Network Working Group
Request for Comments: 4080
Category: Informational

R. Hancock
Siemens/RMR
G. Karagiannis
University of Twente/Ericsson
J. Loughney
Nokia
S. Van den Bosch
Alcatel

June 2005

Next Steps in Signaling (NSIS): Framework

Status of This Memo

Copyright Notice

Abstract

The Next Steps in Signaling (NSIS) working group is considering protocols for signaling information about a data flow along its path in the network. The NSIS suite of protocols is envisioned to support various signaling applications that need to install and/or manipulate such state in the network. Based on existing work on signaling requirements, this document proposes an architectural framework for these signaling protocols.

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

Table of Contents

1. Introduction 3
   1.1. Definition of the Signaling Problem 3
   1.2. Scope and Structure of the NSIS Framework 3
2. Terminology 4
3. Overview of Signaling Scenarios and Protocol Structure 6
   3.1. Fundamental Signaling Concepts 6
        3.1.1. Simple Network and Signaling Topology 6

3.1.2. Path-Coupled and Path-Decoupled Signaling 7
3.1.3. Signaling to Hosts, Networks, and Proxies 8
3.1.4. Signaling Messages and Network Control State 10
3.1.5. Data Flows and Sessions 10
3.2. Layer Model for the Protocol Suite 11
3.2.1. Layer Model Overview 11
3.2.2. Layer Split Concept 12
3.2.3. Bypassing Intermediate Nodes 13
3.2.4. Core NSIS Transport Layer Functionality 15
3.2.5. State Management Functionality 16
3.2.6. Path-Decoupled Operation 17
3.3. Signaling Application Properties 18
3.3.1. Sender/Receiver Orientation 18
3.3.2. Uni- and Bi-Directional Operation 19
3.3.3. Heterogeneous Operation 19
3.3.4. Aggregation 20
3.3.5. Peer-Peer and End-End Relationships 21
3.3.6. Acknowledgements and Notifications 21
3.3.7. Security and Other AAA Issues 22
4. The NSIS Transport Layer Protocol 23
4.1. Internal Protocol Components 23
4.2. Addressing 24
4.3. Classical Transport Functions 24
4.4. Lower Layer Interfaces 26
4.5. Upper Layer Services 27
4.6. Identity Elements 28
4.6.1. Flow Identification 28
4.6.2. Session Identification 28
4.6.3. Signaling Application Identification 29
4.7. Security Properties 30
5. Interactions with Other Protocols 30
5.1. IP Routing Interactions 30
5.1.1. Load Sharing and Policy-Based Forwarding 31
5.1.2. Route Changes 31
5.2. Mobility and Multihoming Interactions 33
5.3. Interactions with NATs 36
5.4. Interactions with IP Tunneling 36
6. Signaling Applications 37
6.1. Signaling for Quality of Service 37
6.1.1. Protocol Message Semantics 38
6.1.2. State Management 39
6.1.3. Route Changes and QoS Reservations 39
6.1.4. Resource Management Interactions 41
6.2. Other Signaling Applications 42
7. Security Considerations 42
8. References 43
8.1. Normative References 43
8.2. Informative References 44

1. Introduction

1.1. Definition of the Signaling Problem

The Next Steps in Signaling (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, independently 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 (i.e., the 'path-coupled' case; see Section 3.1.2) and that only unicast data flows are considered.

The signaling problem in this sense is very similar to that addressed by RSVP. However, there are two generalizations. First, the intention is that components of the NSIS protocol suite will be usable 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).

Second, 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 work is to define both the generic protocol and, initially, upper layers suitable for QoS signaling (similar to the corresponding functionality in RSVP) and middlebox signaling. Further 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 [1] and a separate security threats document [2]; other related requirements can be found in [3] and [4] for QoS/Mobility and middlebox communication, respectively. This framework does not replace or update these requirements. Discussions about lessons to be learned from existing signaling and resource management protocols are contained in separate analysis documents [5], [6].

The role of this framework is to explain how NSIS signaling should work within the broader networking context, and to describe the overall structure of the protocol suite itself. Therefore, it

discusses important protocol considerations such as routing, mobility, security, and interactions with network 'resource' management (in the broadest sense).

The basic context for NSIS protocols is given in Section 3. Section 3.1 describes the fundamental elements of NSIS protocol operation in comparison to RSVP [7]; in particular, Section 3.1.3 describes more general signaling scenarios, and Section 3.1.4 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 detailed design of the protocol or even design options, although some are described as examples. It describes the interfaces between this lower-layer protocol and the IP layer (below) and signaling application protocols (above), including the identifier elements that appear on these interfaces (Section 4.6). 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 (NAT).

Section 6 describes particular signaling applications. The example of signaling for QoS (comparable to core RSVP QoS signaling functionality) is given 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; we do not intend 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

Classifier: an entity that selects packets based on their contents according to defined rules.

[Data] flow: a stream of packets from sender to receiver that is a distinguishable subset of a packet stream. Each flow is distinguished by some flow identifier (see Section 4.6.1).

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

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

NSIS Entity (NE): the function within a node that 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 in which the signaling messages follow a path that is tied to the data messages.

Path-decoupled 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 that is receiving the data packets in a flow.

Sender: the node in the network that 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 3.1.5).

Signaling application: the purpose of the NSIS signaling. A signaling application 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 that 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 here we make no assumptions about the particular sequence of signaling messages that will be invoked.

```
       Sender                                                Receiver
   +-----------+      +----+      +----+      +----+      +-----------+
   |Application|----->| R1 |----->| R2 |----->| R3 |----->|Application|
   |   +--+    |      |+--+|      |+--+|      +----+      |   +--+    |
   |   |NE|====|======||NE||======||NE||=================|===|NE|    |
   |   +--+    |      |+--+|      |+--+|                  |   +--+    |
   +-----------+      +----+      +----+                  +-----------+

      +--+
      |NE| = NSIS      ==== = Signaling    ---> = Data flow messages
      +--+   Entity           Messages            (unidirectional)
```

Figure 1: Simple Signaling and Data Flows

3.1.2. Path-Coupled and Path-Decoupled Signaling

We can consider two basic paradigms for resource reservation signaling, which we refer to as "path-coupled" and "path-decoupled".

