduced from the corresponding QoS option. Desired, minimal QoS indicated in the QoS option and current local resource information are used for admission control; reserved and available resource information are stored in the resource structure which can be accessed by both local admission control module and the QoS daemon. The QoS daemon processes QoS options and provides adequate interfaces with admission control and mobility management modules. When a QoS reservation failure occurs (e.g., due to insufficient resource availability or the policy does not grant them), a QoS+BA message with negative flag being set is generated and triggers the removal of QoS reservations in intermediate routers which have been made before.

IV. EXPERIMENTAL RESULTS

A. Experimental Setup

To evaluate the HMIPv6 and QCH schemes' performance in comparison with MIPv6, we established an experimental setup of the QoS-conditionalized handoff prototype system, which is shown in Fig. 5. Particularly, we used a manageable SNMP hub to perform emulated movements between different access routers (ARs) and prevent the noise typically introduced by a wireless link. Furthermore, a "virtual switch", which tries to minimize influences not related to our protocol implementation itself, was developed to measure the performance that would be achievable with ideally link-layer triggered handoffs. When a movement occurred, such a virtual switch triggers the MN to issue a Router Solicitation message and perform required mobility operations immediately rather than relying on the lifetime of its currently active default router (and the related last received Router Advertisement). The virtual switch is based on another Netfilter module developed in the MN kernel space. It essentially controls which of the received packets are to be dropped or accepted for further processing and thereby pretends connectivity or no connectivity to a range of physically available ARs. This virtual switch thus simulates the behavior of a link layer trigger that immediately observes the loss or establishment of link connections to ARs.

A WAN emulator is placed between the CN/HA and the MAP to emulate the transmission latency and loss over a WAN; also there is another WAN emulator between the AR and the MAP used to emulate transmission delay in the AN. Since no IPv6 enabled WAN emulator was available, to enforce some controllable delay to traversing packets, we used the NIST Net Delay tool for IPv4 [19] and built an IPv6 over IPv4 tunnel to constitute an IPv6 WAN emulator. We also developed a handy tool to generate traffic and perform measurements for our setup, which is called udp6traffic. Further details of the setup and used tools are described in [16].

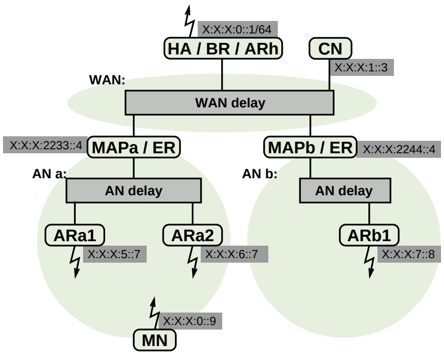

Fig. 5. A Testbed Setup for HMIPv6 and QCH Implementations

We evaluated the HMIPv6 implementation and the QCH scheme by two performance measures: registration latency and loss rate during the MIPv6, HMIPv6 and QCH handoffs (with and without the virtual switch) and the results are reported as follows.

B. Registration Latency

Table I summarizes the average, minimum, and maximum recorded registration latency from 10 handoffs for HMIPv6, QCH, and MIPv6 scenarios. For this purpose the registration latency is specified as the duration of the whole registration process depending on the handoff scenario (registration including reception of corresponding acknowledgement with MAP in case of a micro-mobility HMIPv6 or QCH handoff, MAP and HA in case of a macro-mobility HMIPv6 handoff, and HA in case of a MIPv6 handoff) but not including the time required for movement detection.

In addition, the total artificially controlled transmission delays applied to the registration messages by the WAN and AN emulators are given in the column delay, and the standard deviations are given in the column s.

The experimental results achieved with this setup show that the measured signaling latencies mostly depend on the delays enforced by the WAN and AN emulators. Delay caused by neighbor discovery and packet processing in the involved nodes is only of minor importance. The results also demonstrate that HMIPv6 outperforms MIPv6 in local movements and causes only a small increase of signaling latency in a global movement. Additionally, it could be shown that a QoS-conditionalized handoff compared to an HMIPv6 handoff introduces only a small increase of processing delay in micro-mobility scenarios. However, depending on the distance between the AN and the CN, it can be much faster than a MIPv6 handoff.

