NSIS Working Group
Internet Draft

Robert Hancock (editor)
Siemens/Roke Manor Research
Ilya Freytsis
Cetacean Networks
Georgios Karagiannis
Ericsson
John Loughney
Nokia
Sven Van den Bosch
Alcatel

Document: draft-ietf-nsis-fw-02.txt
Expires: September 2003 March 2003

Next Steps in Signaling: Framework

Status of this Memo

Abstract

The Next Steps in Signaling working group is considering protocols for signaling information about a data flow along its path in the network. Based on existing work on signaling requirements, this document proposes an architectural framework for such signaling protocols.

This document provides a model for the network entities that take part in such signaling, and the relationship between signaling and the rest of network operation. We decompose the overall signaling protocol suite into a generic (lower) layer, with a separate upper

layers for each specific signaling application. An initial proposal for the split between these layers is given, describing the overall functionality of the lower layer, and discussing the ways that upper layer behavior can be adapted to specific signaling application requirements.

This framework also considers the general interactions between signaling and other network layer functions, specifically routing and mobility. The different routing and mobility events that impact signaling operation are described, along with how their handling should be divided between the generic and application-specific layers. Finally, an example signaling application (for Quality of Service) is described in more detail.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [2]. [Editor's note: if - as is likely - we don't end up using these words in the framework, this paragraph will disappear.]

Table of Contents

1. Introduction

1.1 Definition of the NSIS Signaling Problem

The NSIS working group is considering protocols for signaling information about a data flow along its path in the network.

It is assumed that the path taken by the data flow is already determined by network configuration and routing protocols, independent of the signaling itself; that is, signaling to set up the routes themselves is not considered. Instead, the signaling simply interacts with nodes along the data flow path. Additional simplifications are that the actual signaling messages pass directly through these nodes themselves; this is 'path-coupled' signaling in the sense described in [3], and that only unicast data flows are considered.

The signaling problem in this sense is very similar to that addressed by RSVP [4]. However, there are two generalizations. Firstly, the intention is that NSIS designs protocols that can be used in different parts of the Internet, for different needs, without requiring a complete end-to-end deployment (in particular, the signaling protocol messages may not need to run all the way between the data flow endpoints).

Secondly, the signaling is intended for more purposes than just QoS (resource reservation). The basic mechanism to achieve this flexibility is to divide the signaling protocol stack into two layers: a generic (lower) layer, and an upper layer specific to each signaling application. The scope of NSIS is to define both the generic protocol, and initially an upper layer suitable for QoS signaling similar to the corresponding functionality in RSVP. Further signaling applications may be considered later.

1.2 Scope and Structure of the NSIS Framework

The underlying requirements for signaling in the context of NSIS are defined in [3]; other related requirements can be found in [5] and [6]. This framework does not replace or update these requirements. Discussions about lessons to be learned from existing signaling and resource protocols are contained in a separate analysis document [7].

The role of this framework is to explain how NSIS signaling should work within the broader networking context, and how the signaling protocols should be structured at the overall level. Therefore, it discusses important protocol considerations, such as routing, mobility, security, and interactions with network 'resource' management (in the broadest sense).

The basic framework for NSIS is given in section 3. Section 3.1 describes the fundamental elements of NSIS operation in comparison to RSVP; in particular, section 3.1.2 describes more general signaling scenarios, and 3.1.3 defines a broader class of signaling applications for which the NSIS protocols should be useful. The two-layer protocol architecture that supports this generality is

described in section 3.2, and section 3.3 gives examples of the ways in which particular signaling application properties can be accommodated within signaling layer protocol behavior.

The overall functionality required from the lower (generic) protocol layer is described in section 4. This is not intended to define the protocol detailed design or even design options, although some are described as examples. The emphasis is on defining the interfaces between this lower layer protocol and the IP layer and signaling application protocols, including the identity elements that appear on these interfaces. Following this, section 5 describes how signaling applications that use the NSIS protocols can interact sensibly with network layer operations, specifically routing (and re-routing), IP mobility, and network address translation.

Section 6 describes particular signaling applications. The example of signaling for QoS (comparable to core RSVP QoS signaling functionality) is described in detail in section 6.1, which describes both the signaling application specific protocol and example modes of interaction with network resource management and other deployment aspects. However, note that these examples are included only as background and for explanation; it is not intended to define an over-arching architecture for carrying out resource management in the Internet. Further possible signaling applications are outlined in section 6.2.

2. Terminology

[Editor's note: it is a matter of taste where we put this.]

Classifier - an entity which selects packets based on their contents according to defined rules.

[Data] flow - a stream of packets from sender to receiver which is a distinguishable subset of a packet stream. Each flow is distinguished by some flow identifier (see section 4.5.1).

Edge node - a (NSIS-capable) node on the boundary of some administrative domain.

Interior nodes - the set of (NSIS-capable) nodes which form an administrative domain, excluding the edge nodes.

NSIS Entity (NE) - the function within a node which implements an NSIS protocol. In the case of path-coupled signaling, the NE will always be on the data path.

NSIS Signaling Layer Protocol (NSLP) - generic term for an NSIS protocol component that supports a specific signaling application. See also section 3.2.1.

NSIS Transport Layer Protocol (NTLP) - placeholder name for the NSIS protocol component that will support lower layer (signaling application independent) functions. See also section 3.2.1.

Path-coupled signaling - a mode of signaling where the signaling messages follow a path that is tied to the data messages.

Path-decoupled signaling - signaling - signaling for state manipulation related to data flows, but only loosely coupled to the data path, e.g. at the AS level.

Peer discovery - the act of locating and/or selecting which NSIS peer to carry out signaling exchanges with for a specific data flow.

Peer relationship - signaling relationship between two adjacent NSIS entities (i.e. NEs with no other NEs between them).

Receiver - the node in the network which is receiving the data packets in a flow.

Sender - the node in the network which is sending the data packets in a flow.

Session - application layer flow of information for which some network control state information is to be manipulated or monitored (see section 4.5.2).

Signaling application - the purpose of the NSIS signaling: a service could be QoS management, firewall control, and so on. Totally distinct from any specific user application.

3. Overview of Signaling Scenarios and Protocol Structure

3.1 Fundamental Signaling Concepts

3.1.1 Simple Network and Signaling Topology

The NSIS suite of protocols is envisioned to support various signaling applications that need to install and/or manipulate state in the network. This state is related to a data flow and is installed and maintained on the NSIS Entities (NEs) along the data flow path through the network; not every node has to contain an NE. The basic protocol concepts do not depend on the signaling application, but the details of operation and the information carried do. This section

discusses the basic entities involved with signaling as well as interfaces between them.

Two NSIS entities that communicate directly are said to be in a 'peer relationship'. This concept might loosely be described as an 'NSIS hop'; however, there is no implication that it corresponds to a single IP hop. Either or both NEs might store some state information about the other, but there is no assumption that they necessarily establish a long-term signaling connection between themselves.

It is common to consider a network as composed of various domains, e.g. for administrative or routing purposes, and the operation of signaling protocols may be influenced by these domain boundaries. However, it seems there is no reason to expect that an 'NSIS domain' should exactly overlap with an IP domain (AS, area) but it is likely that its boundaries would consist of boundaries (segments) of one or several IP domains.

Figure 1 shows a diagram of nearly the simplest possible signaling

configuration. A single data flow is running from an application in the sender to the receiver via routers R1, R2 and R3. Each host and two of the routers contain NEs which exchange signaling messages - possibly in both directions - about the flow. This scenario is essentially the same as that considered by RSVP for QoS signaling; the main difference is that we make no assumptions here about the particular sequence of signaling messages that will be invoked.

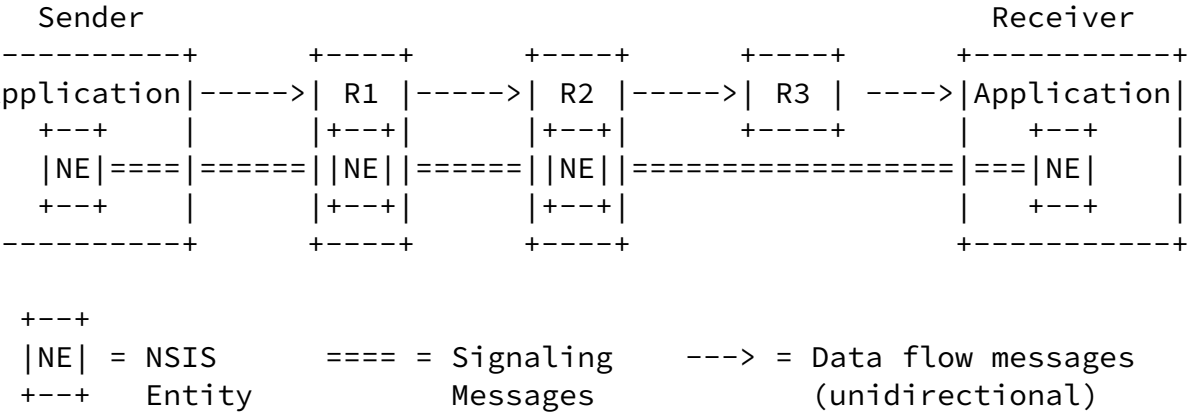


Figure 1: Simple Signaling and Data Flows

3.1.2 Signaling to Hosts, Networks and Proxies

There are different possible triggers for the NSIS signaling. Amongst them are signaling applications (that are using NSIS signaling services), other instances of the signaling, network management actions, some network events, and so on. The variety of possible

triggers requires that the signaling can be initiated and terminated in the different parts of the network - hosts, domain boundary nodes (edge nodes) or interior domain nodes.

NSIS extends the RSVP model to consider this wider variety of possible signaling exchanges. As well as the basic end-to-end model already described, examples such as end-to-edge and edge-to-edge can be considered. The edge-to-edge case might involve the edge nodes communicating directly, as well as via the interior nodes.

While end-to-edge (host-to-network) scenario requires only intra-domain signaling, the other cases might need inter-domain NSIS signaling as well if the signaling endpoints (hosts or network edges) are connected to different domains. Depending on the trust relation

between concatenated NSIS domains the edge-to-edge scenario might cover single domain or multiple concatenated NSIS domains. The latter case assumes the existence of the trust relation between domains.