In the path-coupled case, signaling messages are routed only through NEs that are on the data path. They do not have to reach all the nodes on the data path. (For example, there could be intermediate signaling-unaware nodes, or the presence of proxies such as those shown in Figure 2 could prevent the signaling from reaching the path end points.) Between adjacent NEs, the route taken by signaling and data might diverge. The path-coupled case can be supported by various addressing styles, with messages either explicitly addressed to the neighbor on-path NE, or addressed identically to the data packets, but also with the router alert option (see [8] and [9]), and intercepted. These cases are considered in Section 4.2. In the second case, some network configurations may split the signaling and data paths (see Section 5.1.1); this is considered an error case for path-coupled signaling.

In the path-decoupled case, signaling messages are routed to nodes (NEs) that are not assumed to be on the data path, but that are (presumably) aware of it. Signaling messages will always be directly addressed to the neighbor NE, and the signaling endpoints may have no relation at all with the ultimate data sender or receiver. The implications of path-decoupled operation for the NSIS protocols are considered briefly in Section 3.2.6; however, the initial goal of NSIS and this framework is to concentrate mainly on the path-coupled case.

3.1.3. Signaling to Hosts, Networks, and Proxies

There are different possible triggers for the signaling protocols. Among them are user applications (that are using NSIS signaling services), other signaling applications, 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.

The NSIS protocol suite 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.

Although the 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 a single domain or multiple concatenated NSIS domains. The latter case assumes the existence of trust relations 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 some other entities in 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 a kind of "proxy on the data path"; see Figure 2.

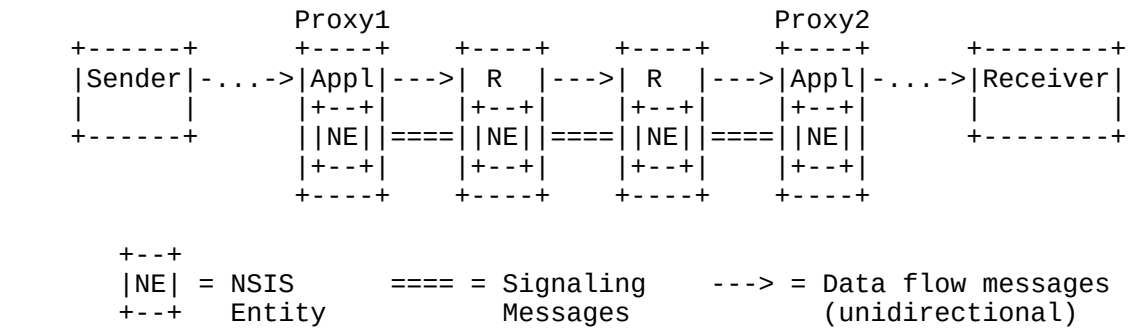

Figure 2: "On path" NSIS proxy

This configuration presents two specific challenges for the signaling:

- A proxy that terminates signaling on behalf of the NSIS-unaware host (or part of the network) should be able to determine that it is the 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 a QoS signaling application).

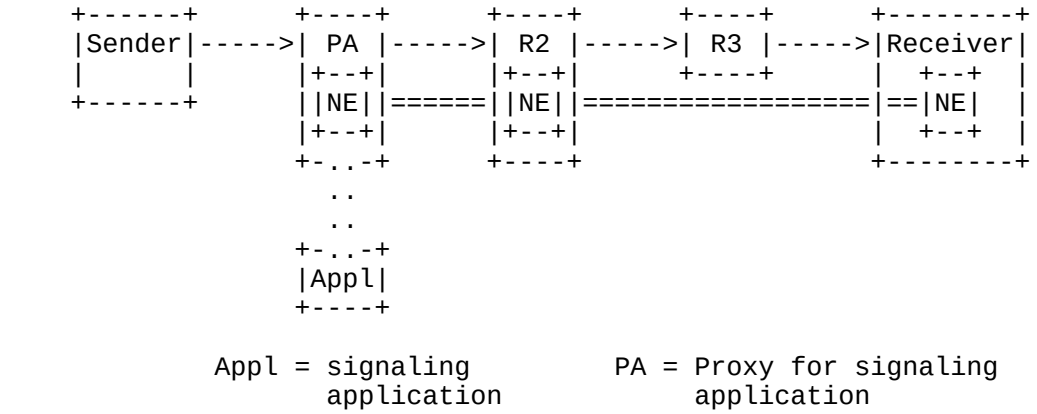

Figure 3: "Off path" NSIS proxy

Another possible configuration, shown in Figure 3, is where an NE can send and receive signaling information to a remote processor. The NSIS protocols may or may not be suitable for this remote interaction, but in any case it is not currently part of the NSIS problem. This configuration is supported by considering the NE a proxy at the signaling application level. This is a natural implementation approach for some policy control and centralized control architectures; see also Section 6.1.4.

3.1.4. 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 that are of interest 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 'network control state' generically. 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 ('per-class') state management is also possible.

3.1.5. Data Flows and Sessions

Formally, a data flow is a (unidirectional) sequence of packets between the same endpoints that all follow a unique path through the network (determined by IP routing and other network configuration). A flow is defined by a packet classifier (in the simplest 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 an exchange of packets between two endpoints, for which some network state is to be allocated or monitored. In simple cases, a session may map to a specific flow; however, signaling applications are allowed to create

more flexible flow:session relationships. (Note that this concept of 'session' is different from that of RSVP, which defines a session as a flow with a specific destination address and transport protocol. The NSIS usage is closer to the session concepts of higher-layer protocols.)

