HyPath: an Approach for Hybrid On-path Off-path End-to-End Signaling

Luís Cordeiro, Vítor Bernardo, Marilia Curado, Edmundo Monteiro
Department of Informatics Engineering
University of Coimbra
{cordeiro, vmbern, marilia, edmundo}@dei.uc.pt

Abstract— **In a multi-domain Internet that offers Quality of Service guaranties, there is the need of signaling among the domain entities that are responsible for the management of Quality of Service. Because different domains have different network protocols and topologies, there is no solution that is able to signal these entities using an off-path approach and, in particular, that is able to interwork with the on-path signaling mechanisms. The HyPath approach extends the NSIS framework and its interactions with the local routing protocols to achieve off-path signaling in these hybrid environments. This document presents HyPath and its evaluation on a test-bed, showing that the mechanisms proposed have the potential to perform off-path signaling without introducing an excessive overhead in the network.**

Index Terms— **HyPath, off-path, signaling, NSIS, QoS, hybrid**

I. INTRODUCTION

NOWADAYS the Internet is part of everyone's life. In recent years, we have witnessed a rise of new technologies in the telecommunication and computer networks fields. This evolution led to the emergence of new types of applications involving multimedia, such as Voice over IP (VoIP), Video on Demand (VoD), tele-engineering and telemedicine. These applications have constraints and requirements concerning parameters such as delay and jitter. Therefore, the support of new services is required, besides those provided by the actual Internet. In this context it is necessary to address Quality of Service (QoS) issues, both at the intra and inter-domain levels. Signaling plays a key role in this scenario, and it has been the subject of extensive study. However, there are still several open issues, namely, end-to-end signaling across heterogeneous domains, which supports both on-path and off-path signaling. Usually, signaling messages are sent from one client to another through the same network path as the data, following thus the *on-path* signaling approach. Nevertheless, sometimes entities that are not in the data path need also to be signaled. Protocols that allow the signaling of entities that are not in the data path are called *off-path* signaling protocols.

The proposal presented in this paper aims at defining a global architecture to achieve QoS signaling within a multi-domain Internet context. Inside the domains, the QoS is managed through central entities, which are in charge of installing and handling QoS, based on internal rules. This concept was introduced in the Differentiated Services (DiffServ) domains, and is associated with Bandwidth Brokers [1]. At the present, a new requirement appears: signaling must take place, not only among devices strictly on the data path, but also among new domain-optimized central entities, that are called hereafter Resource Manager (RM).

Several signaling protocols have been proposed, especially in the Internet Engineering Task Force (IETF) Next Steps in Signaling (NSIS) working group [2]. The goal of the NSIS framework is to manipulate the network state related to data flows with the constraint that the signaling protocol will be processed on the nodes which also handle the data flows themselves ("path-coupled signaling"). This paper discusses a NSIS multi-domain, multi-service, RM based Internet that, besides path-coupled signaling, allows also off-path signaling.

This paper is structured as follows: Section 2 describes the NSIS framework, with emphasis on the GIST protocol; Section 3 presents the HyPath proposal, addressing two different ways of extending GIST to perform both on-path and off-path signaling; Section 4 describes the evaluation scenario and the performance results of the proposed schema; Finally, Section 5 draws the conclusions and highlights open issues to be addressed in future work.

II. NEXT STEPS IN SIGNALLING

Next Steps in Signaling is a new framework being developed in the NSIS Working Group of the IETF [3][4]. This group is responsible for standardizing an IP signaling protocol suite with QoS signaling as the first use case.

NSIS is structured as a two-layer modular solution, comprising the signaling transport layer and the signaling application layer. With this approach, the transport of the signaling messages and the signaling application are separated, which allows the framework to be used for more general purposes.

The NSIS Transport Layer Protocol (NTLP), named General Internet Signaling Transport (GIST) [5], is responsible for moving signaling messages among network entities. The NSIS Signaling Layer Protocol (NSLP) contains the specific functionalities of the signaling applications, namely the QoS-NSLP [6] and the Network Address

Translation (NAT) and Firewall (FW) NSLP [7]. Fig. 1 depicts the NSIS structure described.