C. Packet Loss During Handoff

The minimum, maximum, and average packet loss of 100 handoff measurements for each mobility scenario is presented in Table II.

The metric used for the measurements is as follows: udp6traffic was configured to send packets up- and downstream between the MN and the CN with an interval of 10 ms

	delay	min	max	average	*s*
MIPv6	70ms	70.2ms	70.4ms	70.26ms	0.08ms
HMIP1	10ms	10.3ms	10.5ms	10.42ms	0.06ms
HMIP2	80ms	81.1ms	82.8ms	81.4ms	0.51ms
QCH	10ms	11.4ms	12.2ms	11.8ms	0.18ms

TABLE I

Experimental results for registration latency for different handoff scenarios (WAN delay: 30ms, AN delay: 5ms. HMIP1 – HMIPv6 micro-mobility, HMIP2 – HMIPv6 macro-mobility)

and a UDP payload size of 1000 bytes for a duration of 20 seconds. Within each 20 seconds slot one overlapping handoff was forced by enabling and disabling related ports of the manageable hub.

	min	max	average	*s*
MIPv6	18ms	163ms	104.4ms	36.8ms
HMIP1	13ms	155ms	87.6ms	35.0ms
HMIP2	19ms	165ms	97.9ms	40.7ms
QCH	18ms	158ms	92.1ms	39.6ms

TABLE II

Experimental results downstream packet losses (for a 10ms emission period) for different handoff scenarios (movement detection based on router advertisement period of 1.5 seconds, WAN delay: 30ms, AN delay: 5ms. HMIP1 – HMIPv6 micro-mobility, HMIP2 – HMIPv6 macro-mobility)

The results given in Table II show that as long as the MN's movement detection relies only on the RtAdvs' lifetime and retransmission time of 1.5 seconds, no significant benefits regarding the application packet loss can be measured or expected. The numbers of the last column of Table II also are relatively large, this is because without a fine control to propagate RtAdv immediately after a handoff happens, the timing out of previous RtAdv and the arrival of next RtAdv conforms to uniform distribution and thus follows the corresponding binding update processing.

Nevertheless,with focus on the measured registration latencies given in Table I, it also shows that using HMIPv6 and QCH in combination with an improved movement detection mechanism potentially results in much fewer packet losses compared to MIPv6.

D. Ideal Link Layer Trigger Case

The previously described experiments and their discussion regarding the application packet loss during a handoff revealed the following fact: as long as the execution of the MN's registration activities relies on a movement detection algorithm that is much more time-intensive than the actual protocol-dependent registration latency, no significant differences could be expected.

The experimental results illustrated in Fig. 6 show the measured handoff loss depending on the handoff frequency. The handoff loss is represented in percent divided by its current handoff frequency. However, this metric could also be regarded as the absolute packet loss per handoff or as the transmission interruption time per handoff in units of 10 ms. These results are achieved by using the virtual switch and are given for the mobility scenarios and protocols of a MIPv6, micro- and macro-mobility in HMIPv6, and the micro-mobility QoS-conditionalized handoff.

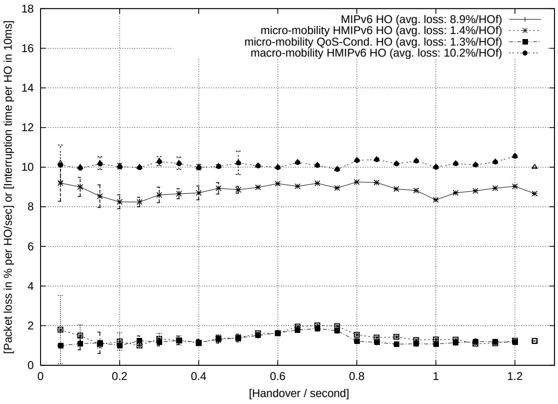

Fig. 6. Experimental results for downstream packet loss in MIPv6, HMIPv6, and QoS-conditionalized handoff scenarios over handoff rate (WAN delay: 30ms, AN delay: 5ms, movement detection delay reduced to less than 1ms by using a virtual switch)