The simplest service provided by NSIS signaling protocols is the management of network control state at the level of a specific flow, as described in the previous subsection. In particular, it should be 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 and multihoming related ones; see [1] and the mobility discussion of [5]), it is desirable to update the flow:session mapping 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 are best managed by the end systems (generally, systems that understand the flow:session mapping and are aware of the packet classifier change). To enable this, it must be possible to relate signaling messages to sessions as well as to data flows. A session identifier (Section 4.6.2) is one component of the solution.

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, the NSIS protocol suite will be structured in two layers:

- 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 within the signaling application layer; in the end, there will be several NSLPs, largely independent of each other. These relationships are

illustrated in Figure 4. Note that the NTLP may or may not have an interesting internal structure (e.g., including existing transport protocols), but that is not relevant at this level of description.

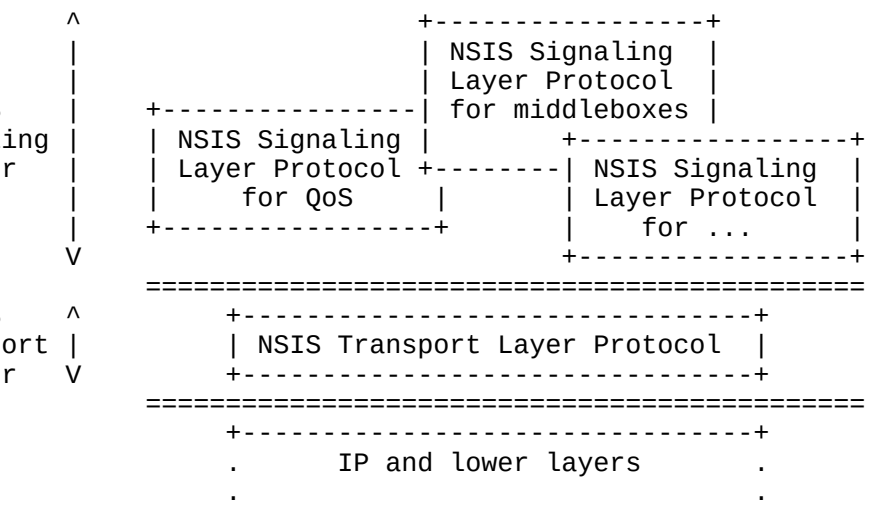

Figure 4: NSIS Protocol Components

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 what interacts strongly with other transport and lower-layer operations.

3.2.2. Layer Split Concept

This section describes the basic concepts underlying the functionality of the NTLP. First, 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 (upstream or downstream), 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 is achieved.

This definition relates to NTLP operation. It does not 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 or acknowledgement of some signaling application event. However, 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 NTLP to include such non-local functionality. Instead, an NSLP that requires such messages and wants to avoid traversing the path of NEs should use some other existing transport protocol. For example, UDP or DCCP 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 3.3.6.

One motivation for restricting the NTLP to 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 Internet.

3.2.3. Bypassing Intermediate Nodes

Because the 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. In addition, a node inside an aggregation region will still wish to ignore signaling messages that are per-flow, even if they are for a signaling application that the node is generally able to process.

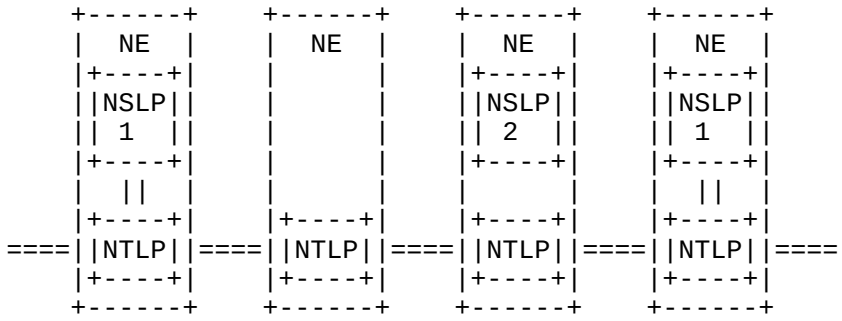

Figure 5: Signaling with Heterogeneous NSLPs

Where signaling messages traverse such NSIS-aware intermediate nodes, it is desirable to process them at the lowest level possible (in particular, on the fastest path). In order to offer a non-trivial message transfer service (in terms of security, reliability and so on) to the peer NSLP nodes, it is important that the NTLP at intermediate nodes is as transparent as possible; that is, it carries out minimal processing. In addition, if intermediate nodes have to do slow-path processing of all NSIS messages, this eliminates many of the scaling benefits of aggregation, unless tunneling is used.

Considering first the case of messages sent with the router alert option, there are two complementary methods to achieve this bypassing of intermediate NEs:

- At the IP layer, a set of protocol numbers or a range of values in the router alert option can be used. In this way, messages can be marked with an implied granularity, and routers can choose to apply further slow-path processing only to configured subsets of messages. This is the method used in [10] to distinguish per-flow and per-aggregate signaling.

- The NTLP could process the message but determine that there was no local signaling application it was relevant to. At this stage, the message can be returned unchanged to the IP layer for normal forwarding; the intermediate NE has effectively chosen to be transparent to the message in question.

In both cases, the existence of the intermediate NE is totally hidden from the NSLP nodes. If later stages of the signaling use directly addressed messages (e.g., for reverse routing), they will not involve the intermediate NE at all, except perhaps as a normal router.