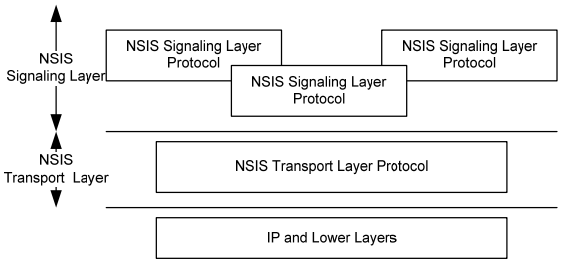

Fig. 1 – Two-layer protocol model

The GIST layer is responsible for the transport of signaling messages. When a signaling message is ready to be sent, it is given to the GIST layer along with information about the flow it is related to; it is then up to the GIST layer to get the message to the next network element (NE) along the path (downstream, in the flow direction from the source to the destination; or upstream, in the opposite direction of the flow, from the destination to the source), where it is received and the local GIST responsibility ends.

In the receiving NE, GIST either forwards the message directly to the next hop or, if there is an appropriate signaling application, passes it upwards for further processing; the signaling application can then generate another message to be sent via GIST.

GIST allows two modes of operation, the Datagram mode (D-mode) and the Connection mode (C-mode). D-mode uses UDP to encapsulate the messages and it is used for small and infrequent messages. All Query messages must be sent in D-mode. The C-mode uses TCP or any other stream or message oriented transport protocol (currently only the Stream Control Transmission Protocol (SCTP) [8][9], is being considered) which allows GIST to support reliability and security (for example using Transport Layer Security, TLS, [10] over TCP).

GIST defines a 3-way handshake to setup the necessary connection with the adjacent peers. This 3-way handshake contains a QUERY, a RESPONSE and an optional CONFIRM message. The QUERY message is the first to be sent. This message is always sent in D-mode and with the IP Router Alert Option (RAO) flag active. Then, the message travels along the network and every router that checks this flag analyses the packet content. GIST entities in the network analyze all packets flagged with IP RAO and process all QUERY messages.

When a QUERY message is intercepted, the NSLP ID is checked and, if the corresponding NSLP is present, the message is processed by GIST. Otherwise, the message is forwarded to the flow destination so that other GIST entities can intercept the message or the destination is reached.

The purposes of the QUERY message are the discovery of the next NSIS hop in the path and the transport of a proposal for the establishment of a connection between the two entities. GIST entities that receive a QUERY message need to reply with a RESPONSE message. This message is sent to the previous GIST entity by getting its identity from the QUERY message. If the received QUERY message requested an association, the RESPONSE message also includes the association response.

If the association between the two GIST entities was requested (by the NSLP or by a local GIST decision/configuration), when the RESPONSE message is received in the upstream GIST, the association is created, and a CONFIRM message is sent to the downstream GIST using the association. This association can be done through the mechanisms described earlier. Only after the CONFIRM message is sent, the NSLPs payload can start flowing between the two GIST entities.

The associations created via the 3-way handshake can be re-used for different sessions and NSLPs when the downstream peer and the association characteristics are the same. Even though the 3-way handshake is needed for each new session, the RESPONSE and CONFIRM messages are sent using the already established association.

GIST was designed as a soft-state protocol to manage all the messages and associations. GIST uses states for each action that occurs in the system and associates a timer to each state. Each time the state is updated, the timer is restarted. If the state is not updated, the timer expires and the state is removed. GIST has two main state tables: Message Routing State (MRS) and Message Association State (MAS). The MRS is responsible for managing individual flows and the MAS is responsible for managing the associations between individual peers. When a timer expires (if no message is received for the corresponding flow or association) the state is automatically removed from the state tables. If a state is required again, a new handshake is needed and a new association must be created.

After the handshake is completed, Data messages can be sent within the NSLP payload. GIST does not check the NSLP payload, and the only processing done to the message is the decrement of the message hop-count. After the corresponding states are refreshed (MRS and MAS), the payload is sent to the corresponding NSLP.

III. HYPATH PROPOSAL

The requirements for a hybrid on-path off-path approach for end-to-end signaling (HyPath) across NSIS and non-NSIS domains are not fully solved by the NSIS framework, as it is being currently defined in the IETF NSIS Working Group. The major requirements to achieve end-to-end network signaling are the following:

- Signaling messages must follow the same path as the user data;
- All the Resource Managers must be signaled.