In some cases it is desired to be able to initiate and/or terminate NSIS signaling not from the end host that sends/receives the data flow, but from the some other entities on the network that can be called signaling proxies. There could be various reasons for this: signaling on behalf of the end hosts that are not NSIS-aware, consolidation of the customer accounting (authentication, authorization) in respect to consumed application and transport resources, security considerations, limitation of the physical connection between host and network and so on. This configuration can be considered as a kind of "on the data path", see Figure 2.

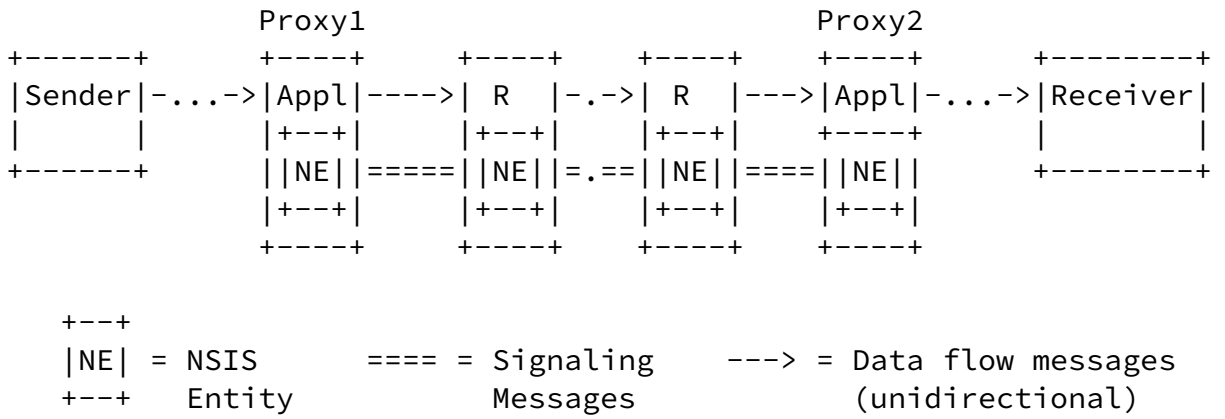


Figure 2: "On path" NSIS proxy

This configuration presents a set of specific challenges to the NSIS signaling:

*) The proxy that terminates signaling on behalf of the NSIS-unaware host (or part of the network) should be able to make determination that it is a last NSIS aware node along the path.
*) Where a proxy initiates NSIS signaling on behalf of the NSIS unaware host, interworking with some other "local" technology might be required, for example to provide QoS reservation from proxy to the end host in the case of QoS signaling application.

Another possible configuration, shown in Figure 3 is where an NE can

send and receive signaling information off path for and from remote processing. The NSIS protocols may or may not be suitable for this remote processing but in any case it is not currently part of the NSIS problem. This configuration is supported by considering the NE as a proxy at the signaling application level. This is a natural implementation approach for some policy control and centralized control architectures, see also section 6.1.5.

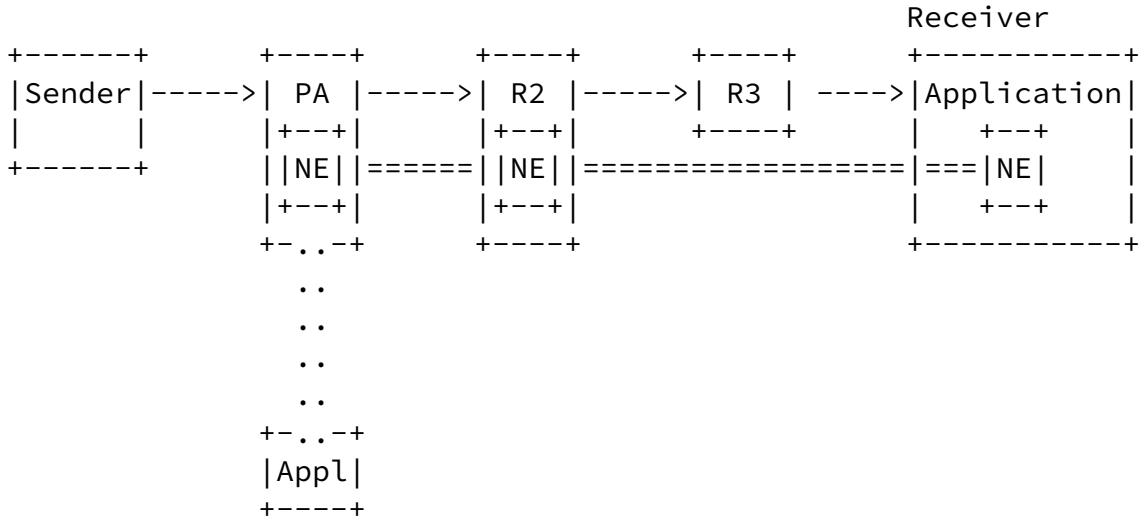


Figure 3: "Off path" NSIS proxy

3.1.3 Signaling Messages and Network Control State

The distinguishing features of the signaling supported by the NSIS protocols are that it is related to specific flows (rather than to network operation in general), and that it involves nodes in the network (rather than running transparently between the end hosts).

Therefore, each signaling application (upper layer) protocol must carry per-flow information for the aspects of network-internal operation corresponding to that signaling application. An example for the case of an RSVP-like QoS signaling application would be state

data representing resource reservations. However, more generally, the per-flow information might be related to some other control function in routers and middleboxes along the path. Indeed, the signaling might simply be used to gather per-flow information, without

modifying network operation at all.

We call this information generically 'network control state'. Signaling messages may install, modify, refresh, or simply read this state from network elements for particular data flows. Usually a network element will also manage this information at the per-flow level, although coarser-grained state management is also possible.

3.1.4 Data Flows and Sessions

Formally, a data flow is a (unidirectional) sequence of packets between the same endpoints which follow a unique path through the network (determined by IP routing and other network layer configuration). A flow is defined by a packet classifier (in the simple cases, just the destination address and topological origin are needed). In general we assume that when discussing only the data flow path, we only need to consider 'simple' fixed classifiers (e.g. IPv4 5-tuple or equivalent).

A session is an application layer concept for a (unidirectional) flow of information between two endpoints, for which some network state is to be allocated or monitored. (Note that this use of the term 'session' is distinct from the usage in RSVP. It is closer to the session concept of, for example, the Session Initiation Protocol.)

The simplest service provided by NSIS signaling is network control state management at the flow level, as described in the previous subsection. In particular, it is possible to monitor routing updates as they change the path taken by a flow and, for example, update network state appropriately. This is no different from the case for RSVP (local path repair). Where there is a 1:1 flow:session relationship, this is all that is required.

However, for some more complex scenarios (especially mobility-related ones, see [3] and [8]) it is desirable to update the flow:session relationship during the session lifetime. For example, a new flow can be added, and the old one deleted (and maybe in that order, for a 'make-before-break' handover), effectively transferring the network control state between data flows to keep it associated with the same session. Such updates can only be managed by the end systems (because of the packet classifier change). To enable this, it must be possible for end systems to relate signaling messages to sessions as well as data flows.

3.2 Layer Model for the Protocol Suite

3.2.1 Layer Model Overview

In order to achieve a modular solution for the NSIS requirements, it is proposed to structure the NSIS protocol suite into 2 layers, similar to the original proposal in [9]:
*) a 'signaling transport' layer, responsible for moving signaling messages around, which should be independent of any particular signaling application; and
*) a 'signaling application' layer, which contains functionality such as message formats and sequences, specific to a particular signaling application.

For the purpose of this document, we use the term 'NSIS Transport Layer Protocol' (NTLP) to refer to the component that will be used in the transport layer. We also use the term 'NSIS Signaling Layer Protocol' (NSLP) to refer generically to any protocol component within the signaling application layer; in the end, there will be several NSLPs. These relationships are illustrated in Figure 4. Note that the NTLP may or may not have an interesting internal structure (e.g. based on the use of existing transport protocols) but that is not relevant at this level.

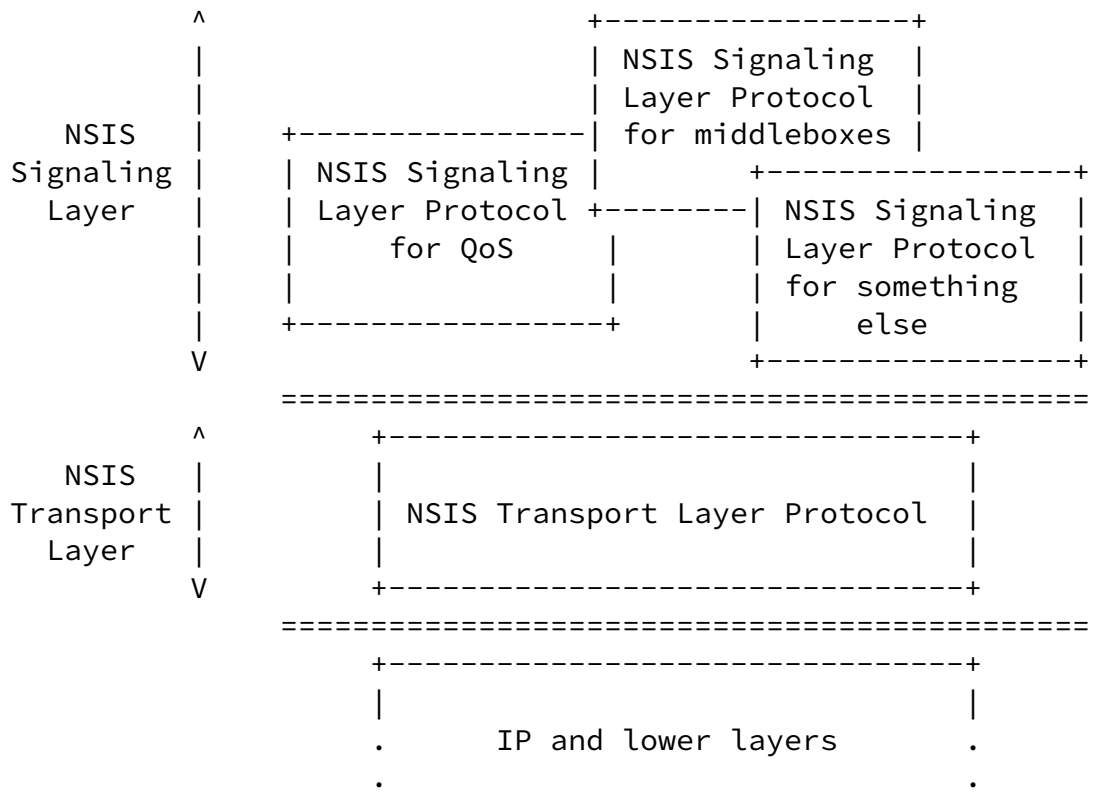


Figure 4: NSIS Protocol Components

Note that not every generic function has to be located in the NTLP. Another option would be to have re-usable components within the signaling application layer. Functionality within the NTLP should be restricted to that which interacts strongly with other transport and lower layer operations.

