# QoS Support for Mobile Users using NSIS

Roland Bless and Martin Röhricht *

Institute of Telematics
Universität Karlsruhe (TH)
Zirkel 2, D–76128 Karlsruhe, Germany
Email: {bless, roehricht}@tm.uka.de

**Abstract** Resource reservations in the Internet become more and more important with the advent of real-time multimedia services like Voice-over-IP and IPTV. At the same time we see an increasing interest in accessing Internet services by using mobile devices. In this paper we describe how Quality-of-Service guarantees can be achieved in mobile environments across different domains using the *Next Steps in Signaling* (NSIS) framework. We provide an analysis of mobility scenarios in combination with QoS signaling and propose to use an additional node local *Flow Information Service* element that supplies the necessary mobility support within NSIS capable mobility-aware nodes. We show that reservations can be setup quickly along the new path after a handover happened. Even the tear down of the reservation of the old path after a successful handover is performed quickly.

**Keywords:** QoS, NSIS, Signaling, MobileIP

## 1 Introduction

Controlling resources in the Internet requires manipulation of state in network elements along the path of a given data flow. In order to install, maintain, or tear down state on nodes on a given path signaling must be performed accordingly. The *Resource reSerVation Protocol* (RSVP) was once designed as a signaling protocol for Quality-of-Service establishment in IP networks. In response to some deficiencies of RSVP the IETF working group *Next Steps in Signaling* (NSIS) [1] was formed to design a framework for generic signaling on the IP layer.

With the advent of bandwidth demanding Internet applications and multimedia streams such as video broadcasts, voice-over-IP, or IPTV a continuously growing need for Quality-of-Service (QoS) arises. NSIS elaborated a QoS signaling protocol as its first use case which enables applications to reserve resources along a given path.

As mobile devices are becoming increasingly powerful, mobility and mobile computing become more and more attractive. That rises the desire to have continuous network connectivity. Since the Internet Protocol was not designed to

---

* The authors would like to thank Max Laier for his implementation efforts and conceptual input. Part of this work was supported by Deutsche Telekom Laboratories and T-Systems within the ScaleNet project.

cope with mobility, MobileIP [2] was developed by the IETF in the mid 1990's as an extension in order to provide transparent mobility support of applications.

However, QoS resource reservations are established for one particular data path and are thus not aware of mobility. MobileIP on the other hand does not cover QoS mechanisms and is primarily concerned with the correct routing of data packets towards the mobile endpoint. Even though the NSIS framework was designed from the beginning with support for mobility scenarios in mind, only very basic protocol mechanisms—e. g. a session identifier—were specified to accomplish this goal.

This paper provides an analysis of mobility scenarios in combination with QoS signaling. We propose an additional node local *Flow Information Service* element that supplies the necessary mobility support within NSIS capable mobility-aware nodes. We show that this support enables the NSIS QoS signaling protocol to work well in mobile scenarios and that the implied overhead in terms of additional reservation setup latency is small compared to non-mobile scenarios.

Though an existing Internet-Draft [3] discusses some mobility aspects of the NSIS protocol suite, a careful and detailed analysis of mobility supported is, however, still required. For instance, there are only a few mobility scenarios elaborated and considerations as well as more practical guidance for implementers is missing.

We will briefly introduce the NSIS framework and the basic operation of MobileIP in the following section. The use of QoS NSLP is of particular interest and will be discussed in more detail along different mobility scenarios in Section 3. As a further step we will discuss and propose solutions to these problems before we outline the necessary design decisions. The applicability of the NSIS protocols in mobile scenarios will be proven by evaluation in Section 5.

## 2 NSIS-based Signaling and Mobility

In order to develop an understanding for the main problems of NSIS-based signaling in mobile environments, we introduce the Next Steps in Signaling protocol suite and the basic operation of MobileIP that is relevant for the findings and solutions discussed in the remainder of this paper.

### 2.1 The Next Steps in Signaling Framework

Reliable QoS guarantees along a path and across different networks can only be accomplished by means of (possibly aggregated) resource reservations at routers residing on this path. In order to negotiate, install, and maintain on-demand resource reservations, signaling and admission control must be performed. The NSIS framework [4] was designed to perform signaling in IP-based next generation networks and employs a two-layer approach by separating the transport of signaling messages from the signaling application logic.