Specifically, if the signaling is generated from an entity different than the data source, there is no mechanism that obliges it to go through the same path as the data path to the destination. Moreover, since RM servers are central entities in their domains, usually they are not in the data path. Therefore, an off-path approach is required to signal the RM. The HyPath solution [11] has been developed to fulfill the requirements above, empowering, in a transparent way, the NSIS framework with off-path signaling capabilities.

In a QoS scenario composed by tree distinct autonomous systems (AS) HyPath permits the off-path signaling of the RM entities, as shown in Fig. 2.

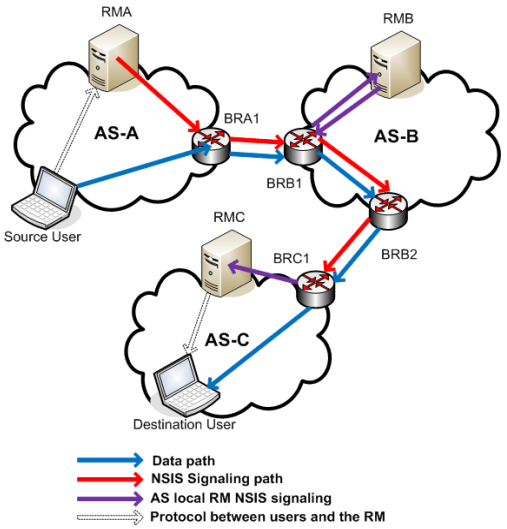

Fig. 2 – NSIS with HyPath support

In this scenario, the source user sends a request to the local RM, using a signaling protocol (independent from HyPath) to initiate HyPath signaling toward the destination user.

HyPath signaling starts in the RM A, where the NSLP requests HyPath to send a message from the Source User to the Destination User. Before signaling starts, the local data path egress border router needs to be discovered (using an external function). With this information, the RM A sends the message to the respective egress border router, BR A1.

When BR A1 receives the message it changes the received MRM with the original information available there (source IP address, destination IP address and direction) and sends the message towards the destination. This is the first message that follows the data path, like as if it had been generated by the source user.

In the next domain, AS B, the message is intercepted by the ingress border router, BR B1. This border router checks whether the message needs to be sent to the local RM. Since, in this case, the message was received from another domain (by checking the border router Address field), the message is forwarded to the local RM, RM B. In order to achieve this, the MRI information is changed by the HyPath mechanisms: the source is changed to the BR B1 IP address and the destination to the local RM IP address, RM B. The direction of the message is also set to downstream. All the original values are stored in the message HyPath additional fields.

When HyPath in RM B receives a Hypath message, sends it to the respective NSLP. If the NSLP sends a response message, it is sent back to the corresponding border router, BR B1. The RM B, before sending the message, changes the destination IP address to the BR B1 IP address and then sends the message. When BR B1 receives the message from the RM B, it injects it to the network, like in BR A1. The message is sent to the destination following the data path.

In AS B the message is intercepted by the egress border router, BR B2. In this border router, the message does not need to be sent to the local RM because the previous node is a local node (BR B1). This means that no processing is needed and the message is forwarded to the destination. Only the BR IP address must be updated.

In AS C the message is again intercepted by the ingress border router, BR C1. The ingress border routing procedure is the same as described above, in AS B, and the message is sent to the local RM (RM C).

RM C is treated as the last node because the destination user is in the RM network. Here, the HyPath signaling stops and another type of signaling (protocol independent from the HyPath, such as the Session Initiation Protocol [12]) can interact with the destination user.

When the NSLP requests a response message to a HyPath message in the RMs, the procedure is similar to the one described previously.

To implement these functionalities we have analyzed two different approaches:

- HyPath NSLP
- HyPath Extension

These two solutions have the same objective and the same capabilities. The difference between them is how they are integrated in the NSIS framework.

A. HyPath NSLP

The HyPath NSLP is a routing extension to the NSIS framework that connects to GIST and to NSLPs. To be able to connect the HyPath with the NTLP layer and the NSLP layer without changing their specifications, the HyPath is developed as a middle layer between these two layers.