There may be cases where the intermediate NE would like to do some restricted protocol processing, such as the following:

- Translating addresses in message payloads (compare Section 4.6.1); note that this would have to be done to messages passing in both directions through a node.

- Updating signaling application payloads with local status information (e.g., path property measurement inside a domain).

If this can be done without fully terminating the NSIS protocols, it would allow a more lightweight implementation of the intermediate NE, and a more direct 'end-to-end' NTLP association between the peer NSLPs where the signaling application is fully processed. On the other hand, this is only possible with a limited class of possible NTLP designs, and makes it harder for the NTLP to offer a security service (since messages have to be partially protected). The feasibility of this approach will be evaluated during the NTLP design.

3.2.4. Core NSIS Transport Layer Functionality

This section describes the basic functionality to be supported by the NTLP. Note that the overall signaling solution will always be the result of joint operation of both the NTLP and the signaling layer protocols (NSLPs); 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).

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 many of the functions that enable signaling messages to pass through middleboxes, since this is closely related to the problem of routing the signaling messages in the first place. Further details about NTLP functionality are contained in Sections 3.2.5 and 4.3.

3.2.5. State Management Functionality

Internet signaling requires the existence and management of state within the network for several reasons. This section describes how state management functionality is split across the NSIS layers. (Note that how the NTLP internal state is managed is a matter for its design and indeed implementation.)

1. Conceptually, the NTLP provides a uniform message delivery service. It is unaware of the difference in state semantics between different types of signaling application messages (e.g., whether a message changes, just refreshes signaling application state, or even has nothing to with signaling application state at all).

2. An NTLP instance processes and, if necessary, forwards all signaling application messages "immediately". (It might offer different service classes, but these would be distinguished by, for example, reliability or priority, not by state aspects.) This means that the NTLP does not know explicit timer or message sequence information for the signaling application; and that signaling application messages pass immediately through an NSLP-unaware node. (Their timing cannot be jittered there, nor can messages be stored up to be re-sent on a new path in case of a later re-routing event.)

3. Within any node, it is an implementation decision whether to generate/jitter/filter refreshes separately within each signaling application that needs this functionality, or to integrate it with the NTLP implementation as a generic "soft-state management toolbox". The choice doesn’t affect the NTLP specification at all. Implementations might piggyback NTLP soft-state refresh information (if the NTLP works this way) on signaling application messages, or they might even combine soft-state management between layers. The state machines of the NTLP and NSLPs remain logically independent, but an implementation is free to allow them to interact to reduce the load on the network to the same level that would be achieved by a monolithic model.

4. It may be helpful for signaling applications to receive state-management related ’triggers’ from the NTLP indicating that a peer has failed or become available ("down/up notifications"). These triggers would be about adjacent NTLP peers, rather than signaling application peers. We can consider this another case of route change detection/notification (which the NTLP is also allowed to do anyway). However, apart from generating such

triggers, the NTLP takes no action itself on such events, other than to ensure that subsequent signaling messages are routed correctly.

5. The existence of these triggers doesn't replace NSLP refreshes as the mechanism for maintaining liveness at the signaling application level. In this sense, up/down notifications are advisories that allow faster reaction to events in the network, but that shouldn't be built into NSLP semantics. (This is essentially the same distinction, with the same rationale, that SNMP makes between notifications and normal message exchanges.)

3.2.6. Path-Decoupled Operation

Path-decoupled signaling is defined as signaling for state installation along the data path, without the restriction of passing only through nodes that are located on the data path. Signaling messages can be routed to nodes that are off the data path, but that are (presumably) aware of it. This allows a looser coupling between signaling and data plane nodes (e.g., at the autonomous system level). Although support for path-decoupled operation is not one of the initial goals of the NSIS work, this section is included for completeness and to capture some initial considerations for future reference.

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. Path-decoupled signaling 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; on the other hand, 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 planes can no longer be assumed. Furthermore, the interpretation of sender/receiver orientation becomes less natural. With the local operation of the 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 NSIS nodes are loosely tied to the data path suggests, however, that peer determination can still be based on L3 routing information. This

means that a path-decoupled signaling solution could be implemented using a lower-layer protocol presenting the same service interface to NSLPs as the path-coupled NTLP. A new message transport protocol (possibly derived from the path-coupled NTLP) would be needed, but NSLP specifications and the inter-layer interaction would be unchanged from the path-coupled case.

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 signaling application in question.

3.3.1. Sender/Receiver Orientation

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

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

The sender- and receiver-initiated approaches have several differences in their 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. In a sender-initiated approach, provided that acknowledgements and notifications can be delivered securely to the sending node, backward routing is not necessary, and nodes do not have to maintain backward routing state.

- 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, thus allowing the receiver to initiate a reservation for these flows. For incoming flows, the reverse argument applies.

- In general, setup and modification will be fastest if the node responsible for authorizing these actions can initiate them directly within the NSLP. 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. Depending on how the authorization for a particular signaling application is done, this may favor either sender- or receiver-initiated signaling. Note that this may complicate modification of network control state for existing flows.

3.3.2. Uni- and Bi-Directional Operation

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

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 for which NSIS is signaling will vary from one part of the network to another (depending on the 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 domain to another).

One way to address this issue is to consider the state description used within the NSLP as carried in globally-understood objects and locally-understood objects. The local objects are only applicable for intra-domain signaling, while the global objects are mainly used in inter-domain signaling. Note that the local objects are still part of the protocol but are 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 the objects applicable in a particular setting are used. One approach for reflecting the distinction is that local objects could be put into separate local messages that are initiated and terminated within one single domain; an alternative is that they could be "stacked" within the NSLP messages that are used anyway for inter-domain signaling.

3.3.4. Aggregation

It is a well-known problem that per-flow signaling in large-scale networks presents scaling challenges because of the large number of flows that may traverse individual nodes.