The protocol stack is conceptually divided into two layers. The lower layer is called the *NSIS Transport Layer Protocol* (NTLP). It is responsible to discover the next NSIS capable node on the path which also supports the specific signaling application, to setup state between both nodes, and to transport the higher layer signaling messages. Instead of deploying a new set of transport and security protocols it makes use of independent and wide-spread protocols, such as UDP, TCP, or TCP with TLS. The *General Internet Signaling Transport* (GIST) [5] is a protocol that fulfills the requirements of an NTLP.

The NSIS framework basically distinguishes between the *initiator* of a signaling message exchange (e.g., the entity initiating a reservation), the *responder* to such a request, and the *sender* or *receiver* of the corresponding data flow. Thus, if a *receiver-initiated* reservation is performed, the data flow receiver initiates the resource reservation. We use this important distinction during the further discussion, especially the one in section 3.4.

The *NSIS Signaling Layer Protocol* (NSLP) builds the higher layer of the protocol stack. Unlike GIST, which operates only between two hops, the NSLP provides end-to-end signaling functionality. Currently there are two signaling applications defined, namely a QoS NSLP [6] and a NAT/FW NSLP. Whereas the former establishes state among nodes in order to fulfill resource reservation requests, the latter is concerned with configuration of NAT-Gateways and Firewalls.

### 2.2 Mobility Management by Mobile IP

Mobility in an IP-based network can be achieved by using MobileIPv6 [7] which was proposed by the IETF as an extension to the base IPv6 protocol. Figure 1 gives a conceptual overview of MobileIPv6's basic operations, where each cloud denotes a different IP network (could be also different provider domains or Autonomous Systems). A mobile node is assigned two IP addresses, one is the *Home Address* (HoA) that is assigned at the mobile node's home network and is used to identify the communication endpoints, i. e. all connections to and from the mobile node. In addition one or more *Care-of-Addresses* (CoA) are assigned to the mobile node that represent its current location. MobileIPv6 was designed to perform a transparent mapping between both address types, i. e. it hides the actual location and therefore the CoA of a mobile node from its communication peer, the *Correspondent Node* (CN). In order to fulfill this request two operational modes were specified: the *Tunnel mode*, which is used initially and whenever the CN is not MobileIPv6-aware, and the *Route-optimization mode*, which can be used whenever the CN is MobileIPv6-aware. In this case the MN and the CN establish a binding between CoA and HoA, which can then be used to send traffic directly from one endpoint to another, instead of redirecting it through the MN's home network. In order to differentiate route-optimized packets from normal packets, special IPv6 options in extension headers are used.

Tunnel mode is used initially and whenever the CN is not MobileIPv6-aware. The traffic from the MN is then tunneled to a specific service node within the MN's home network, called the *Home Agent* (HA). Once the HA retrieves a data

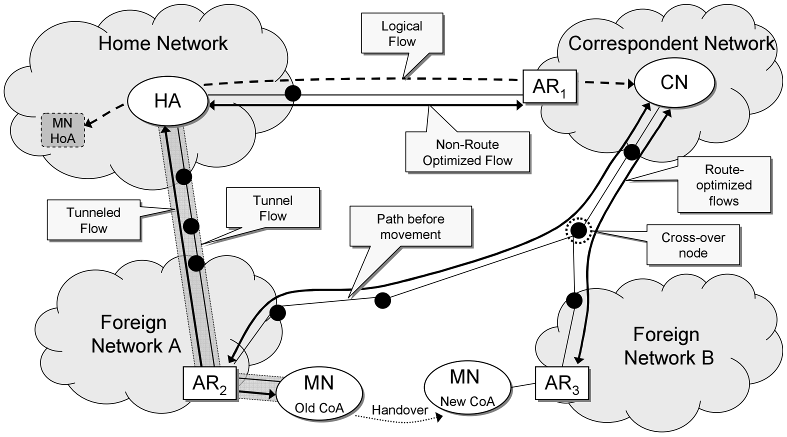


Figure 1: Basic operations of MobileIPv6

packet destined to the CN, it forwards the traffic on behalf of the MN. If the CN sends data towards the MN, the traffic is intercepted by the HA and forwarded using the established tunnel.