For this set of measurements the WAN / AN delay was set to 30ms / 5ms. Unlike the previously described handoff measurements, the handoffs performed here were non-overlapping handoffs, also referred to as hard handoff. For a handoff (HO) in this setup it means that at the very moment when the link from the old AR is virtually disabled the link from the new one is enabled. Each illustrated point in the graph represents the average packet loss, measured during one 100-second slot. One complete measurement lasts for 25 slots where the handoff rate has been gradually increased by 5 handoffs per slot, starting with 5 handoffs per slot (0.05 HO / sec) to finally 125 handoffs per slot (1.25 HO / sec). The unit [number of HO per sec] is also referred to as handoff rate or handoff frequency (HOf). Fig. 6 shows also that the packet loss per HO is independent of the different handoff frequencies. Based on this knowledge we averaged the measured packet loss for all performed handoff frequencies and focused the following measurements on different transmission delays in the WAN simulator. The results are illustrated in Fig. 7.

E. Summary of Experimental Results

To summarize, these performance results show that in the micro mobility scenarios, the QCH scheme only introduces a very small amount of handoff latency and loss rate over the HMIPv6 scheme, which in turn is much less than the MIPv6 scheme. On the other hand, HMIPv6 causes only a small increase of registration latency for macro movements.

Regarding the application packet loss, it could be concluded that an improved movement detection mechanism (improved over the simple reliance on IPv6 RtAdvs) is obligatory to make use of the enhanced registration latency.

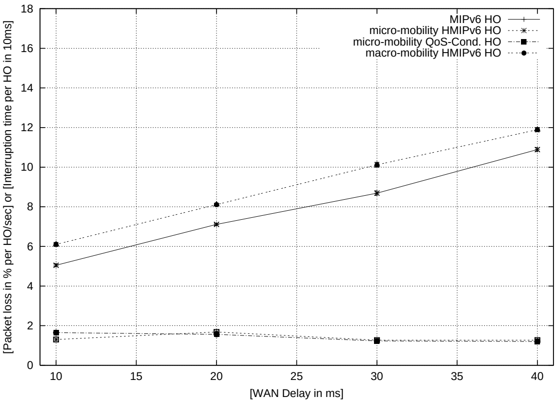

Fig. 7. Experimental results for downstream packet loss in MIPv6, HMIPv6, and QoS-conditionalized handoff scenarios over WAN delay (AN delay: 5ms, movement detection delay reduced to less than 1ms by using a virtual switch)

By simulating an ideal link layer trigger, it has been demonstrated that the downlink packet loss for micro movements could be reduced by over 80% per handoff (depending on the transmission delay between the AN and the CN) when using a protocol based on HMIPv6 instead of MIPv6. This is even the case when performing a QoS-conditionalized handoff: the performance penalty for QoS functionality is actually negligible. Measurements, comparing a macro movement between HMIPv6 and MIPv6, resulted in an increase of packet loss per handoff for at most 20% using the hierarchical approach.

V. DISCUSSIONS AND CONCLUSIONS

In this paper, the design and implementation of the QoS-conditionalized handoff (QCH) scheme for Mobile IPv6 has been presented. A prototype system implementing HMIPv6 and QCH extensions to Mobile IPv6 has been developed.

Performance experiments showed that the packet loss per handoff for micro movements can be reduced by more than 80 percent (depending on the transmission delay between the AN and the CN) when using HMIPv6 or QCH instead of MIPv6. However, it was also shown that an enhanced movement detection mechanism is necessary to let the application layer benefit from the reduced signaling latency introduced by the use of HMIPv6.

Nevertheless, a number of open issues still exist. In particular, as a QoS signaling scheme for micro-mobility, QCH needs to interact with an end-to-end QoS signaling solution; the latter is currently still under investigation, e.g., in the IETF NSIS working group [20]. Additional challenges emerge when considering QoS support for multiple, independent flows or when allowing some of these flows to traverse via different MAPs, which might also lead to asymmetric paths in the access network.