The possibilities for aggregation at the level of the NTLP are quite limited; the primary scaling approach for path-coupled signaling is for a signaling application to group flows together and to perform signaling for the aggregate, rather than for the flows individually. The aggregate may be created in a number of ways; for example, the individual flows may be sent down a tunnel, or given a common Differentiated Services Code Point (DSCP) marking. The aggregation and de-aggregation points perform per flow signaling, but nodes within the aggregation region should only be forced to process signaling messages for the aggregate. This depends on the ability of the interior nodes to ignore the per-flow signaling as discussed in Section 3.2.3.

Individual NSLPs will need to specify what aggregation means in their context, and how it should be performed. For example, in the QoS context it is possible to add together the resources specified in a number of separate reservations. In the case of other applications, such as signaling to NATs and firewalls, the feasibility (and even the meaning) of aggregation is less clear.

3.3.5. Peer-Peer and End-End Relationships

The assumption 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 concatenating these relationships. Non-local operation (if any) will take place in 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 could be achieved by keeping per-flow state at the NSIS entities, as is done in RSVP. Another approach would be to maintain a record route object in the messages; this object would be carried within the NSIS protocols, rather than depend on the route-recording functionality provided by the IP layer.

3.3.6. Acknowledgements and Notifications

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

However, we expect that some signaling applications will require 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 that can allow NAT traversal in both directions. (If this direction is towards the sender, it implies maintaining reverse routing state in the NTLP.) In certain circumstances (e.g., trusted domains), an optimization could be to send acknowledgements directly to the signaling initiator outside the NTLP (see Section 3.2.2), although any such approach would have to take into account the necessity of handling denial of service attacks launched from outside the network.

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 how 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 that this NE can be determined).

3.3.7. 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 [11] at the level of the NTLP, and the possibilities are described in Section 4.7. 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 to apply 'object security' to specific fields within the NSLP messages.

In addition to authentication, the authorization (to manipulate network control state) has to be considered as functionality above the NTLP level, since it will be entirely application specific. Indeed, authorization decisions may be handed off to a third party in the protocol (e.g., for QoS, the resource management function as described in Section 6.1.4). Many different authorization models are possible, and the variations impact:

- what message flows take place -- for example, whether authorization 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 authorization 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 authorization aspects are left open to be defined by each NSLP. Further background discussion of this issue is contained in [12].

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, as well as the assumed transport and state management functionality. 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.

This discussion is not intended to design the NTLP or even to enumerate design options, although some are included 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 outlined in Section 3.2.4, with some more details in Sections 3.2.5 and 4.3.

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

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.

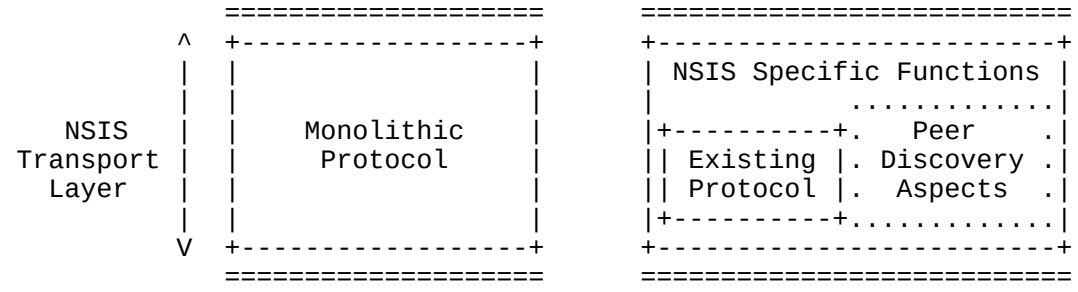

Figure 6: Options for NTLP Structure

4.2. Addressing

There are two ways to address a signaling message being transmitted between NTLP 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 flow destination directly and intercepted by an intervening NE.

With peer-peer addressing, an NE will determine the address of the next NE based on the payload of the message (and potentially on the previous NE). This requires that the address of the destination NE be derivable from the information present in the payload, either by using some 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 [2]). In addition, it is a matter of the protocol design what should be used as the source address of the message (the flow source or signaling source).

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 4.6.1) or locally stored NTLP state.

4.3. Classical Transport Functions

The NSIS signaling protocols are responsible for transporting (signaling) data around the network; in general, this requires functionality such as congestion management, reliability, and so on. This section discusses how much of this functionality should be provided within the NTLP. It appears that this doesn'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.

Note that, per the discussion at the end of Section 3.2.3, there may be cases where intermediate nodes wish to modify messages in transit even though they do not perform full signaling application processing. In this case, not all the following functionality would be invoked at every intermediate node.

The following functionality is assumed to lie within the NTLP:

1. Bundling together of small messages (comparable to [13]) can be provided locally by the NTLP as an option, if desired; it doesn't affect the operation of the network elsewhere. The NTLP should always support unbundling, to avoid the cost of negotiating the feature as an option. (The related function of refresh summarization -- where objects in a refresh message are replaced with a reference to a previous message identifier -- is left to NSLPs, which can then do this in a way tuned to the state management requirements of the signaling application. Additional transparent compression functionality could be added to the NTLP design later as a local option.) Note that end-to-end addressed messages for different flows cannot be bundled safely unless the next node on the outgoing interface is known to be NSIS-aware.

2. When needed, message fragmentation should be provided by the NTLP. The use of IP fragmentation for large messages may lead to reduced reliability and may be incompatible with some addressing schemes. Therefore, this functionality should be provided within the NTLP as a service for NSLPs that generate large messages. How the NTLP determines and accommodates Maximum Transmission Unit (MTU) constraints is left as a matter of protocol design. To avoid imposing the cost of reassembly on intermediate nodes, the fragmentation scheme used should allow for the independent forwarding of individual fragments towards a node hosting an interested NSLP.