Because NSIS problem includes multiple signaling applications, it is very likely that a particular NSLP will only be implemented on a subset of the NSIS-aware nodes on a path, as shown in Figure 5. Messages for unrecognized NSLPs are forwarded at the NTLP level.

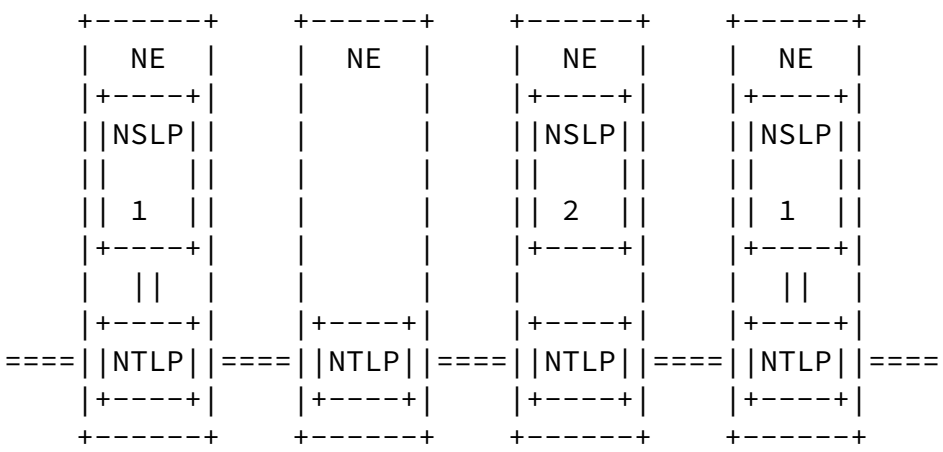


Figure 5: Signaling with Heterogeneous NSLPs

3.2.2 Layer Split Concept

This section describes the basic concepts which underlie how the necessary functionality within the NTLP can be determined. Firstly, we make a working assumption that the protocol mechanisms of the NTLP operate only between adjacent NEs (informally, the NTLP is a 'hop-by-hop' protocol), whereas any larger scope issues (including e2e aspects) are left to the upper layers.

The way in which the NTLP works can be described as follows: When a signaling message is ready to be sent from one NE, it is given to the NTLP along with information about what flow it is for; it is then up to the NTLP to get it to the next NE along the path (up- or down-stream), where it is received and the responsibility of the NTLP ends. Note that there is no assumption here about how the messages are actually addressed (this is a protocol design issue, and the options are outlined in section 4.2). The key point is that the NTLP

for a given NE does not use any knowledge about addresses, capabilities, or status of any NEs other than its direct peers.

The NTLP in the receiving NE either forwards the message directly, or, if there is an appropriate signaling application locally, passes it upwards for further processing; the signaling application can then generate another message to be sent via the NTLP. In this way, larger scope (including end-to-end) message delivery can be automatically achieved.

This definition relates to NTLP operation. It is not intended to restrict the ability of an NSLP to send messages by other means. For example, an NE in the middle or end of the signaling path could send a message directly to the other end as a notification of or acknowledgement for some signaling application event. However, it appears that the issues in sending such messages (endpoint discovery, security, NAT traversal and so on) are so different from the direct peer-peer case that there is no benefit in extending the scope of the NTLP to include such non-local functionality; instead, an NSLP which requires such messages and wants to avoid traversing the path of NEs should use some other existing transport protocol - for example, UDP would be a good match for many of the scenarios that have been proposed. Acknowledgements and notifications of this type are considered further in section 3.3.5.

One motivation for restricting the NTLP to only peer-relationship scope is that if there are any options or variants in design approach - or, worse, in basic functionality - it is easier to manage the resulting complexity if it only impacts direct peers rather than potentially the whole network.

3.2.3 Core NTLP Functionality

This section describes the basic functionality to be supported by the NTLP. Note that the analysis has to be based on considering NSLP and NTLP operation jointly; for example, we can always assume that an NSLP is operating above the NTLP and taking care of end-to-end issues (e.g. recovery of messages after restarts and so on).

Therefore, NTLP functionality is essentially just efficient upstream and downstream peer-peer message delivery in a wide variety of network scenarios. Message delivery includes the act of locating

and/or selecting which NTLP peer to carry out signaling exchanges with for a specific data flow. This discovery might be an active process (using specific signaling packets) or a passive process (a side effect of using a particular addressing mode). In addition, it appears that the NTLP can sensibly carry out most of the functions of enabling signaling messages to pass through middleboxes, since this is closely related to the problem of routing the signaling messages in the first place.

Two major open issues remain about NTLP functionality, namely what classical transport capabilities (congestion avoidance, retransmission and so on) it should have, or whether these functions can be left entirely to the upper layers; and to what extent the NTLP should provide a common state management service to the signaling applications. These questions are discussed in section 3.4.

3.2.4 Path De-Coupled Operation

Path-decoupled signaling is defined as signaling for state installation along the data path, without the restriction of passing only through nodes (NEs) that are located on the data path. Signaling messages can be routed to NEs off the data path, but which are (presumably) aware of it. This allows a looser coupling between NEs and data plane nodes, e.g. at the AS level.

The main advantages of path-decoupled signaling are ease of deployment and support of additional functionality. The ease of deployment comes from a restriction of the number of impacted nodes in case of deployment and/or upgrade of an NSLP. It would allow, for instance, deploying a solution without upgrading any of the routers in the data plane. Additional functionality that can be supported includes the use of off-path proxies to support authorization or accounting architectures.

There are potentially significant differences in the way that the two signaling paradigms should be analyzed. Using a single centralized off-path NE may increase the requirements in terms of message handling. This effect, however, is orthogonal to the NSIS charter, since path-decoupled signaling is equally applicable to distributed off-path entities. Failure recovery scenarios need to be analyzed differently because fate-sharing between data and control plane can no longer be assumed. Furthermore, the interpretation of sender/receiver orientation becomes less obvious. With the local

operation of NTLP, the impact of path-decoupled signaling on the routing of signaling messages is presumably restricted to the problem of peer determination. The assumption that the off-path NEs are loosely tied to the data path suggests, however, that peer determination can still be based on L3 routing information.

3.3 Signaling Application Properties

It is clear that many signaling applications will require specific protocol behavior in their NSLP. This section outlines some of the options for NSLP behavior; further work on selecting from these options would depend on detailed analysis of the application in question.

3.3.1 Sender/Receiver Orientation

In some signaling applications, one end of the data flow takes responsibility for requesting special treatment - such as a resource reservation - from the network. The appropriate end may depend on the signaling application, or characteristics of the network deployment.

A sender-initiated approach is when the sender of the data flow requests and maintains the resource reservation used for that flow. In a receiver-initiated approach the receiver of the data flow requests and maintains the resource reservation used for that flow. The NTLP has no freedom in this area: next peers have to be discovered in the sender to receiver direction, but after that time the default assumption is that signaling is possible both upstream and downstream (unless possibly an application specifically indicates this is not required). This implies that backward routing state must be maintained or that backward routing information must be available in the signaling packet.

The sender and receiver initiated approaches have several differences in operational characteristics. The main ones are as follows:

*) In a receiver-initiated approach, the signaling messages traveling from the receiver to the sender must be backward routed such that they follow exactly the same path as was followed by the signaling messages belonging to the same flow traveling from the sender to the receiver. This implies that a backward routing state per flow must be maintained. When using a sender-initiated approach, provided acknowledgements and notifications can be securely delivered to the

sending node, backward routing is not necessary, and nodes do not have to maintain backward routing states.
*) In a sender-initiated approach, a mobile node can initiate a reservation for its outgoing flows as soon as it has moved to another roaming subnetwork. In a receiver-initiated approach, a mobile node has to inform the receiver about its handover procedure, thus allowing the receiver to initiate a reservation for these flows. For incoming flows, the reverse argument applies.
*) A sender- (receiver-) initiated approach will allow faster setup and modification if the sender (receiver) is also authorized to carry out the operation. A mismatch between authorizing and initiating NEs will cause additional message exchanges either in the NSLP or in a protocol executed prior to NSIS invocation. Note that this may complicate modifications of network control state for existing flows.
*) Where the signaling is looking for the last (nearest to receiver) NE on the data path, receiver oriented signaling is most efficient; sender orientation would be possible, but take more messages.
*) In either case, the initiator can generally be informed faster about reservation failures than the remote end.

3.3.2 Uni- and Bi-Directional Operation

For some signaling applications and scenarios, signaling may only be considered for one direction of the data flow. However, in other cases, there may be interesting relationships between the signaling for the two directions; an example is QoS for a voice call. In the basic case, bi-directional signaling can simply use a separate instance of the same signaling mechanism in each direction. Note that the path in the two directions may differ due to asymmetric routing.

In constrained topologies where parts of the route are symmetric, it may be possible to use a more unified approach to bi-directional signaling, e.g. carrying the two signaling directions in common messages. This optimization might be used for example to make mobile QoS signaling more efficient.

In either case, the correlation of the signaling for the two flow directions is carried out in the NSLP. The NTLP would simply be enabled to bundle the messages together.

3.3.3 Heterogeneous Operation

It is likely that the appropriate way to describe the state NSIS is

signaling for will vary from one part of the network to another (depending on signaling application). For example in the QoS case, resource descriptions that are valid for inter-domain links will probably be different from those useful for intra-domain operation (and the latter will differ from one NSIS domain to another).