Therefore, the HyPath interface with the NTLP layer must be the same as the NSLP layer interface already defined. Likewise the interface with the NSLP layer must be the same as the defined NTLP layer interface.

B. HyPath Extension

The HyPath Extension uses GIST extensibility capabilities to integrate off-path signaling in NSIS.

All GIST messages are composed by a sequence of type-length-value (TLV) objects. Those objects have a common header in which there are two bits for control of processing rules when a type is unknown for some peer. This pair of bits can be combined to oblige all nodes that are unaware of some TLV type, to do the forwarding of this information to the next peer, as described in [5].

This solution will aggregate all the required data to do the off-path signaling in a new TLV, called HyPath-TLV. Then, this new object will be included in all data messages received from NSLP. The usage of the GIST TLV A/B flags [5] in the HyPath-TLV ensures the forwarding of this new TLV in GIST entities that are unaware of HyPath.

With this approach, GIST processing remains unchanged, avoiding thus the problems that could occur if non-compliant HyPath peers rejected unidentified data. For example, in an autonomous system (AS) where all intra-domain routers have support for the NSIS framework, but are unaware of HyPath, the message will be forwarded, until it reaches some peer that

can process off-path data. Unlike the HyPath Extension solution, the HyPath NSLP does not change the already specified NSIS protocols, namely GIST and NSLPs. This solution takes advantage of the NSLP ID feature to handle non HyPath entities. On the other hand, the HyPath Extension solution, by being an extension to the GIST protocol, uses the GIST TLV extensibility option to extend the protocol.

IV. Experimental Evaluation

This section presents the evaluation of the two approaches for HyPath implementation within the NSIS framework, the HyPath NSLP and the HyPath Extension and compares the performance with normal NSIS operation.

The equivalent scenario to the one presented in Fig. 2 for the normal NSIS operation without HyPath is presented in Fig. 3. Due to the testing scenario configuration, since no alternative path was configured, the signaling messages will follow the same path as the data.

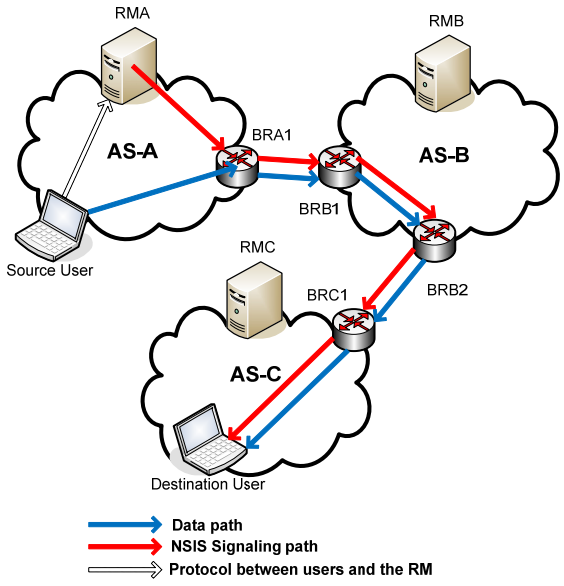

Fig. 3 – NSIS without HyPath support

In the HyPath unaware scenario, the signaling starts also in RM A, even though the message is sent to the destination, Destination User, and not to the BR A1. However, this router (BR A1) will intercept the message because the only way to send a message from Source User to Destination User is through it.

In AS B the message is intercepted by the ingress border router (BR B1). This router is also unaware of HyPath, so it will process the message and forward it to the Destination User. The local RM (RM B) will not be signaled.

When the message reaches the AS C, it is again intercepted by the ingress border router, BR C1. This border router will send the message to the Destination User. Thus, as happened in the other autonomous systems, the local RM (RM C) in not signaled.

The tests performed in this scenario and in the scenario with the HyPath solution described in Section III allowed the comparison of the overhead introduced by the HyPath proposal with the standard NSIS signaling.

The test conditions and the results obtained are described in the next sub-sections.

A. Test conditions

The tests were performed with the University of Coimbra implementation of the NSIS framework [13]. Performance tests to this GIST implementation are presented in [14]. The evaluation of another GIST implementation can be found in [15]. Currently, this implementation supports GIST version 15 and HyPath version 5. All the NSIS modules were implemented in Java 1.6, using Linux 2.6.