3. There can be significant benefits for signaling applications if state-changing messages are delivered reliably (as introduced in [13] for RSVP; see also the more general analysis of [14]). This does not change any assumption about the use of soft-state by NSLPs to manage signaling application state, and it leaves the responsibility for detecting and recovering from application layer error conditions in the NSLP. However, it means that such functionality does not need to be tuned to handle fast recovery from message loss due to congestion or corruption in the lower layers, and it also means that the NTLP can prevent the

amplification of message loss rates caused by fragmentation. Reliable delivery functionality is invoked by the NSLP on a message-by-message basis and is always optional to use.

4. The NTLP should not allow signaling messages to cause congestion in the network (i.e., at the IP layer). Congestion could be caused by retransmission of lost signaling packets or by upper layer actions (e.g., a flood of signaling updates to recover from a route change). In some cases, it may be possible to engineer the network to ensure that signaling cannot overload it; in others, the NTLP would have to detect congestion and to adapt the rate at which it allows signaling messages to be transmitted. Principles of congestion control in Internet protocols are given in [15]. The NTLP may or may not be able to detect overload in the control plane itself (e.g., an NSLP-aware node several NTLP-hops away that cannot keep up with the incoming message rate) and indicate this as a flow-control condition to local signaling applications. However, for both the congestion and overload cases, it is up to the signaling applications themselves to adapt their behavior accordingly.

4.4. 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).
- The NTLP receives indications from the IP layer (including local forwarding tables and routing protocol state) that provide some information about 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. In general, it needs to know how to select the outgoing interface for a signaling message and where this must 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. Further network layer information would be needed to handle scoped addresses (if such things ever exist).

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

4.5. 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 the following:

- 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 upon between the NTLP and signaling application to 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.

4.6. Identity Elements

4.6.1. 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 the 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; and
- DSCP/TOS field.

It is assumed that at most limited wildcarding on these identifiers is needed.

We assume here that the flow identification is not hidden within the NSLP, but is explicitly part of the NTLP. The justification for this is that being able to do NSIS processing, even at a node which was unaware of the specific signaling application (see Section 3.2.3) might be valuable. An example scenario would be messages passing through an addressing boundary where the flow identification had to be re-written.

4.6.2. Session Identification

There are circumstances in which being able to refer to signaling application state independently of the underlying flow is important. 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 4.6.3 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 that 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 be used for this purpose.

4.6.3. Signaling Application Identification

Because 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; and
- 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 the NTLP design.

4.7. Security Properties

It is assumed that the only security service required within the 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. It is also not clear how tightly an NSLP can 'bind' to the channel security service provided by the NTLP.

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. (Alternatively, if the NAT takes part in NTLP security procedures, it only needs to be given access to the message fields containing addresses, often just the flow id.) Which parts of the NTLP messages need protecting is an open question, as is what type of protection should be applied to each.

5. Interactions with Other Protocols

5.1. IP Routing Interactions

The NTLP is responsible for determining the next node to be visited by the signaling protocol. For path-coupled signaling, this next node should be 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 from the corresponding data flow packets. Divergence between the data and signaling paths 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. It has been proposed for a number of routing protocols, such as OSPF [16] and others. In general, load sharing means that packet forwarding will take into account header fields in addition to the destination address; a general discussion of such techniques and the problems they cause is provided in [17].

The significance of load sharing in the context of NSIS is that routing of signaling messages using end-to-end addressing does not guarantee that these 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 packets may diverge because of both of these techniques.

If signaling packets are given source and destination addresses identical to data packets, signaling and data may still diverge because of layer-4 load balancing (based on protocol or port). Such techniques would also cause ICMP errors to be misdirected to the source of the data because of source address spoofing. If signaling packets are made identical in the complete 5-tuple, divergence may still occur because of the presence of router alert options. The same ICMP misdirection applies, and it becomes difficult for the end systems to distinguish between data and signaling packets. Finally, QoS routing techniques may base the routing decision on any field in the packet header (e.g., DSCP).

5.1.2. Route Changes

In a connectionless 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 problem of route changes is reduced if route pinning is performed. 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). Nothing about NSIS signaling prevents route pinning from being used as a network engineering technique, provided that it is done in a way

that preserves the common routing of signaling and data. However, even if route pinning is used, it cannot be depended on to prevent all route changes (for example, in the case of link failures).

Handling 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. removal of state on the old path

Many route change detection methods can be used, some needing explicit protocol support, and some of which are implementation-internal. They differ in their speed of reaction and in the types of change they can detect. In rough order of increasing applicability, they can be summarized as follows:

1. monitoring changes in local forwarding table state

2. monitoring topology changes in a link-state routing protocol

3. inference from changes in data packet TTL

4. inference from loss of packet stream in a flow-aware router

5. inference from changes in signaling packet TTL

6. changed route of an end-to-end addressed signaling packet

7. changed route of a specific end-to-end addressed probe packet

These methods can be categorized as being based on network monitoring (methods 1-2), on data packet monitoring (methods 3-4) and on monitoring signaling protocol messages (methods 5-7); method 6 is the baseline method of RSVP. The network monitoring methods can only detect local changes; in particular, method 1 can only detect an event that changes the immediate next downstream hop, and method 2 can only detect changes within the scope of the link-state protocol. Methods 5-7, which are contingent on monitoring signaling messages, become less effective as soft-state refresh rates are reduced.

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. 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 identification (e.g., in case of a mobility event when the IP source might change) and route changes without change of the flow identification (e.g., in case of a link failure along the path). The former case requires an identifier independent from the flow identification; i.e., the session identifier (Section 4.6.2). Mobility issues are discussed in more detail in Section 5.2.

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.