One way to address this issue is to consider the state description carried within the NSLP as divided in globally-understood objects ("global objects") and locally-understood objects ("local objects"). The local objects are only applicable for intra-domain signaling, while the global objects are mainly used in inter-domain signaling. Note that such local objects are still part of the NSIS protocol and can be inserted, used and removed by one single domain.

The purpose of this division is to provide additional flexibility in defining the objects carried by the NSLP such that only those objects that are applicable in a particular setting are used. An example approach for reflecting the distinction in the signaling is that local objects could be put into separate local messages that are initiated and terminated within one single NSIS domain and/or they could be "stacked" within the NSLP messages that are used for inter-domain signaling.

3.3.4 Peer-Peer and End-End Relationships

The assumption taken in this framework is that the NTLP will operate 'locally', that is, just over the scope of a single peer relationship. End-to-end operation is built up by simply concatenating these relationships. Any non-local operation (if any) will take place only in particular NSLPs.

The peering relations may also have an impact on the required amount of state at each NSIS entity. When direct interaction with remote peers is not allowed, it may be required to keep track of the path that a message has followed through the network. This can be achieved by keeping per-flow state at the NSIS entities or by maintaining a record route object in the messages.

Note that, for the reasons discussed in section 3.2.1, NSLP peers are not inevitably NTLP peers. This has a number of implications for the relationship between the signaling layers, in that NSLP peers may depend on the service provided by a concatenation of NTLP peer

relationships rather than a single one, which makes it harder to exploit fully some NTLP properties (e.g. channel security, reliability).

3.3.5 Acknowledgements and Notifications

We are assuming that the NTLP provides a simple message transfer service, and any acknowledgements or notifications it generates are handled purely internally (and apply within the scope of a single peer relationship).

However, we expect that some signaling applications will requires acknowledgements regarding the failure/success of state installation along the data path, and this will be an NSLP function.

Acknowledgements can be sent along the sequence of NTLP peer relationships towards the signaling initiator, which relieves the requirements on the security associations that need to be maintained by NEs and can ensure NAT traversal in both directions. (If this direction is towards the flow sender, it implies maintaining reverse routing state in the NTLP). In certain circumstances (e.g. trusted domains), an optimization can be made by sending acknowledgements directly to the signaling initiator (see section 3.2.2).

The semantics of the acknowledgement messages are of particular importance. An NE sending a message could assume responsibility for the entire downstream chain of NEs, indicating for instance the availability of reserved resources for the entire downstream path. Alternatively, the message could have a more local meaning,

indicating for instance that a certain failure or degradation occurred at a particular point in the network.

Notifications differ from acknowledgements because they are not (necessarily) generated in response to other signaling messages. This means that it may not be obvious to determine where the notification should be sent. Other than that, the same considerations apply as for acknowledgements. One useful distinction to make would be to differentiate between notifications that trigger a signaling action and others that don't. The security requirements for the latter are less stringent, which means they could be sent directly to the NE they are destined for (provided this NE can be determined).

3.3.6 Security and other AAA Issues

In some cases it will be possible to achieve the necessary level of signaling security by using basic 'channel security' mechanisms [10] at the level of the NTLP, and the possibilities are described in section 4.6. In other cases, signaling applications may have specific security requirements, in which case they are free to invoke their own authentication and key exchange mechanisms and apply 'object security' to specific fields within the NSLP messages.

In addition to authentication, authorisation (to manipulate network control state) has to be considered as functionality above the NTLP level, since it will be entirely application specific. Indeed, authorisation decisions may be handed off to a third party in the protocol (e.g. for QoS, the resource management function as described in section 6.1.5). Many different authorisation models are possible, and the variations impact
*) what message flows take place - for example, whether authorisation information is carried along with a control state modification request, or is sent in the reverse direction in response to it;
*) what administrative relationships are required - for example, whether authorisation takes place only between peer signaling applications, or over longer distances.

Because the NTLP operates only between adjacent peers, and places no constraints on the direction or order in which signaling applications can send messages, these authorisation aspects are left open to be defined by each NSLP. Further background discussion of this issue is contained in [11].

3.4 Open Layer Model Issues

3.4.1 Classical Transport Functionality

The first major issue is the extent to which the NTLP should include 'traditional' transport like functions, or whether these should be seen as either fundamentally unnecessary or automatically handled by the upper layers. The following functions have been identified as candidates:

1. Local retransmission to improve reliability. The argument in favor is that the NTLP can recover from congestive loss or corruption much more rapidly than end-to-end (NSLP) mechanisms; the argument against is that the additional complexity in the NTLP is not needed for all signaling applications. (It's assumed that the NTLP is not actually providing perfect message delivery guarantees or notifications, for example because NSLP peers may be separated by more than one NTLP peer relationship. A signaling application that needs peer-peer acknowledgements of this nature should define them within the NSLP.) In-order message delivery and duplicate detection are related functions.

2. Congestion control. Here, the question is whether the NTLP should include a common mechanism which protects the local portion of the network from overload, or whether this can be derived from the behavior of each signaling application.

3. Message fragmentation. For NSLPs that generate large messages, it will be necessary to fragment and re-assemble them efficiently within the network, where the use IP fragmentation may lead to reduced reliability and be incompatible with some addressing schemes. (It's assumed that the counterpart functionality, of bundling small messages together, can be provided locally by the NTLP as an option if desired; it doesn't affect the operation of the network elsewhere.)

4. Flow control. Here, the question is how a receiving NSLP should be protected from overload - whether the NTLP should provide a flow controlled channel, or whether this should be managed using application layer acknowledgements, for example.

It appears that all these issues don't affect the basic way in which the NSLP/NTLP layers relate to each other (e.g. in terms of the semantics of the inter-layer interaction); it is much more a question of the overall performance/complexity tradeoff implied by placing certain functions within each layer.

3.4.2 State Management

It is clear that the NTLP may have to manage some per-flow state to carry out its message delivery functions (for example, state about

the reverse route for signaling messages, or state needed for route change detection). How this state is stored and managed is an internal matter for the NTLP (see section 4), and the details (in particular, any connection model it might use) is in any case entirely invisible to the signaling applications.

However, signaling applications are frequently managing network control state for their own purposes, and it is an open issue how much the NTLP should provide a common service to do this for them.

The simplest case is that the NTLP simply delivers messages, and any state-related aspects (lifetimes, message semantics and so on) are entirely invisible to it, being part of the signaling application data. This provides the simplest interface between NTLP and NSLP.

The other extreme is where the NTLP provides a complete state management service, including explicit commands for creation, modification and deletion of state with known lifetimes in remote nodes. This service could make it easy to write new signaling applications, at the cost of increasing the complexity of the NTLP/NSLP interface; in particular, there would be many more events and error conditions to generate within the NTLP, and there may be several different types of state management semantics required by different signaling applications. The commonality with other parts of NTLP functionality is not clear.

An intermediate case is where there is particular support for the refresh messages used for soft-state maintenance. The characteristics of these messages are that they can be sent and processed without invoking signaling application specific logic, and that their timing can be manipulated to fit in with other NTLP requirements (e.g. jittering to avoid network synchronization, or to allow bundling with other messages). Therefore, provided this functionality can be defined simply and universally, there may be benefits in supporting it within the NTLP itself. The implication would be that some NTLP messages contain timing and other control information (to allow the refresh to be handled correctly at intermediate NSLP-unaware nodes). In addition, the automatic generation and reception of refreshes could be handled above or below the NSLP/NTLP boundary (this seems to be mainly an API issue).

4. The NSIS Transport Layer Protocol

This section describes the overall functionality required from the NTLP. It mentions possible protocol components within the NTLP layer and the different possible addressing modes that can be utilized. The interfaces between NTLP and the layers above and below it are identified, with a description of the identity elements that appear on these interfaces.

It is not the intention of this discussion to design the NTLP or even to discuss design options, although some are described as examples. The goal is to provide a general discussion of required functionality and to highlight some of the issues associated with this.

4.1 Internal Protocol Components

The NTLP includes all functionality below the signaling application layer and above the IP layer. The functionality that is required within the NTLP is described in section 3.2.

Some NTLP functionality could be provided via components existing as sublayers within the NTLP design. For example, if specific transport capabilities are required, such as congestion avoidance, retransmission, security and so on, then existing protocols, such as TCP or TLS, could be incorporated into the NTLP. This possibility is not required or excluded by this framework.

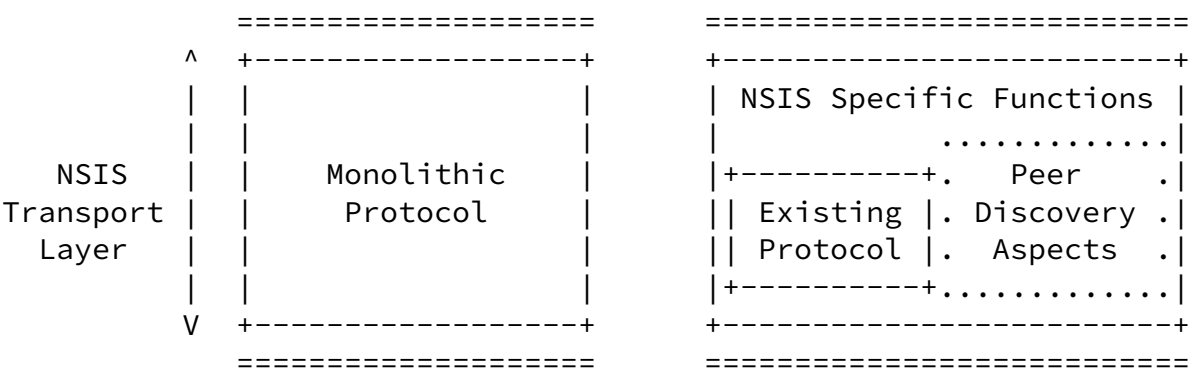


Figure 6: Options for NTLP Structure

If peer-peer addressing (section 4.2) is used for some messages, then active next-peer discovery functionality will be required within the NTLP to support the explicit addressing of these messages. (This could use message exchanges for dynamic peer discovery as a sublayer within the NTLP; there could also be an interface to external mechanisms to carry out this function.)

4.2 Addressing

There are two ways to address a signaling message being transmitted between NSIS peers:
*) peer-peer, where the message is addressed to a neighboring NSIS entity that is known to be closer to the destination NE.
*) end-to-end, where the message is addressed to the destination directly, and intercepted by an intervening NE.