Although a QoS NSLP and NATFW NSLP implementations are available, we have implemented a new NSLP to perform the tests, the PerformanceNSLP, based on PingTool [16] and InteropNSLP [17]. This NSLP allows the creation of bundles of simultaneous session requests (1, 8, 32, 128, 256 and 512 messages). In every hop, this NSLP gathers the local IP address and then continues the signaling to the destination user. When the message reaches the destination user, the message is sent back to the source user. When the response reaches the source user, the signaling stops.

The equipment used to perform these tests was the following:

- Intel Celeron 1000Mhz
- 512MB Ram
- 80GB HDD

No tuning of the Java Virtual Machine was performed and no special Java Garbage Collector mechanism was implemented.

The results of the tests performed are presented in the next sub-section. Each result is an average of 5 tests performed with the same test conditions.

B. Results

The session setup time is the first parameter evaluated. This value represents the time needed to signal all the machines in the scenario and to respond back to the source.

Fig. 4 presents the session setup time for the three different scenarios described: GIST; GIST and HyPath NSLP; and GIST with the HyPath Extension.

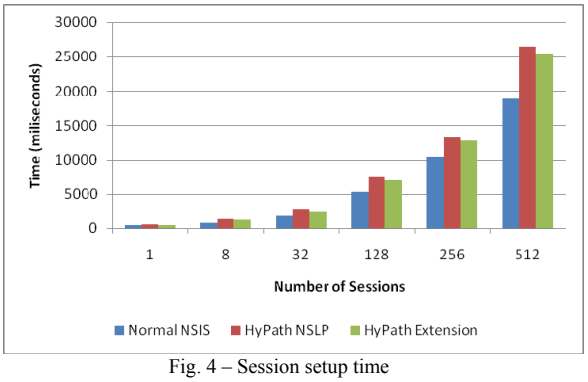

Fig. 4 – Session setup time

As expected, both HyPath approaches introduce some overhead in the session setup time, which attains ~35% in the case of 512 requests for the HyPath Extension. This overhead is due to the additional functions performed by HyPath and

also to the additional signaling of the RMs which increases the number of hops between the source and the destination.

In order to analyze the GIST protocol and the impact of the two HyPath alternatives with more detail, the processing time for specific GIST functionalities was measured, including Message Processing, Message Routing State (MRS) and Message Association State (MAS).

The Message Processing time is the time required by GIST to manage all the messages received either from the network or from the NSLPs. In the case of the HyPath NSLP, all NSLP Data Messages (i.e. messages where NSLPs put the data requested by applications) will carry HyPath specific information plus the usual NSLP payload. On the other hand, the HyPath Extension does not need to carry additional information in NSLP messages. This approach uses a new TLV object (i.e. HyPath TLV) to transport the required information.

Fig. 5 compares the time required by GIST to perform the Message processing using these two approaches.

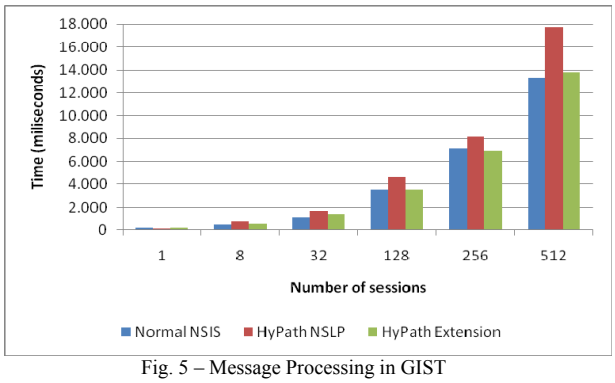

Fig. 5 – Message Processing in GIST

As it can be seen, the HyPath Extension introduces negligible overhead when compared to original NSIS, while the HyPath NSLP introduces a higher burden. As an example for 512 session setup messages HyPath Extension message processing introduces an overhead of ~4%.

The MRS time is the time required for the management of all the sessions. This parameter includes the GIST handshake process time and the network propagation time.

Fig. 6 compares the time required by GIST to handle the MRS using the two HyPath approaches and the normal NSIS.