In some environments, it is desirable to provide connectivity and per-flow or per-class state management with high-availability characteristics; i.e., with rapid transparent recovery, even in the presence of route changes. This may require interactions with protocols that are used to manage the routing in this case, such as Virtual Router Redundancy Protocol (VRRP) [18].

Our basic assumption about such interactions is that the NTLP would be responsible for detecting the route change and ensuring that signaling messages were re-routed consistently (in the same way as the data traffic). However, further state re-synchronization (including failover between 'main' and 'standby' nodes in the high availability case) would be the responsibility of the signaling application and its NSLP, and would possibly be triggered by the NTLP.

5.2. Mobility and Multihoming Interactions

The issues associated with mobility and multihoming are a generalization of the basic route change case of the previous section. As well as the fact that packets for a given session are no longer traveling over a single topological path, the following extra considerations arise:

1. The use of IP-layer mobility and multihoming means that more than one IP source or destination address will be associated with a single session. The same applies if application-layer solutions (e.g., SIP-based approaches) are used.

2. Mobile IP and associated protocols use some special encapsulations for some segments of the data path.

3. The double route may persist for some time in the network (e.g., in the case of a 'make-before-break' handover being done by a multihomed host).

4. Conversely, the re-routing may be rapid and routine (unlike network-internal route changes), increasing the importance of rapid state release on old paths.

The interactions between mobility and signaling have been extensively analyzed in recent years, primarily in the context of RSVP and Mobile IP interaction (e.g., the mobility discussion of [5]), but also in that of other types of network (e.g., [19]). A general review of the fundamental interactions is given in [20], which provides further details on many of the subjects considered in this section.

We assume that the signaling will refer to 'outer' IP headers when defining the flows it is controlling. There are two main reasons for this. The first is that the data plane will usually be unable to work in terms of anything else when implementing per-flow treatment (e.g., we cannot expect that a router will analyze inner headers to decide how to schedule packets). The second reason is that we are implicitly relying on the security provided by the network infrastructure to ensure that the correct packets are given the special treatment being signaled for, and this is built on the relationship between packet source and destination addresses and network topology. (This is essentially the same approach that is used as the basis of route optimization security in Mobile IPv6 [21].) The consequence of this assumption is that we see the packet streams to (or from) different addresses as different flows. Where a flow is carried inside a tunnel, it is seen as a different flow again. The encapsulation issues (point (2) above) are therefore to be handled the same way as other tunneling cases (Section 5.4).

Therefore, the most critical aspect is that multiple flows are being used, and the signaling for them needs to be correlated. This is the intended role of the session identifier (see Section 4.6.2, which also describes some of the security requirements for such an identifier). Although the session identifier is visible at the NTLP, the signaling application is responsible for performing the correlation (and for doing so securely). The NTLP responsibility is limited to delivering the signaling messages for each flow between the correct signaling application peers. The locations at which the correlation takes place are the end system and the signaling-

application-aware node in the network where the flows meet. (This node is generally referred to as the "crossover router"; it can be anywhere in the network.)

Although much work has been done in the past on finding the crossover router directly from information held in particular mobility signaling protocols, the initial focus of NSIS work should be a solution that is not tightly bound to any single mobility approach. In other words, it should be possible to determine the crossover router based on NSIS signaling. (This doesn't rule out the possibility that some implementations may be able to do this discovery faster; e.g., by being tightly integrated with local mobility management protocols. This is directly comparable to spotting route changes in fixed networks by being routing aware.)

Note that the crossover router discovery may involve end-to-end signaling exchanges (especially for flows towards the mobile or multihomed node), which raises a latency concern. On the other hand, end-to-end signaling will have been necessary in any case, at the application level not only to communicate changed addresses, but also to update packet classifiers along the path. It is a matter for further analysis to decide how these exchanges could be combined or carried out in parallel.

On the shared part of the path, signaling is needed at least to update the packet classifiers to include the new flow, although if correlation with the existing flow is possible it should be possible to bypass any policy or admission control processing. State installation on the new path (and possibly release on the old one) are also required. Which entity (one of the end hosts or the crossover router) controls all these procedures depends on which entities are authorized to carry out network state manipulations, so this is therefore a matter of signaling application and NSLP design. The approach may depend on the sender/receiver orientation of the original signaling (see Section 3.3.1). In addition, in the mobility case, the old path may no longer be directly accessible to the mobile node; inter-access-router communication may be required to release state in these circumstances.

The frequency of handovers in some network types makes fast handover support protocols desirable, for selecting the optimal access router for handover (for example, [22]), and for transferring state information to avoid having to regenerate it in the new access router after handover (for example, [23]). Both of these procedures could have strong interactions with signaling protocols. The access router selection might depend on the network control state that could be

supported on the path through the new access router. Transfer of signaling application state or NTLP/NSLP protocol state may be a candidate for context transfer.

5.3. Interactions with NATs

Because at least some messages will almost inevitably contain addresses and possibly higher-layer information as payload, we must consider the interaction with address translation devices (NATs). These considerations apply both to 'traditional' NATs of various types (as defined in [24]) as well as some IPv4/v6 transition mechanisms, such as Stateless IP/ICMP Translation (SIIT) [25].

In the simplest case of an NSIS-unaware NAT in the path, payloads will be uncorrected, and signaling will refer to the flow incorrectly. Applications could attempt to use STUN [26] 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 more cautious low-end NAT devices.

A simple 'NSIS-aware' NAT would require flow identification information to be in the clear and not to be integrity protected. An alternative conceptual approach is to consider the NAT functionality 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 that was otherwise ignorant of any particular signaling applications. This is the motivation for including basic flow identification information in the NTLP (Section 4.6.1).