It is important to note, that the route-optimized flow is not related to the *logical flow*. Especially in case of signaling under these circumstances the signaling application must be aware of the *actual flow*.

## 3 Mobility Analysis for QoS NSLP

Now we briefly describe some issues that we found while trying to implement mobility support for QoS NSLP. Due to limited space we can present only some issues very briefly, whereas more details can be found in a technical report [8]. Solutions are discussed in Section 4.

### 3.1 Mobility Awareness

The NSIS mobility applicability draft [3] assumes that QoS NSLP or the application are mobility aware, e. g., that they know the current *Care-of-Address* (CoA). It proposes that the reservation refers to the actual current flow, so that a CoA is contained in the corresponding *Path-Coupled Message Routing Information* (PC-MRI), which describes all relevant addressing information of the concerned data flow. On the one hand such a scheme requires knowledge of the current CoA within GIST, QoS NSLP, and the application. On the other hand this approach contradicts the use of Mobile IP that hides mobility from applications and transport protocols. However, since Mobile IP adds overhead to IP layer packets, either in form of tunneling packets or by adding IPv6 extension headers, the QoS NSLP *must* be aware of a node's mobility to take this overhead into account[1] when requesting resources. More specifically, it must adjust the TMOD parameter of the QSPEC object [9] that conveys the necessary resources information, depending on whether the *Mobile Node* (MN) is in its home domain, in a foreign domain and/or using route optimization. In this respect it does not make sense trying to hide mobility from QoS NSLP.

### 3.2 Upstream Signaling

Path-coupled signaling, which is the default message routing method for GIST, makes it difficult to signal for a new flow $f_n$ in upstream direction (cf. Figure 2). This could be required in order to trigger a reservation request from the other end or cross-over node. Upstream forwarding of a message is a problem if no message routing state exists already in the next GIST hop. The upstream message would require the new flow's MRI and the resulting GIST Query must be sent in upstream direction. In this case there is the problem of choosing the correct encapsulation for the GIST Query: only upstream Q-mode encapsulation would be an option, but this is not appropriate to use in this case due to its limited applicability to certain environments (e. g., restricted topologies with only one default router). Even if the MN could send the NOTIFY to its new access router, there is no routing state installed yet in upstream direction in this node, i. e., it does not know any next peer in upstream direction.

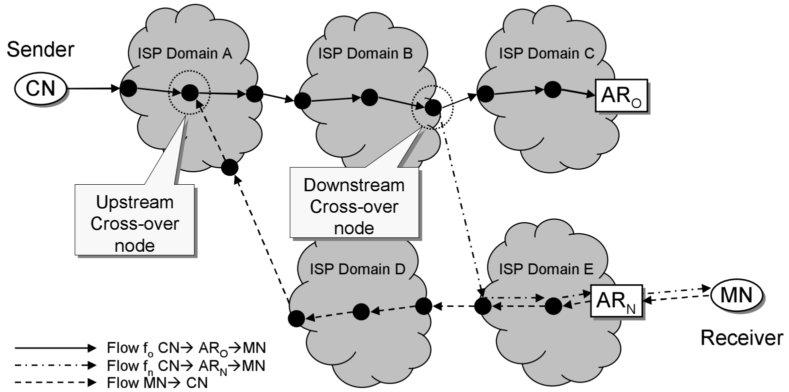


Figure 2: Example for signaling and data flows, upstream and downstream paths may be different

---

[1] even if this overhead is considered to be small in total, it may be relatively large when small packets such as Voice-over-IP packets are transmitted.

### 3.3 Resource Release

After performing a handover resources along the inactive path should be released as soon as possible in order to reduce reservation blocking for new reservation requests. Thus, there must be a possibility to release resources as soon as they are not used anymore. Though NSIS protocols use a soft-state approach that automatically removes unused state, it may be of advantage to release resources explicitly.