With peer-peer addressing, an NE will determine that address of the next NE based on the payload of the message (and potentially on the previous NE). This requires the address of the destination NE to be derivable from the information present in the payload. This can be achieved through the availability of a local routing table or through participation in active peer discovery message exchanges. Peer-peer addressing inherently supports tunneling of messages between NEs, and is equally applicable to the path-coupled and path-decoupled cases.

In the case of end-to-end addressing, the message is addressed to the data flow receiver, and (some of) the NEs along the data path intercept the messages. The routing of the messages should follow exactly the same path as the associated data flow (but see section 5.1.1 on this point). Note that securing messages sent this way raises some interesting security issues (these are discussed in [12]).

It is not possible at this stage to mandate one addressing mode or the other. Indeed, each is necessary for some aspects of NTLP operation: in particular, initial discovery of the next downstream peer will usually require end-to-end addressing, whereas reverse routing will always require peer-peer addressing. For other message types, the choice is a matter of protocol design. The mode used is not visible to the NSLP, and the information needed in each case is available from the flow identifier (section 4.5.1) or locally stored NTLP state.

4.3 Lower Layer Interfaces

The NTLP interacts with 'lower layers' of the protocol stack for the purposes of sending and receiving signaling messages. This framework places the lower boundary of the NTLP at the IP layer. The interface to the lower layer is therefore very simple:
*) The NTLP sends raw IP packets
*) The NTLP receives raw IP packets. In the case of peer-peer

addressing, they have been addressed directly to it. In the case of end-to-end addressing, this will be achieved by intercepting packets that have been marked in some special way (by special protocol number

or by some option interpreted within the IP layer, such as the Router Alert option [13] and [14]).
*) The NTLP receives indications from the IP layer regarding route changes and similar events (see section 5.1).

For correct message routing, the NTLP needs to have some information about link and IP layer configuration of the local networking stack. For example, it needs to know:
*) [in general] how to select the outgoing interface for a signaling message, in case this needs to match the interface that will be used by the corresponding flow. This might be as simple as just allowing the IP layer to handle the message using its own routing table. There is no intention to do something different from IP routing (for end-to-end addressed messages); however, some hosts allow applications to bypass routing for their data flows, and the NTLP processing must account for this.
*) [in the case of IPv6] what address scopes are associated with the interface that messages are sent and received on (to interpret scoped addresses in flow identification, if these are to be allowed).

Configuration of lower layer operation to handle flows in particular ways is handled by the signaling application.

4.4 Upper Layer Services

The NTLP offers transport layer services to higher layer signaling applications for two purposes: sending and receiving signaling messages, and exchanging control and feedback information.

For sending and receiving messages, two basic control primitives are required:
*) Send Message, to allow the signaling application to pass data to the NTLP for transport.
*) Receive Message, to allow the NTLP to pass received data to the signaling application.

The NTLP and signaling application may also want to exchange other control information, such as:
*) Signaling application registration/de-registration, so that

particular signaling application instances can register their presence with the transport layer. This may also require some identifier to be agreed between the NTLP and signaling application to allow support the exchange of further control information and to allow the de-multiplexing of incoming data.
 *) NTLP configuration, allowing signaling applications to indicate what optional NTLP features they want to use, and to configure NTLP operation, such as controlling what transport layer state should be maintained.

 *) Error messages, to allow the NTLP to indicate error conditions to the signaling application and vice versa.
 *) Feedback information, such as route change indications so that the signaling application can decide what action to take.

The exact form of the primitives used across this interface and the information exchanged by them depends on a decision about what the responsibility of the layers is either side of the interface, and where state is managed (see section 3.4.2).

4.5 Identity Elements

4.5.1 Flow Identification

The flow identification is a method of identifying a flow in a unique way. All packets associated with the same flow will be identified by the same flow identifier. The key aspect of the flow identifier is to provide enough information such that the signaling flow receives the same treatment along the data path as that actual data itself, i.e. consistent behavior is applied to the signaling and data flows by a NAT or policy-based forwarding engine.

Information that could be used in flow identification may include:
 *) source IP address;
 *) destination IP address;
 *) protocol identifier and higher layer (port) addressing;
 *) flow label (typical for IPv6);
 *) SPI field for IPSec encapsulated data;
 *) DSCP/TOS field
It is assumed that wildcarding on these identifiers is not needed (further analysis may be required).

We've assumed here that the flow identification is not hidden within the NSLP, but is explicitly part of the NTLP. The justification for

this is that it might be valuable to be able to do NSIS processing even at a node which was unaware of the specific signaling application (see section 3.2.1): an example scenario would be messages passing through an addressing boundary where the flow identification had to be re-written.

4.5.2 Session Identification

There are circumstances where it is important to be able to refer to signaling application state independently of the underlying flow. For example, if the address of one of the flow endpoints changes due to a mobility event, it is desirable to be able to change the flow identifier without having to install a completely new reservation.

The session identifier provides a method to correlate the signaling about the different flows with the same network control state.

The session identifier is essentially a signaling application concept, since it is only used in non-trivial state management actions that are application specific. However, we assume here that it should be visible within the NTLP. This enables it to be used to control NTLP behavior, for example, by controlling how the transport layer should forward packets belonging to this session (as opposed to this signaling application). In addition, the session identifier can be used by the NTLP to demultiplex received signaling messages between multiple instances of the same signaling application, if such an operational scenario is supported (see section 4.5.3 for more information on signaling application identification).

To be useful for mobility support, the session identifier should be globally unique, and it should not be modified end-to-end. It is well known that it is practically impossible to generate identifiers in a way which guarantees this property; however, using a large random number makes it highly likely. In any case, the NTLP ascribes no valuable semantics to the identifier (such as 'session ownership'); this problem is left to the signaling application, which may be able to secure it to use for this purpose.

4.5.3 Signaling Application Identification

Since the NTLP can be used to support several NSLP types, there is a need to identify which type a particular signaling message exchange is being used for. This is to support:

*) processing of incoming messages - the NTLP should be able to demultiplex these towards the appropriate signaling applications;
*) processing of general messages at an NSIS aware intermediate node - if the node does not handle the specific signaling application, it should be able to make a forwarding decision without having to parse upper layer information.

No position is taken on the form of the signaling application identifier, or even the structure of the signaling application 'space' - free-standing applications, potentially overlapping groups of capabilities, etc. These details should not influence the rest of NTLP design.

4.6 Security Properties

It is assumed that the only security service required within NTLP is channel security. Channel security requires a security association to be established between the signaling endpoints, which is carried out

via some authentication and key management exchange. This functionality could be provided by reusing a standard protocol.

In order to protect a particular signaling exchange, the NSIS entity needs to select the security association that it has in place with the next NSIS entity that will be receiving the signaling message. The ease of doing this depends on the addressing model in use by the NTLP (see section 4.2).

Channel security can provide many different types of protection to signaling exchanges, including integrity and replay protection and encryption. It is not clear which of these is required at the NTLP layer, although most channel security mechanisms support them all.

Channel security can also be applied to the signaling messages with differing granularity, i.e. all or parts of the signaling message may be protected. For example, if the flow is traversing a NAT, only the parts of the message that do not need to be processed by the NAT should be protected. It is an open question as to which parts of the NTLP messages need protecting, and what type of protection should be applied to each.

5. Interactions with Other Protocols

5.1 IP Routing Interactions

The NSIS Transport Layer (NTLP) is responsible for discovering the next node to be visited by the signaling protocol. For path-coupled signaling, this next node should be the one that will be visited by the data flow. In practice, this peer discovery will be approximate, as any node could use any feature of the peer discovery packet to route it differently than the corresponding data flow packets. Divergence between data and signaling path can occur due to load sharing or load balancing (section 5.1.1). An example specific to the case of QoS is given in section 6.1.1. Route changes cause a temporary divergence between the data path and the path on which signaling state has been installed. The occurrence, detection and impact of route changes is described in section 5.1.2. A description of this issue in the context of QoS is given in section 6.1.2.

5.1.1 Load Sharing and Policy-Based Forwarding

Load sharing or load balancing is a network optimization technique that exploits the existence of multiple paths to the same destination in order to obtain benefits in terms of protection, resource efficiency or network stability. The significance of load sharing in the context of NSIS is that, if the load sharing mechanism in use will forward packets on any basis other than the destination address,

routing of messages using end-to-end addressing does not guarantee that the messages will follow the data path. Policy-based forwarding for data packets - where the outgoing link is selected based on policy information about fields additional to the packet destination address - has the same impact.

Signaling and data flow packets may diverge because of these techniques. In OSPF, load balancing can be used between equal cost paths [15] or unequal cost paths. An example of the latter approach is Optimized Multi Path (OMP). OMP discovers multiple paths, not necessarily equal cost paths, to any destinations in the network, but based on the load reported from a particular path, it determines which fraction of the data to direct to the given path. Incoming packets are subject to a (source, destination address) hash computation, and effective load sharing is accomplished by means of adjusting the hash thresholds. BGP [16][17] advertises the routes chosen by the BGP decision process to other BGP speakers. In the basic specification, routes with the same Network Layer reachability information (NLRI) as previously advertised routes implicitly replace

the original advertisement, which means that multiple paths for the same prefix cannot exist. Recently, however, a new mechanism was defined that will allow the advertisement of multiple paths for the same prefix without the new paths implicitly replacing any previous ones [18]. The essence of the mechanism is that each path is identified by an arbitrary identifier in addition to its prefix.

If the routing decision is based on both source and destination address, signaling and data flow packets may still diverge because of layer 4 load-balancing (based on TCP/UDP or port-based). Such techniques would, however, constrain the use of proxies. Proxies would cause ICMP errors to be misdirected to the source of the data because of source address spoofing.

If the routing decision is based on the complete five-tuple, divergence may still occur because of the presence of router alert options. In this case, the same constraint on proxy use applies as above. Additionally, it becomes difficult for the end systems to distinguish between data and signaling packets. Finally, QoS routing techniques (section 6.1.3) may base the routing decision on any field in the packet header (e.g. DSCP, ...).

Most load-balancing techniques use the first n bytes of the packet header (including SA, DA and protocol field) in the hashing function. In this case, the above considerations regarding source/destination address or five-tuple based forwarding apply.