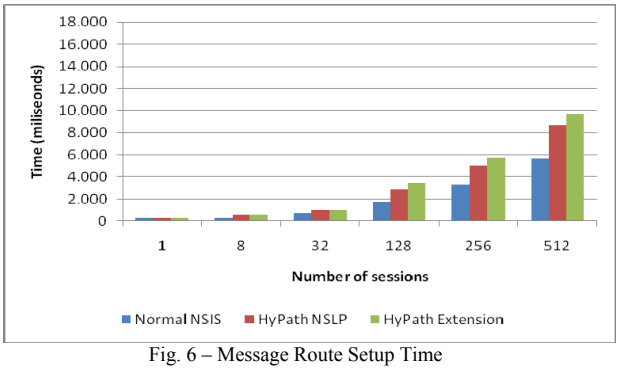

Fig. 6 – Message Route Setup Time

As it can be seen, the MRS time of the two HyPath proposals increase with the number of requests. These values are higher than the normal GIST mainly due to the signaling of new entities (i.e. local RMs) in the path. However, although the HyPath NSLP and the HyPath Extension will reach exactly the same machines, for high number of simultaneous sessions, the latter increases faster. This is due to the approach followed in the implementation of GIST. The HyPath NSLP implementation uses two Java Virtual Machines (JVM) to process the messages, so in an overload scenario, the performance is better. The HyPath Extension performs similar processing (compared to the HyPath NSLP) but in the same JVM, as GIST. These results could be improved by performing JVM tuning.

The MAS time is the time required for the creation of associations between individual peers. This association can be established in D-mode or C-mode, as explained previously. These tests were performed using a TCP connection. In this case, the MAS time is the time required by GIST to create a new TCP connection between adjacent nodes. All the tests were performed using the GIST association re-use functionality.

Fig. 7 compares the time required by GIST to handle the MAS using the two HyPath approaches. This figure shows the effect of HyPath NSLP and HyPath Extension in the GIST association mechanism. The values show that both HyPath implementations have a minimal impact on MAS due to the GIST association reuse feature.

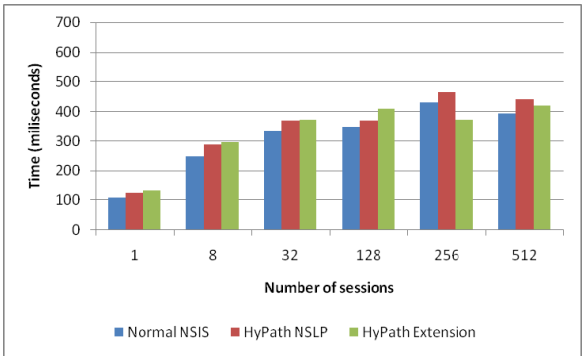

Fig. 7 – Message Association Time

In order to better evaluate the two HyPath approaches, the processing time was measured. In the case of the HyPath Extension, the HyPath processing time is the time required by GIST to process the new TLV object (only used in Data Messages). In the case of the HyPath NSLP, the HyPath processing time is the time between the reception of a message from the GIST layer and the transmission to the upper NSLP layer and the other way around.

Fig. 8 compares the time taken by both approaches.

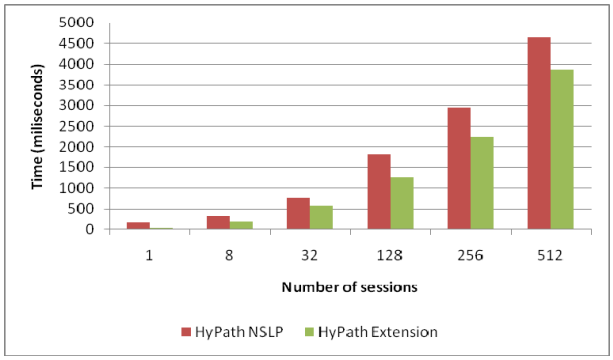

Fig. 8 – HyPath Processing Time

As can be seen in this figure, the HyPath NSLP introduces more processing time than the HyPath Extension. These results contradict the values in Fig. 6. This means that even though HyPath has better performance as a GIST extension, it has impact in the GIST MRS. Like stated before this can be a result of JVM limitations or limitations in the current GIST implementation.