### 3.4 Mobility Scenarios

At least four different cases must be considered in the context of QoS NSLP:

1. Mobile node is sender and initiator
2. Mobile node is sender and responder
3. Mobile node is receiver and initiator
4. Mobile node is receiver and responder

The case of the MN being sender is not so difficult to handle, because mobility events may trigger appropriate QoS NSLP actions in order to adapt existing reservations. In case 1 the *MN* can initiate a new RESERVE message for the new flow $f_n$ directly after it has changed its point of attachment and got a new CoA. In case of a vertical handover the QSPEC may be adapted, but the Session ID will stay the same and a new PC-MRI corresponding to $f_n$ is used, so that the message can be sent downstream towards the *CN*. Even in case of an unchanged QSPEC, the signaling message must be forwarded up to the *CN*, because the PC-MRI has been changed and the related states must be updated along the whole path. In case 2 the *MN* could simply emit a new QUERY QoS NSLP message with a RESERVE-INIT flag set.

The case of the MN being receiver (scenarios 3 and 4) is more difficult, because it is especially hard to notify the sender ($CN$) of any change from the *MN*'s side using QoS NSLP as described in Section 3.2. Since the path from *CN* to *MN* is the downstream direction, the signaling path upstream from $AR_N$ towards the *CN* as source is not known in advance (cf. Figure 2). It is even difficult to determine the correct cross-over node, because the data path downstream (from $CN \rightarrow AR_N$) may be asymmetric to the upstream signaling path ($AR_N \rightarrow CN$). Furthermore, it may be often the case that the message must go back up along the old path to the sender in the worst case: If the flow address has changed, the profile in the first-hop router must be updated at sender side. In some cases the available QoS at the new point of attachment ($AR_N$) may be different from the one before at $AR_O$. In this case the changed resources require a re-negotiation along the whole path in most cases. Probably, in addition to that an application level signaling (e. g., SIP negotiation) is required in order to re-negotiate the content that should be sent or its coding, respectively.

### 3.5 Messaging Associations

In case an MN uses reliable or secure message transport, GIST establishes a Messaging Association to the next signaling peer, i. e., the next QoS NSLP aware node. In most cases this would be the access router. After changing the access router, the MN must establish a new MA to the new access router. Therefore, the new CoA must be used as source address, in order to avoid that the signaling connection is established via the home agent, which would be a long detour, resulting in long additional signaling delays in most cases.

## 4 Solutions to Mobility Problems

Since the solution space varies with the underlying assumptions about the environment, we have to distinguish between the following cases:

A. The application is mobility-aware and tries to manage mobility itself, i. e., no use of MobileIP, but SIP mobility support for instance.
B. Mobile Node (MN) and Correspondent Node (CN) are using MobileIPv6 and route optimization.
C. MN uses MobileIP, but the CN is not using MobileIP or does not support Route Optimization.

Case A does not impose any problems, because RESERVE and QUERY messages can be generated as soon as the MN moves. The CN can be notified about flow changes by some application level signaling protocol if it must send the RESERVE or QUERY.

In case B mobility events should be reported to GIST which in turn should notify the affected NSLPs. They can initiate new RESERVE/QUERY messages upon such a NetworkNotification. This works for all four cases of mobility scenarios that were described above in Section 3.4 and avoids the upstream signaling problem described in Section 3.2.

In case C the home agent must split the reservations and could act upon receiving binding updates from the MN by re-initiating reservations for the tunnel to the MN's current CoA as in the previous case. This HA-based solution also includes the case when MobileIPv4 is going to be used that does not support route optimization.

In general, we must deal with the case that nodes have probably QoS NSLP and MobileIPv6 support, but the QoS NSLP is not mobility-aware and consequently gets not notified about the corresponding events. In most cases, it is sufficient if the MN has such mobility support in the NSLP, because it can detect for instance that the HA is not initiating a reservation for the tunnel although a reservation request for a flow was received. In such a case the MN may request a reservation for the tunnel on its own. In some other cases a tunnel reservation will not be possible or reservations for the route optimized flow paths cannot be made.

With respect to the release of resources mentioned in Section 3.3, there are several options possible: the MN could resend a RESERVE with the "Replace" flag set so that the cross-over router would automatically initiate a tear down of the reservation along the old branch. Since a teardown can only be initiated in direction from the *QoS NSLP Initiator* (QNI) to the *QoS NSLP Responder* (QNR), the cross-over router can send a RESERVE with a Tear flag set only if the branch is downstream. Another option would be a NOTIFY message that is sent in order to initiate a tear down of the old branch from the right direction.