5.1.2 Route Changes

In a routed network, each packet is independently routed based on its header information. Whenever a better route towards the destination becomes available, this route is installed in the forwarding table and will be used for all subsequent (data and signaling) packets. This can cause a divergence between the path along which state has been installed and the path along which forwarding will actually take place.

The possibility of route changes requires the presence of three processes in the signaling protocol
1. route change detection
2. installation of state on the new path

3. teardown of state on the old path

Many route change detections methods can be used, some of which need explicit protocol support and some of which are implementation-internal. They differ in their speed of reaction and the types of change they can detect. In rough order of increasing applicability, they can be summarized as:
a) monitoring changes in local interface state
b) monitoring network-wide topology changes in a link-state routing protocol
c) inference from changes in data packet TTL
d) inference from loss of packet stream in a downstream flow-aware router
e) inference from changes in signalling packet TTL
f) changed route of a PATH-like (end-to-end addressed) signaling packet
g) changed route of a specific end-to-end addressed probe packet

There are essentially three ways in which detection can happen: based on network monitoring (method a-b), based on data packet monitoring (method c-d) and based on monitoring signaling protocol messages (method e-g). Methods contingent on monitoring signaling messages become less effective as refresh reduction techniques are used.

When a route change has been detected, it is important that state is installed as quickly as possible along the new path. It is not guaranteed that the new path will be able to provide the same characteristics that were available on the old path. In order to be able to avoid duplicate state installation or, worse, rejection of the signaling message because of previously installed state, it is important to be able to recognize the new signaling message as belonging to an existing session. In this respect, we distinguish between route changes with associated change of the flow specification (e.g. in case of a mobility event when the IP source

might change) and route changes without change of the flow specification (e.g. in case of a link failure along the path). The former case requires an identifier independent from the flow specification.

When state has been installed along the new path, the existing state on the old path needs to be removed. With the soft-state principle, this will happen automatically because of the lack of refresh messages. Depending on the refresh timer, however, it may be required

to tear down this state much faster (e.g. because it is tied to an accounting record). In that case, the teardown message needs to be able to distinguish between the new path and the old path.

The problem of route changes would not occur if there was a way to do route pinning. Route pinning refers to the independence of the path taken by certain data packets from reachability changes caused by routing updates from an Interior Gateway Protocol (OSPF, IS-IS) or an Exterior Gateway Protocol (BGP).

5.1.3 Router Redundancy

In some environments, it is desired to provide connectivity and per flow or per class flow management with high-availability characteristics, i.e. with rapid transparent recovery even in the presence of route changes. This may involve interactions with the basic protocols which are used to manage the routing in this case, such as VRRP [19]. A future version of this document may consider interactions between NSIS and such protocols in support of high availability functionality.

5.2 Mobility Interactions

Mobility, in most cases, causes changes to the data path that packets take. Assuming that signaling has taken place prior to any mobility to establish some state along the data path, new signaling may be needed in order to (re)establish state along the changed data path.

The interactions between mobility and signaling protocols have been extensively analyzed in recent years, primarily in the context of RSVP and Mobile IP interaction (e.g. [20]), but also in the context of other types of network (e.g. [21]). This analysis work has shown that some difficulties in the interactions are quite deeply rooted in the detailed design of these protocols; however, the problems and their possible solutions fall under five broad headings. The main issues for a resource signaling application are limiting the period after handovers during for which the resource states are not available along the path; and avoiding double reservations -

reservations on both the old path and new path. Similar issues may apply to other types of signaling application.

5.2.1 Addressing and Encapsulation

A mobility solution typically involves address reallocation on handover (unless a network supports per host routing) and may involve special packet formats (e.g. the routing header and Home Address option of MIPv6). Since NSIS may depend on end system addresses for forwarding signaling messages and defining flows (section 4.5.1), the special implications of mobility for addressing need to be considered. Examples of possible approaches that could be used to solve the addressing and encapsulation problem are as follows:

* Use a flow identification based on low level IP addresses (e.g. the Care of Address) and other 'standard' fields in the IP header. This makes least demands on the packet classification engines within the network. However, it means that even on a part of the flow path that is unchanged, the session will need to be modified to reflect the changed flow identification (see section 5.2.3).

* Use a flow identification that does not change (e.g. based on Home Address); this is the approach assumed in [22]. This simplifies the problem of session update, at the likely cost of considerably complicating the flow identification requirements.

In the first approach, to prevent double reservation, NSIS entities need to be able to recognize that a session with the new flow identifier is to be correlated with an existing one. A session identifier could be used for this purpose. This is why the session identifier as described in section 4.5.2 has to have end-to-end semantics.

While the feasibility and performance of this first approach needs to be assessed, given the high impact of requiring more sophisticated packet classifiers, it still seems more plausible than the second approach. This implies that signaling applications should define flows in terms of real, routable (care of) addresses rather than virtual (home) addresses.

5.2.2 Localized Path Repair

In any mobility approach, a handover will cause at least some changes in the path of upstream and downstream packets. At some point along the joined path, an NSIS entity should be able to recognize this situation, based upon session identification. New state needs to be installed on the new path, and removed from the old. Who triggers the new state may be constrained by which entities are allowed to carry

out which state manipulations, which is then a signaling application question.

A critical point here is the signaling that is used to discover the crossover node. This is a generalization of the problem of finding next-NSIS peer: it requires extending the new path over several hops until it intersects the old one. This is easy for the uplink direction (where the mobile is the sender), but much harder for downlink without signaling via the correspondent. There is no reason for the crossover node for uplink and downlink flows to be the same, even for the same correspondent. The problem is discussed further in [23].

5.2.3 Update on the Unchanged Path

On the path between the crossover node(s) and the correspondent, it is necessary to avoid, if possible, double reservations, but rather to update the network control state to reflect new flow identification (this is needed, by the default assumption of section 5.2.1). Examples of approaches that could be used to solve this problem are the following:

*) Use a session state identification that does not change even if the flow definition changes (see Section 4.5.2). Signaling is still needed to update a changed flow identification, but it should be possible to avoid AAA and admission control processing.

*) Use an NSIS-capable crossover router that manages this update autonomously (more efficiently than the end nodes could), with similar considerations to the local path repair case.

Note that in the case of an address change, end to end message exchanges will be required at the application layer anyway, so signaling to update the flow identifier does not necessarily add to the handover latency.

5.2.4 Interaction with Mobility Signaling

In existing work on mobility protocol and signaling protocol interactions, several framework proposals describing the protocol interactions have been made. Usually they have taken existing protocols (Mobile IP and RSVP respectively) as the starting point; it should be noted that an NSIS protocol might operate in quite a different way. In this section, we provide an overview of how these proposals would be reflected in framework of NSIS. The mobility aspects are described using Mobile IP terminology, but are generally applicable to other network layer mobility solutions. The purpose of this overview is not to select or prioritise any particular approach,

but simply to point out how they would fit into our framework and any major issues with them.

We can consider that two signaling processes are active: mobility signaling and NSIS signaling. The discussion so far considered how NSIS signaling should operate. There is still a question of how the interactions between the NSIS and mobility signaling should be considered.

The basic case of totally independent specification and implementation seems likely to lead to ambiguities and even interoperability problems (see [22]). At least, the addressing and encapsulation issues for mobility solutions that use virtual links or their equivalents need to be specified in an implementation-neutral way.

A type of 'loose' integration is to have independent protocol definitions, but to define how they trigger each other - in particular, how the mobility protocol triggers sending of refresh/modify/tear messages. A pair of implementations could use these triggers to improve performance, primarily reducing latency. (Existing RSVP modifications consider the closer interaction of making the RSVP implementation mobility routing aware, e.g. so it is able to localize refresh signaling; this would be a self contained aspect of NSIS.) This information could be developed by analyzing message flows for various mobility signaling scenarios as was done in [20].

An even tighter level of integration is to consider a single protocol carrying both mobility and network control state information. Logically, there are two cases:

1. Carry mobility routing information (a 'mobility object') in the signaling messages, as is done in [22]. (The prime purpose in this approach is to enable crossover router discovery.)

2. Carry signaling in the mobility messages, typically as a new extension header. This was proposed in [24] and followed up in [25]; [26] also anticipates this approach. In our framework, we could consider this a special case of NSIS layering, with the mobility protocol playing the role of the signaling transport.

Other modes of interaction might also be possible. The critical point with all these models is that the general solutions developed by NSIS

should be independent of mobility protocols. Tight integration would have major deployment issues especially in interdomain cases. Therefore, any tightly integrated solution is considered out of scope of initial NSIS development.

5.2.5 Interaction with Context Transfer

In the context of mobility between different access routers, it is common to consider performance optimizations in two areas: selection of the optimal access router to handover to, and transfer of state information between the access routers to avoid having to regenerate it in the new access router after handover. The Seamoby Working Group is developing solutions for these protocols (CARD [27] and Context Transfer [28] respectively); alternative approaches with similar characteristics are also possible.

As these solutions are still underdevelopment, it is premature to specify details on the interaction. It is thought that Context Transfer transfers state between access routers based upon changes caused by mobility. NSIS protocol state may be a candidate for context transfer. Such work, should it be undertaken, will be done in the Seamoby working group.

5.3 Interactions with NATs

Because at least some messages will almost inevitably contain address and possibly higher layer information as payload, we must consider the interaction with address translation devices (NATs). As well as 'traditional' NATs of various types (as defined in [29]) very similar considerations would apply to some IPv4/v6 transition mechanisms such as SIIT [30].

In the simplest case of an NSIS unaware NAT in the signaling path, payloads will be uncorrected and the signaling will be for the incorrect flow. Applications could attempt to use STUN [31] or similar techniques to detect and recover from the presence of the NAT. Even then, NSIS protocols would have to use a well known encapsulation (TCP/UDP/ICMP) to avoid being dropped by the more cautious low-end NAT devices.

A simple 'NSIS-aware' NAT would require flow identification information to be in the clear and not integrity protected. An alternative conceptual approach is to consider the NAT functionality

being part of message processing itself, in which case the translating node can take part natively in any NSIS protocol security mechanisms. Depending on NSIS protocol layering, it would be possible for this processing to be done in an NSIS entity which was otherwise ignorant of any particular signaling applications. This is the motivation for including basic flow identification information in the NTLP (section 4.5.1).