Finally, it can be concluded that the HMIPv6 and QCH implementations presented here provide a flexible and extensible prototype for testing and evaluating the HMIPv6 protocol and the QCH scheme. The main result is that the deployment of these schemes can provide significant performance advantages over MIPv6 even when QoS provisioning is desired.

REFERENCES

[1] D. Johnson, C. Perkins, and J. Arkko, “Mobility Support in IPv6,” Internet draft, work in progress, Jan. 2003.

[2] H. Soliman, C. Castelluccia, K. El-Malki, and L. Bellier, “Hierarchical MIPv6 Mobility Management,” Internet draft, work in progress, July 2002.

[3] Q. Shen, A. Lo, W. Seah, and C. Ko, “On Providing Flow Transparent Mobility Support for IPv6-based Real-Time Services,” in *Proc. of MoMuC 2000*, Tokyo, Japan, Oct. 2000.

[4] C.Q. Shen, W. Seah, A. Lo, H. Zheng, and M. Greis, “Mobility Extensions to RSVP in an RSVP-Mobile IPv6 Framework,” Internet draft, work in progress, July 2002.

[5] A. Talukdar, B. Badrinath, and A. Acharya, “MRSVP: a Resource Reservation Protocol for an Integrated Services Network with Mobile Hosts,” *Wireless Networks*, vol. 7, no. 1, pp. 5–19, 2001.

[6] J. Manner and K. Raatikainen, “Extended Quality-of-Service for Mobile Networks,” in *IEEE/IFIP Ninth International Workshop on Quality of Service (IWQoS 2001)*, Karlsruhe, Germany, June 2001, pp. 275–280, Lecture Notes in Computer Science 2092, Springer-Verlag.

[7] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, “Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification,” *RFC 2205*, Sept. 1997.

[8] M. Thomas, “Analysis of Mobile IP and RSVP Interactions,” Internet draft, work in progress, Oct. 2002.

[9] S. Lee, G. Ahn, X. Zhang, and A. Campbell, “INSIGNIA: An IP-Based Quality of Service Framework for Mobile ad Hoc Networks,” *Journal of Parallel and Distributed Computing, Special issue on Wireless and Mobile Computing and Communications*, vol. 60, no. 4, pp. 374–406, Apr. 2000.

[10] H. Chaskar and R. Koodli, “QoS Support in Mobile IP Version 6,” in *IEEE Broadband Wireless Summit (Networld+Interop’2001)*, Las Vegas, USA, May 2001.

[11] A. Festag, X. Fu, H. Karl, G. Schaefer, C. Fan, C. Kappler, and M. Schramm, “QoS-Conditionalized Binding Update in Mobile IPv6,” Technical University Berlin, Tech. Rep. TKN-01-013, July 2001.

[12] X. Fu, H. Karl, and C. Kappler, “QoS-Conditionalized Handoff for Mobile IPv6,” in *Networking2002*, Pisa, Italy, May 2002, Lecture Notes in Computer Science 2345, Springer-Verlag.

[13] TKN Hierarchical Mobile IPv6 Implementation on Linux. [Online]. Available: http://www-tkn.ee.tu-berlin.de/research/hmip/

[14] MIPL Mobile IPv6 for Linux Implementation. [Online]. Available: http://www.mipl.mediapoli.com/

[15] Linux netfilter. [Online]. Available: http://www.netfilter.org/

[16] A. Neumann, “Prototypical Implementation and Experimental Testbed Setup of a QoS-Enabled Mobility Concept Based on HMIPv6,” Diploma thesis, Faculty of Electrical Engineering and Computer Science, Technical University Berlin, Nov. 2002.

[17] T. Narten, E. Nordmark, and W. Simpson, “Neighbor Discovery for IP Version 6 (IPv6),” *RFC 2461*, Dec. 1998.

[18] RADVD – Router Advertisement Daemon. [Online]. Available: http://v6web.litech.org/radvd/

[19] NIST Net Project. [Online]. Available: http://is2.antd.nist.gov/itg/nistnet/

[20] IETF Next Steps in Signaling (NSIS) Working Group. [Online]. Available: http://www.ietf.org/html.charters/nsis-charter.html