As sketched in Section 3.4, it is important for a QoS NSLP to be mobility aware: it must reserve the right amount of resources that depends on the current location and mode (tunnel or route optimization) due to the involved MobileIP packet overhead and it must know the current care-of-addresses in order to change the flow identifier for route optimized flows. Very important is also that mobility events, e. g., binding updates, trigger corresponding RESERVE or QUERY messages immediately in order to update the reservation accordingly. Therefore, an NSLP must be able to query the current state from the MobileIP mobility management and the MobileIP implementation must send notifications to the NSLP for significant events, such as handovers, new CoAs and a change in the binding cache and binding update list.

Although [10] specifies a MobileIPv6 Management Information Base (MIB) that would allow implementation independent access to the required information, this is not implemented in any of the MobileIPv6 implementations available to us at the time. Therefore, we created an additional component that provides the necessary additional information about actual flow addresses. For the QoS NSLP it is crucial to use the right CoA and the corresponding overhead values. Figure 3 shows the interaction of the *Flow Information Service* element with GIST and other NSLPs. The Flow Information Service is able to access state information directly from the MobileIPv6 implementation (MIPv6d). Basically, the same information could be supplied by using an SNMP agent that provides access to the MobileIPv6 MIB and SNMP traps could be used to realize mobility triggers.

The interface to the Flow Information Service is mainly based on a simple request/response interface. An NSLP entity sends a request indicating that it wishes to retrieve information about a certain flow. The Flow Information Service replies with the current state of that flow. In addition the Flow Information Service sends notifications whenever the state of an active flow changes. This way the consumer (the NSLP) is able to cache the results provided by the Flow Info Service, but does not need to bootstrap and mirror the complete state of the MobileIPv6 implementation. State information can also be polled every time the NSLP needs current flow addressing information.

As depicted in Figure 3 the request needs to contain a flow address as argument by specifying two IP-addresses (flow source and destination). This could be, for example, the address pair of a logical flow where one of the addresses is the home address. The reply needs to describe three possible flow states:

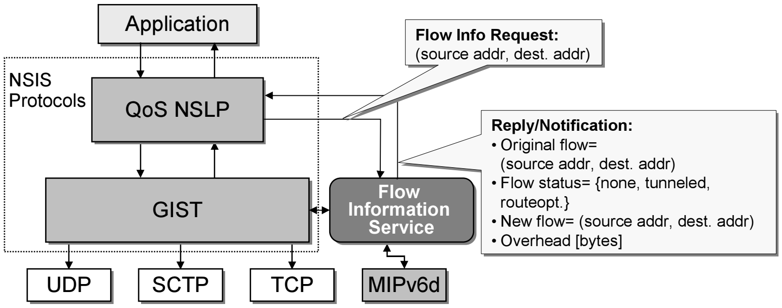


Figure 3: Flow Information Service

1. *No MobileIP flow*—because the source is not a HoA of the node and there is currently no active MobileIP state for the destination.[2]
2. *Tunnel mode*—the flow will enter or exit a tunnel at the current node, on the MN or the HA respectively. This happens when the flow source is a HoA and the peer is either MobileIP unaware or state is not established yet. The response must include the tunnel source and destination in order to enable the NSLP to establish a bound session for the tunnel section or update an existing session accordingly.
3. *Route optimization mode*—there is an active state for this flow and the flow source and/or flow destination will be rewritten. The information required in the reply consists of the new flow addresses.

Notifications reuse the reply message format and simply inform the requester (NSLP entity) about the new state of a flow. The requester must then internally identify the affected flow state information.

Furthermore, GIST is notified in case of mobility events and NSLPs are notified via the internal GIST API [5, Appendix B] primitive *NetworkNotification*. We defined a new *Network-Notification-Type* of *Mobility Event* for this purpose. Mobility-aware NSLPs may use this indication to request new information from the flow information service, for which further details can be found in [8].

## 5 Evaluation

The proposed Flow Information Service was implemented in the freely available NSIS-ka protocol suite [11] that is based on C++ and Linux. In order to evaluate the design we set up a testing environment consisting of six virtual hosts residing on one physical machine as shown in Figure 4. On each of them runs a slightly modified Linux kernel (2.6.26) to be MobileIPv6-aware. All virtual hosts are connected to a *smart switch* on another dedicated physical machine where the topology is set up by bridging VLAN interfaces accordingly.