Note that all of this discussion is independent of the use of a specific NSLP for general control of NATs (and firewalls). This is considered in section 6.2.

6. Signaling Applications

This section describes NSLPs for particular signaling applications. The assumption is that the NSLP uses the generic functionality of the NTLP given earlier; this section describes specific aspects of NSLP operation.

6.1 Signaling for Quality of Service

In the case of signaling for QoS, all the basic NSIS concepts of section 3.1 apply. In addition, there is an assumed directionality of the signaling process, in that one end of the signaling flow takes responsibility for actually requesting the resource. This leads to the following definitions:
*) NSIS Initiator (NI): the signaling entity which makes the resource request, usually as a result of user application request.
*) NSIS Responder (NR): the signaling entity that acts as the endpoint for the signaling and can optionally interact with applications as well.
*) NSIS Forwarder (NF): the signaling entity an NI and NR which propagates NSIS signaling further through the network.
Each of these entities will interact with a resource management function (RMF) which actually allocates network resources (router buffers, interface bandwidth and so on).

Note that there is no constraint on which end of the signaling flow should take the NSIS Initiator role: with respect to the data flow direction it could be at the sending or receiving end.

6.1.1 Protocol Messages

The QoS NSLP will include a set of messages to carry out resource reservations along the signaling path. A message set for the QoS NSLP is shown below (a very similar set of messages was generated in [32]). Note that the 'direction' column in the table below only indicates the 'orientation' of the message. The messages can be originated and absorbed at NF nodes as well as the NI or NR; an example might be NFs at the edge of a domain exchanging messages to set up resources for a flow across a it.

Note the working assumption that responder as well as the initiator can release a reservation (comparable to rejecting it in the first place). It is left open if the responder can modify a reservation, during or after setup. This seems mainly a matter of assumptions

about authorization, and the possibilities might depend on resource type specifics.

Name	Direction	Semantics
Request	I-->R	Create a new reservation for a flow
Modify	I-->R (&R-->I?)	Modify an existing reservation
Release	I-->R & R-->I	Delete (tear down) an existing reservation
Accept/ Reject	R-->I	Confirm (possibly modified?) or reject a reservation request
Notify	I-->R & R-->I	Report an event detected within the network
Refresh	I-->R	State management (see section 4.4)

The table also explicitly includes a refresh message. This does nothing to a reservation except extend its lifetime, and is one possible state management mechanism (see next section).

6.1.2 State Management

The prime purpose of NSIS is to manage state information along the path taken by a data flow. The issues regarding state management within the NTLP (state related to message transport) are described in section 4. The QoS NSLP will typically have to handle additional state related to the desired resource reservation to be made.

There two critical issues to be considered in building a robust NSLP to handle this problem:
*) The protocol must be scalable. It should allow minimization of the resource reservation state storage demands that it implies for intermediate nodes; in particular, storage of state per 'micro' flow is likely to be impossible except at the very edge of the network. A QoS signaling application might require per flow or lower granularity state; examples of each for the case of QoS would be IntServ [33] or RMD [34] (per 'class' state) respectively.
*) The protocol must be robust against failure and other conditions, which imply that the stored resource reservation state has to be moved or removed.

For resource reservations, typically soft state management is considered for robustness reasons. It is currently open whether the soft state protocol aspects should be built into the NSLP for specific signaling applications, or provided as a generic service by the NTLP; this issue is discussed in section 3.4.2.

6.1.3 QoS Forwarding

The assumption is that the NTLP works with standard (i.e. best-effort) layer 3 routing. There are, however, several proposals for the introduction of QoS awareness in the routing protocols. All of these essentially lead to the existence of multiple paths (with different QoS) towards the same destination. As such, they also contain an inherent risk for a divergence between control plane and data plane, similar to the load sharing case. Clearly, any QoS NSLP needs to be able to handle this type of routing.

For intra-domain data flows, the difference in routing may result from a QoS-aware traffic engineering scheme, that e.g. maps incoming flows to LSPs based on multi-field classification. In BGP, several techniques for including QoS information in the routing decision are currently proposed. A first proposal is based on a newly defined BGP-4 attribute, the QoS_NLRI attribute [16]. The QoS_NLRI attribute is an optional transitive attribute that can be used to advertise a QoS

route to a peer or to provide QoS information along with the Network Layer Reachability Information (NLRI) in a single BGP update. A second proposal is based on controlled redistribution of AS routes [17]. It defines a new extended community (the redistribution extended community) that allows a router to influence how a specific route should be redistributed towards a specified set of eBGP speakers. The types of redistribution communities may result in a specific route not being announced to a specified set of eBGP speakers, that it should not be exported or that the route should be prepended n times.

6.1.4 Route Changes and QoS Reservations

In this section, we will explore the expected interworking between a signaling for resource BGP routing updates, although the same applies for any source of routing updates. The normal operation of the NSIS protocol will lead to the situation depicted in Figure 7, where the reserved resources match the data path.

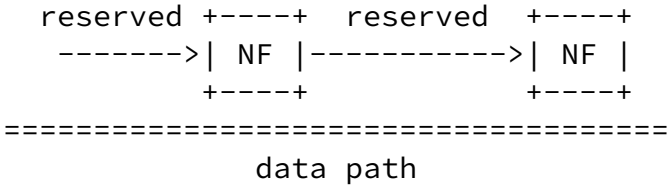
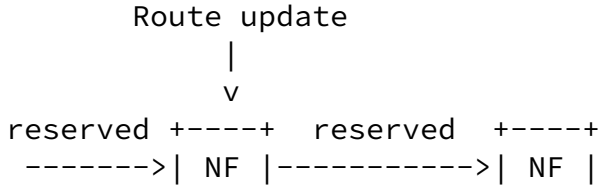


Figure 7: Normal NSIS protocol operation

A route change (triggered by a BGP routing update for instance) can occur while such a reservation is in place. In case of RSVP, the route change will be installed immediately and any data that is sent will be forwarded on the new path. This situation is depicted Figure 8.



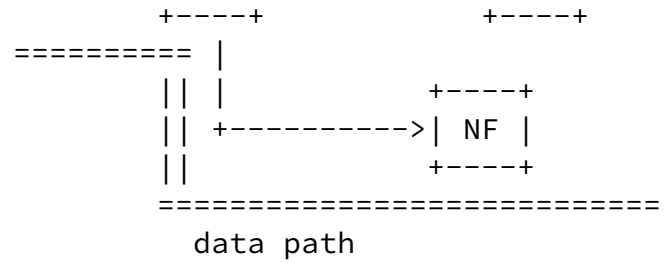


Figure 8: Route Change

Resource reservation on the new path will only be started once the next control message is routed along the new path. This means that there is a certain time interval during which resources are not reserved on (part of) the data path. To minimize this time interval several techniques could be considered. As an example, RSVP [4] has the concept of local repair, where the router may be triggered by a route change. In that case the RSVP node can start sending PATH messages directly after the route has been changed. Note that this option may not be available if no per-flow state is kept in the NF.

It is not guaranteed that the new path will be able to provide the same guarantees that were available on the old path. Therefore, in a more desirable scenario, the NF should wait until resources have been reserved on the new path before installing the route change (unless of course the old path no longer exists). The route change procedure then consists of the following steps:
1. NF receives a route announcement,
2. Refresh messages are forwarded along the current path,

3. A copy of the refresh message is re-marked as a request and send along the new path that was announced,
4. When the NF has been acknowledged about the reservations on the new path the route will be installed and the data will flow along the new path.

Another example related to route changes is denoted as severe congestion and is explained in [34]. This solution adapts to a route change, when a route change creates a congestion on the new routed path.

6.1.5 Resource Management Interactions

The QoS NSLP itself is not involved in any specific resource

allocation or management techniques. The definition of an NSLP for resource reservation with Quality-of-Service, however, implies the notion of admission control. For a QoS NSLP, the measure of signaling success will be the ability to reserve resources from the total resource pool that is provisioned in the network. We define the function responsible for allocating this resource pool as the Resource Management Function (RMF). The RMF is responsible for all resource provisioning, monitoring and assurance functions in the network.

A QoS NSLP will rely on the RMF to do resource management and to provide inputs for admission control. In this model, the RMF acts as a server towards client NSLP(s). It is noted, however, that the RMF may in turn use another NSLP instance to do the actual resource provisioning in the network. In this case, the RMF acts as the initiator (client) of an NSLP.

This essentially corresponds to a multi-level signaling paradigm, with an 'upper' level handling internetworking QoS signaling, possibly running end-to-end, and a 'lower' level handling the more specialised intradomain QoS signaling, running between just the edges of the network (see [35], [36], and [37] for a discussion of similar architectures). Given that NSIS signaling is already supposed to be able to support multiple instances of NSLPs for a given flow, and limited scope (e.g. edge-to-edge) operation, it is not currently clear that supporting the multi-level model leads to any new protocol requirements for the QoS NSLP.

The RMF may or may not be co-located with an NF (note that co-location with an NI/NR can be handled logically as a combination between NF and NI/NR). To cater for both cases, we define a (possibly logical) NF-RMF interface. Over this interface, information may be provided from the RMF about monitoring, resource availability, topology, and configuration. In the other direction, the interface

may be used to trigger requests for resource provisioning. One way to formalize the interface between the NF and the RMF is via a Service Level Agreement (SLA). The SLA may be static or it may be dynamically updated by means of a negotiation protocol. Such a protocol is outside the scope of NSIS.

There is no assumed restriction on the placement of the RMF. It may be a centralized RMF per domain, several off-path distributed RMFs, or an on-path RMF per router. The advantages and disadvantages of

both approaches are well-known. Centralization typically allows decisions to be taken on more global information with more efficient resource utilization as a result. It also facilitates deployment or upgrade of policies. Distribution allows local decision processes and rapid response to data path changes.

6.2 Other Signaling Applications

As well as the use for 'traditional' QoS signaling, it should be possible to use develop NSLPs for other signaling applications which operate on different types of network control state. One specific case is setting up flow-related state in middleboxes (firewalls, NATs, and so on). Requirements for such communication are given in [6], and initial discussions of NSIS-like solutions are contained in [38], [39] and [40]. Other examples include network monitoring and testing, and tunnel endpoint discovery.