V. Conclusion

This paper addresses the problem of the off-path signaling in heterogeneous networks and presents a solution called HyPath. This proposal specifies how to add off-path support in the NSIS framework. Two approaches to implement the HyPath proposal were also presented.

The proposed schemes for hybrid on-path off path end-to-end signaling were evaluated on an experimental test-bed in order to assess their performance. Namely, the overhead introduced by these two approaches was measured and compared to a normal NSIS scenario.

The HyPath Extension showed better results than the HyPath NSLP. In the tests performed the HyPath Extension needed less time to establish the session than the HyPath NSLP, even though the MRS and MAS times are some times higher using the HyPath Extension.

The results showed that the overhead introduced by the HyPath proposal is around 30% of the session setup time, in the worst case. Because this overhead includes the usage of 4 more signaling messages and the signaling of two more entities, the RMs, the overhead introduced by the proposal can be accommodated in the end-to-end signaling scenario.

Further studies will be done in order to evaluate in more detail the impact of the HyPath Extension in the GIST implementation and its impact in the NSIS signaling.

References

[1] Ben Teitelbaum, P. Chimento, “SIBBS: Simple Interdomain Bandwidth Broker Signaling”, QBone, September 1999.
[2] Internet Engineering Task Force Next Steps in Signaling working group. Available: http://www.ietf.org/html.charters/nsis-charter.html
[3] R. Hancock, G. Karagiannis, J. Loughney, S. Van den Bosch, “Next Steps in Signaling (NSIS): Framework”, RFC 4080, IETF, June 2005.
[4] X. Fu, H. Schulzrinne, A. Bader, D. Hogrefe, C. Kappler, G. Karagiannis, H. Tschofenig, S. Bosch, “NSIS: A New Extensible IP Signaling Protocol Suite”, IEEE Communications Magazine, Internet Technology Series, October 2005.
[5] H. Schulzrinne, R. Hancock, “GIST: General Internet Signaling Transport”, draft-ietf-nsis-ntlp, IETF, February 2008.
[6] J. Manner, G. Karagiannis, A. McDonald, “NSLP for Quality-of-Service Signaling”, draft-ietf-nsis-qos-nslp, IETF, February 2008.
[7] M. Stiemerling, H. Tschofenig, C. Aoun, E. Davies, “NAT/Firewall NSIS Signaling Layer Protocol (NSLP)”, draft-ietf-nsis-nslp-natfw, IETF, February 2008.
[8] R. Stewart, “Stream Control Transmission Protocol”, RFC 4960, IETF, September 2007.
[9] X. Fu, C. Dickmann, J. Crowcroft, “General Internet Signaling Transport (GIST) over SCTP”, draft-ietf-nsis-ntlp-sctp-04, IETF, February 2008.
[10] T. Dierks, C. Allen, “The TLS Protocol Version 1.0”, RFC 2246, IETF, January 1999.
[11] L. Cordeiro, M. Curado, E. Monteiro, V. Bernardo, D. Palma, F. Racaru, M. Diaz, C. Chassot, “GIST Extension for Hybrid On-path Off-path Signaling (HyPath)”, draft-cordeiro-nsis-hypath, IETF, February 2008.
[12] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, “SIP: Session Initiation Protocol”, RFC 3261, IETF, June 2002.
[13] NSIS Implementations of University of Coimbra. Available: http://nsis.dei.uc.pt
[14] B. Sousa, P. Neves, G. Leão, D. Palma, J. Silva, S. Sargento, F. Fontes, M. Curado, F. Boavida, “The Cost of Using IEEE 802.16d Dynamic Channel Configuration”, IEEE ICC 2008, May 2008.
[15] X. Fu, H. Schulzrinne, H. Tschofenig, C. Dickmann, D. Hogrefe, “Overhead and Performance Study of the General Internet Signaling Transport (GIST) Protocol”, IEEE INFOCOM 2006, April, 2006.
[16] C. Dickmann, I. Juchem, S. Willert, X. Fu, “A stateless Ping tool for simple tests of GIMPS implementations”, draft-juchem-nsis-ping-tool-02, July, 2005.
[17] X. Fu, I. Juchem, C. Dickmann, H. Tschofenig, “Design Options of NSIS Diagnostics NSLP”, draft-fu-nsis-diagnostics-nslp-01, March 6, 2006.