---

[2] It might turn out later on that the peer is indeed a mobile node, but for the moment the flow is sent unmodified.

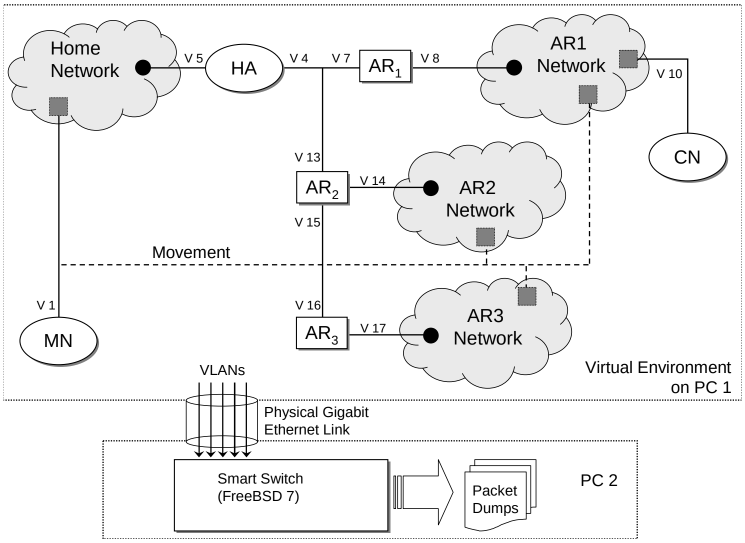


Figure 4: Layout of testing environment and assignment of VLAN interfaces

Given this setup we could easily obtain packet captures of all signaling messages exchanged between the hosts. Furthermore it allows for accurate and easy measurement of the delays obtained, resulting from a handover event and it was not necessary to take care of clock synchronization between each single host. As the smart switch runs on real hardware and the signaling messages travel over a real physical wire, virtualization should not affect the measurements beneficially.

### 5.1 Signaling Performance Benchmarks

The signaling performance was evaluated based on the time between receiving the *Binding Update/Binding Acknowledgment* on the MN/CN respectively and the time the final RESPONSE is received for the new reservation. This time was sampled for 48 consecutive movements of the MN from $\mathrm{AR}_3$ to $\mathrm{AR}_2$ to $\mathrm{AR}_1$ and back. Further we measured the time needed to tear down state on the old path between $\mathrm{AR}_2$ and $\mathrm{AR}_3$ in order to release resources not needed any longer. The results are shown in Table 1 representing the median of all runs. Note that no artificial delay was introduced, which results in round trip times of under 1 ms between the virtual hosts.

In order to setup reservation between the CN and the first hop $\mathrm{AR}_1$ we need less than 14 ms on average. In case of a reservation setup for $\mathrm{AR}_2$ and $\mathrm{AR}_3$ belonging to foreign networks, state is established in less than 34 ms and 44 ms respectively, no matter whether the MN is sender or receiver and whether sender- or receiver-initiated reservations are performed. The time needed to tear down state on the old path differs only—but significantly—for the case of the MN being a sender and by using sender-initiated reservations where approximately

| **Testcase** | **$AR_1$ setup** | **$AR_2$ setup** | **$AR_3$ setup** | **Tear** |
|---|---|---|---|---|
| MN sender, sender-initiated | 11.8 ms | 26.9 ms | 37.5 ms | 20600 ms |
| CN sender, sender-initiated | 13.3 ms | 27.4 ms | 40.3 ms | 26.8 ms |
| MN sender, receiver-initiated | 11.3 ms | 29.0 ms | 43.1 ms | 28.0 ms |
| CN sender, receiver-initiated | 12.5 ms | 33.4 ms | 42.2 ms | 31.9 ms |

Table 1: Median of the measurement results of reservation setup delays and old path tear down delays after movement

21 seconds are necessary. The time depends on the lifetime of the routing state between the MN at the old CoA and the $AR_3$ which times out eventually. This process could be sped up by actively acknowledging the NOTIFY on QoS NSLP level.