A future version of this document may contain more details on how to build NSLPs for these types of signaling application.

7. Security Considerations

This document describes a framework for signaling protocols which assumes a two-layer decomposition, with a common lower layer (NTLP) supporting a family of signaling application specific upper layer protocols (NSLPs). The overall security considerations for the signaling therefore depend on the joint security properties assumed or demanded for each layer.

Security for the NTLP is discussed in section 4.6. We have assumed that the role of the NTLP will be to provide message protection over the scope of a single peer relationship (between adjacent signaling entities), and that this can most likely be provided by some kind of channel security mechanism using an external key management mechanism based on mutual authentication. In addition, the NTLP should be resilient against denial of service attacks on the protocol itself.

Security for the NSLPs is entirely dependent on signaling application requirements. In some cases, no additional protection may be required

compared to what is provided by the NTLP. In other cases, more sophisticated object-level protection and the use of public key based solutions may be required. In addition, the NSLP needs to consider the authorisation requirements of the signaling application.

Another factor is that NTLP security mechanisms operate only locally, whereas NSLP mechanisms may also need to operate over larger regions (not just between adjacent peers) especially for authorisation aspects; this complicates the analysis of basing signaling application security on NTLP protection. Further work on this and other security design will depend on a refinement of the NSIS threats work begun in [12].

8. Change History

8.1 Changes from draft-ietf-nsis-fw-01.txt

This -02 version has been very significantly restructured compared to the previous version, and a section by section change history is probably neither possible or useful. Instead, this section lists the major technical and structural changes.

1. The concept of splitting the protocol suite into two layers is now introduced much earlier, and the rest of the framework restructured around it. In general, the content is supposed to be signaling application independent: possibilities for application dependent behavior are described in section 3.3, and the specific case of QoS/resource management is restricted to section 6.1.
2. Sender and receiver orientation is now assumed to be a signaling application protocol property (section 3.3.1), with the NTLP by default operating bidirectionally (section 3.2.3). As a consequence, the initiator, forwarder, and responder concepts only appear in the later sections.
3. In general, the NTLP is now a 'thinner' layer than previously envisaged (e.g. without specific reserve/tear messages), and so the possible inter-layer coupling with the NSLP is much reduced. However, the option of the NTLP providing some kind of generic state management service is still an open issue (section 3.4.2).
4. In general, authorisation issues are still handled by the NSLP, including the question of which network entities are allowed to modify network state. In particular, the issue of 'session' (previously 'reservation') ownership (section 3.1.4) is assumed to be handled by the NSLP level, although session identification is still visible to the NTLP (section 4.5.2). The implication is that most key aspects of mobility support (section 5.2) are now NSLP responsibilities.

5. Both peer-peer and end-to-end addressing modes are assumed to be needed for the NTLP, and any choice between them is a protocol design issue (not visible externally).

References

1 Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

2 Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997

3 Brunner, M., "Requirements for QoS Signaling Protocols", draft-ietf-nsis-req-05.txt (work in progress), November 2002

4 Braden, R., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997

5 Chaskar, H. (editor), "Requirements of a QoS Solution for Mobile IP", draft-ietf-mobileip-qos-requirements-03.txt (work in progress), July 2002

6 Swale, R. P., P. A. Mart, P. Sijben, S. Brim, M. Shore, "Middlebox Communications (midcom) Protocol Requirements", RFC 3304, August 2002

7 Manner, J. and X. Fu, "Analysis of Existing Quality of Service Signaling Protocols", draft-ietf-nsis-signalling-analysis-01.txt (work in progress), February 2003

8 Thomas, M., "Analysis of Mobile IP and RSVP Interactions", draft-thomas-nsis-rsvp-analysis-00.txt (work in progress), October 2002

9 Braden, R., and B. Lindell, "A Two-Level Architecture for Internet Signaling", draft-braden-2level-signaling-01.txt (work in progress), November 2002

10 Rescorla, E. et al., "Guidelines for Writing RFC Text on Security Considerations", draft-iab-sec-cons-03.txt (work in progress), January 2003

11 Tschofenig, H., M. Buechli, S. Van den Bosch, H. Schulzrinne, "NSIS Authentication, Authorization and Accounting Issues", draft-tschofenig-nsis-aaa-issues-00.txt (work in progress), February 2003

12 Tschofenig, H. and D. Kroeselberg, "Security Threats for NSIS", draft-ietf-nsis-threats-01.txt (work in progress), January 2003

13 Katz, D., "IP Router Alert Option", RFC 2113, February 1997

14 Partridge, C., A. Jackson, "IPv6 Router Alert Option", RFC 2711, October 1999

15 Apostolopoulos, G., D. Williams, S. Kamat, R. Guerin, A. Orda, T. Przygienda, "QoS Routing Mechanisms and OSPF Extensions", RFC 2676, August 1999

16 Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995

17 Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", draft-ietf-idr-bgp4-17.txt (work in progress), January 2002 (expired)

18 Walton, D., D. Cook, A. Retana and J. Scudder, "Advertisement of Multiple Paths in BGP", draft-walton-bgp-add-paths-01.txt (work in progress), November 2002

19 Knight, S., D. Weaver, D. Whipple, R. Hinden, D. Mitzel, P. Hunt, P. Higginson, M. Shand, A. Lindem, "Virtual Router Redundancy Protocol", RFC2338, April 1998

20 Thomas, M., "Analysis of Mobile IP and RSVP Interactions", draft-thomas-nsis-rsvp-analysis-00.txt (work in progress), October 2002

21 Partain, D., G. Karagiannis, P. Wallentin, L. Westberg, "Resource Reservation Issues in Cellular Radio Access Networks", draft-westberg-rmd-cellular-issues-01.txt (work in progress), June 2002

22 Shen, C. et al., "An Interoperation Framework for Using RSVP in Mobile IPv6 Networks", draft-shen-rsvp-mobileipv6-interop-00.txt (work in progress), July 2001 (expired)

23 Manner, J., et al., "Localized RSVP", draft-manner-lrsvp-01.txt (work in progress), January 2003

24 Chaskar, H. and R. Koodli, "A Framework for QoS Support in Mobile

IPv6", draft-chaskar-mobileip-qos-01.txt (work in progress), March 2001 (expired)

25 Fu, X., et al, "QoS-Conditionalized Binding Update in Mobile IPv6", draft-tkn-nsis-qosbinding-mipv6-00.txt (work in progress), January 2002 (expired)

26 Kan, Z., "Two-plane and Three-tier QoS Framework for Mobile IPv6 Networks", draft-kan-qos-framework-01.txt (work in progress), July 2002

27 Trossen, D., G. Krishnamurthi, H. Chaskar, J. Kempf, "Issues in candidate access router discovery for seamless IP-level handoffs", draft-ietf-seamoby-cardiscovery-issues-04.txt (work in progress), October 2002

28 Kempf, J., "Problem Description: Reasons For Performing Context Transfers Between Nodes in an IP Access Network", RFC3374, September 2002

29 Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC2663, August 1999

30 Nordmark, E., "Stateless IP/ICMP Translation Algorithm (SIIT)", RFC2765, February 2000

31 Rosenberg, J., J. Weinberger, C. Huitema, R. Mahy, "STUN - Simple Traversal of UDP Through Network Address Translators", draft-ietf-midcom-stun-05.txt (work in progress), December 2002

32 Westberg, L., G. Karagiannis, D. Partain, V. Rexhepi., "Framework for Edge-to-Edge NSIS Signaling", draft-westberg-nsis-edge-edge-framework-00.txt (work in progress), May 2002

33 Braden, R., D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, June 1994

34 Westberg, L., Csaszar, A., Karagiannis, G., Marquetant, A., Partain, D., Pop, O., Rexhepi, V., Szabó, R., Takács, A., "Resource Management in Diffserv (RMD): A Functionality and Performance Behavior Overview", Seventh International Workshop on

Protocols for High-Speed networks - PfHSN 2002, 22 - 24 April 2002

35 Ferrari, D., A. Banerjea, H. Zhang, "Network Support for Multimedia - A Discussion of the Tenet Approach", Berkeley TR-92-072, November 1992

36 Nichols, K., V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", RFC 2638, July 1999

37 Baker, F., C. Iturralde, F. Le Faucheur, B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations", RFC 3175, September 2001

38 Shore, M., "Towards a Network-friendlier Midcom", draft-shore-friendly-midcom-01.txt (work in progress), June 2002

39 Shore, M., "The TIST (Topology-Insensitive Service Traversal) Protocol", draft-shore-tist-prot-00.txt (work in progress), May 2002

40 Brunner, M. and M. Stiemerling, "Middlebox Signaling in a NSIS Framework", draft-brunner-nsis-mbox-fmwk-00.txt (work in progress), June 2002

Acknowledgments

The authors would like to thank Anders Bergsten, Bob Braden, Maarten Buchli, Eleanor Hepworth, Melinda Shore and Hannes Tschofenig for significant contributions in particular areas of this draft. In addition, the authors would like to acknowledge Cedric Aoun, Marcus Brunner, Danny Goderis, Cornelia Kappler, Mac McTiffin, Hans De Neve, David Partain, Vlora Rexhepi, Henning Schulzrinne and Lars Westberg for insights and inputs during this and previous framework activities.

Authors' Addresses

Ilya Freytsis
Cetacean Networks Inc.
100 Arboretum Drive
Portsmouth, NH 03801

USA
email: ifreytsis@cetacean.com

Robert Hancock
Roke Manor Research
Old Salisbury Lane
Romsey
Hampshire
SO51 0ZN
United Kingdom
email: robert.hancock@roke.co.uk

Georgios Karagiannis
Ericsson EuroLab Netherlands B.V.
Institutenweg 25
P.O.Box 645
7500 AP Enschede
The Netherlands
email: georgios.karagiannis@eln.ericsson.se

John Loughney
Nokia Research Center
11-13 Italahdenkatu
00180 Helsinki
Finland
email: john.loughney@nokia.com

Sven Van den Bosch
Alcatel
Francis Wellesplein 1
B-2018 Antwerpen
Belgium
email: sven.van_den_bosch@alcatel.be

Intellectual Property Considerations

might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

OR FITNESS FOR A PARTICULAR PURPOSE.