Figure 5 shows the measurement results for one of our four possible setups (MN is sender and uses receiver-initiated reservations). The standard deviation of our measurements shows, that the dispersion is relatively small, ranging from 2.7 ms for $AR_1$ to 7.4 ms for $AR_3$.

We set up a further benchmark by introducing artificial delays on the smart switch in order to simulate a more realistic Internet scenario. A 50 ms delay was configured between $AR_1$ and $AR_2$ and another 25 ms delay between $AR_2$ and $AR_3$. Given these settings, we obtained an RTT of 104 ms between the MN being located at $AR_2$ and the CN and 160 ms while the MN is located at $AR_3$ respectively.[3]

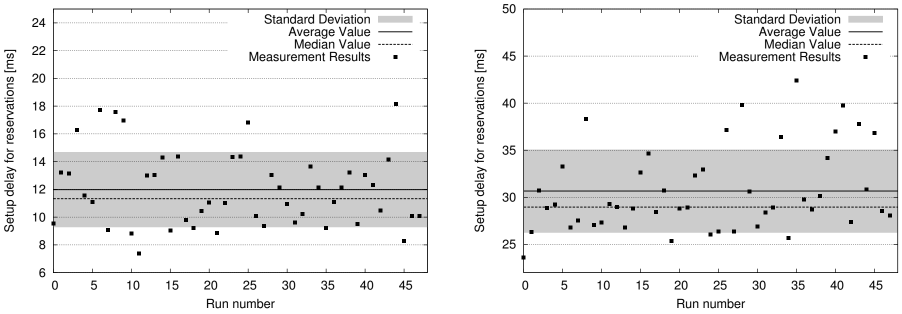
Results of measurements with additional artificial delays are shown in Table 2. The "optimal" delay that can be obtained by using the RTT values (2 RTT in case of sender-initiated mode, 2.5 RTT in case of receiver-initiated mode) are printed in parentheses after each measured value. Figure 6 illustrates the difference from the retrieved results compared to the theoretical possible performance for $AR_2$ and $AR_3$. The overhead incurred by our mobility-aware reservations ranges between 6.78% and 10.15% only.

| **Testcase** | **$AR_2$ setup delay (optimal value)** | **$AR_3$ setup delay (optimal value)** |
|---|---|---|
| MN sender, sender-initiated | 228.8 ms (208 ms) | 348.9 ms (320 ms) |
| CN sender, sender-initiated | 231.5 ms (208 ms) | 353.1 ms (320 ms) |
| MN sender, receiver-initiated | 285.5 ms (260 ms) | 429.3 ms (400 ms) |
| CN sender, receiver-initiated | 287.8 ms (260 ms) | 429.1 ms (400 ms) |

Table 2: Median of the measurement results of reservation setup delays with an artificial delay between different hosts

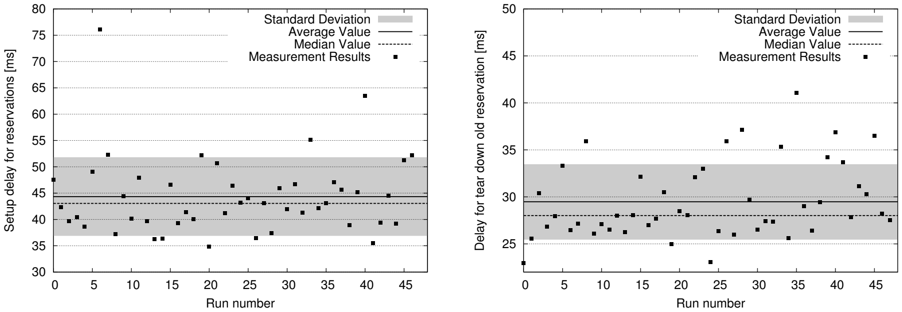
---

[3] The difference compared to theoretical 100 ms and 150 ms is due to scheduling granularity and queuing overhead on the smart switch.



(a) Access Router 1

(b) Access Router 2



(c) Access Router 3

(d) Tear down of old path

Figure 5: Measurement results for reservation setup and tear down delay if MN is sender and receiver-initiated reservations are used.

## 6 Conclusion

QoS NSLP works basically well in mobility scenarios if it has enough information about the actual data flow and gets the necessary mobility triggers. In this paper we introduced a node local component, the *Flow Information Service* element, that allows for getting required mappings from *logical* flows to *actual* flows (using current CoAs) as well as any mobility-related per packet overhead. In case a binding for a data flow changes the NTLP will notify any NSLPs of the change and mobility-aware NSLPs can request more information from the Flow Information Service if required. This makes it possible to re-reserve resources as soon as a MobileIP handover occurred.

Currently, we are working towards a seamless handover solution for NSIS using an anticipated handover concept. This requires, however, larger protocol changes of QoS NSLP and also additional support in GIST for providing path-decoupled signaling as proposed in [12].

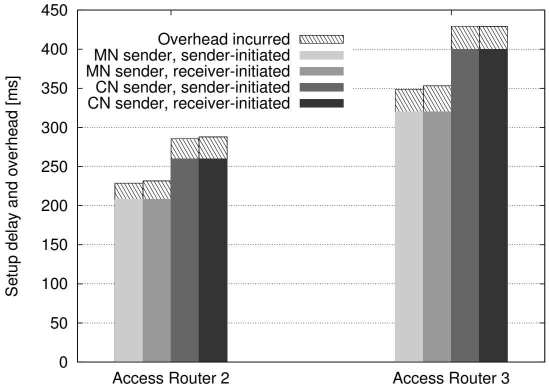


Figure 6: Setup delay and overhead of reservations for movement from $AR_2$ to $AR_3$

## References

1. Next Steps in Signaling (nsis) Working Group: NSIS Charter. Online: http://www.ietf.org/html.charters/nsis-charter.html (February 2009)
2. Perkins, C.E.: Mobile IP. Communications Magazine, IEEE **40**(5) (May 2002) 66–82
3. Sanda, T., Fu, X., Jeong, S.H., Manner, J., Tschofenig, H.: Applicability Statement of NSIS Protocols in Mobile Environments. Online: http://tools.ietf.org/html/draft-ietf-nsis-applicability-mobility-signaling (November 2008) Internet-Draft draft-ietf-nsis-applicability-mobility-signaling-11.
4. Fu, X., Schulzrinne, H., Bader, A., Hogrefe, D., Kappler, C., Karagiannis, G., Tschofenig, H., den Bosch, S.V.: NSIS: A New Extensible IP Signaling Protocol Suite. Communications Magazine, IEEE **43**(10) (October 2005) 133–141
5. Schulzrinne, H., Hancock, R.: GIST: General Internet Signalling Transport. Online: http://tools.ietf.org/id/draft-ietf-nsis-ntlp (October 2008) Internet Draft draft-ietf-nsis-ntlp-17.
6. Manner, J., Karagiannis, G., McDonald, A.: NSLP for Quality-of-Service Signaling. Online: http://tools.ietf.org/id/draft-ietf-nsis-qos-nslp (February 2008) Internet Draft draft-ietf-nsis-qos-nslp-16.
7. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6. RFC 3775 (Proposed Standard) (June 2004)
8. Laier, M.: Analysis and Design of Mobility Support for QoS NSLP. Technical Report TM-2009-1, Institute of Telematics, Universität Karlsruhe (February 2009) Telematics Technical Reports ISSN 1613-849X, http://doc.tm.uka.de/tr/TM-2009-1.pdf.
9. Ash, G., Bader, A., Kappler, C., Oran, D.: QoS NSLP QSPEC Template. Online: http://tools.ietf.org/html/draft-ietf-nsis-qspec-20 (April 2008) Internet-Draft draft-ietf-nsis-qspec-20.txt (Work in progress).
10. Keeni, G., Koide, K., Nagami, K., Gundavelli, S.: Mobile IPv6 Management Information Base. RFC 4295 (Proposed Standard) (April 2006)

11. Institute of Telematics: NSIS-ka – A free C++ implementation of NSIS protocols (February 2009) https://projekte.tm.uka.de/trac/NSIS/.
12. Bless, R.: An Explicit Signaling Target Message Routing Method (EST-MRM) for the General Internet Signaling Transport (GIST) Protocol. http://tools.ietf.org/html/draft-bless-nsis-est-mrm-01 (July 2008) Internet-Draft draft-bless-nsis-est-mrm-01.txt (Work in progress).