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

5.4. Interactions with IP Tunneling

Tunneling is used in the Internet for a number of reasons, such as flow aggregation, IPv4/6 transition mechanisms, mobile IP, virtual private networking, and so on. An NSIS solution must continue to work in the presence of these techniques. The presence of the tunnel should not cause problems for end-to-end signaling, and it should also be possible to use NSIS signaling to control the treatment of the packets carrying the tunneled data.

It is assumed that the NSIS approach will be similar to that of [27], where the signaling for the end-to-end data flow is tunneled along with that data flow and is invisible to nodes along the path of the tunnel (other than the endpoints). This provides backwards compatibility with networks where the tunnel endpoints do not support the NSIS protocols. We assume that NEs will not unwrap tunnel encapsulations to find and process tunneled signaling messages.

To signal for the packets carrying the tunneled data, the tunnel is considered a new data flow in its own right, and NSIS signaling is applied to it recursively. This requires signaling support in at least one tunnel endpoint. In some cases (where the signaling initiator is at the opposite end of the data flow from the tunnel initiator; i.e., in the case of receiver initiated signaling), the ability to provide a binding between the original flow identification and that for the tunneled flow is needed. It is left open here whether this should be an NTLP or an NSLP function.

6. Signaling Applications

This section gives an overview of 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. It includes simple examples that are intended to clarify how NSLPs fit into the framework. It does not replace or even form part of the formal NSLP protocol specifications; in particular, initial designs are being developed for NSLPs for resource reservation [28] and middlebox communication [29].

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:

- QoS NSIS Initiator (QNI): the signaling entity that makes the resource request, usually as a result of user application request.
- QoS NSIS Responder (QNR): the signaling entity that acts as the endpoint for the signaling and that can optionally interact with applications as well.
- QoS NSIS Forwarder (QNF): a signaling entity between a QNI and QNR that propagates NSIS signaling further through the network.

Each of these entities will interact with a resource management function (RMF) that 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 QNI role: With respect to the data flow direction, it could be at the sending or receiving end.

6.1.1. Protocol Message Semantics

The QoS NSLP will include a set of messages to carry out resource reservations along the signaling path. A possible set of message semantics for the QoS NSLP is shown below. Note that the 'direction' column in the table below only indicates the 'orientation' of the message. Messages can be originated and absorbed at QNF nodes as well as the QNI or QNR; an example might be QNFs at the edge of a domain exchanging messages to set up resources for a flow across a it. Note that it is left open if the responder can release or 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.

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

Operation	Direction	Operation
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 6.1.2)

6.1.2. State Management

The primary 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 [30] or RMD [31] (per 'class' state), respectively.

- The protocol must be robust against failure and other conditions that imply that the stored resource reservation state has to be moved or removed.

For resource reservations, soft-state management is typically used as a general robustness mechanism. According to the discussion of Section 3.2.5, the soft-state protocol mechanisms are built into the NSLP for the specific signaling application that needs them; the NTLP sees this simply as a sequence of (presumably identical) messages.

6.1.3. Route Changes and QoS Reservations

In this section, we will explore the expected interaction between resource signaling and routing updates (the precise source of routing updates does not matter). The normal operation of the NSIS protocol will lead to the situation depicted in Figure 7, where the reserved resources match the data path.

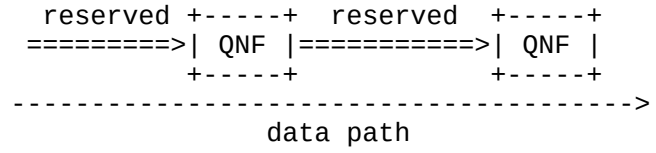

Figure 7: Normal NSIS Protocol Operation

A route change can occur while such a reservation is in place. The route change will be installed immediately, and any data will be forwarded on the new path. This situation is depicted Figure 8.

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, and certain delay or drop-sensitive applications will require that this time interval be minimized. Several techniques to achieve this could be considered. As an example, RSVP [7] has the concept of local repair, whereby 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 QNF. Another approach would be to pre-install backup state, and it would be the responsibility of the QoS-NSLP to do this. However, mechanisms for identifying backup paths and routing the necessary signaling messages along them are not currently considered in the NSIS requirements and framework.

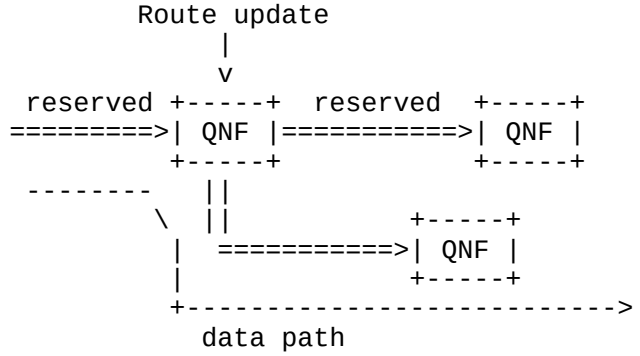

Figure 8: Route Change

The new path might not be able to provide the same guarantees that were available on the old path. Therefore, it might be desirable for the QNF to wait until resources have been reserved on the new path before allowing the route change to be installed (unless, of course, the old path no longer exists). However, delaying the route change installation while waiting for reservation setup needs careful analysis of the interaction with the routing protocol being used, in order to avoid routing loops.

Another example related to route changes is denoted as severe congestion and is explained in [31]. This solution adapts to a route change when a route change creates congestion on the new routed path.

6.1.4. 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). Note, 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 specialized intra-domain QoS signaling (running between just the edges of the network). (See [10], [32], and [33] 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 a QNF (note that co-location with a QNI/QNR can be handled logically as a combination between QNF and QNI/QNR). To cater for both cases, we define a (possibly logical) QNF-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 QNF 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 using 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 develop NSLPs for other signaling applications that 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 [4]. Other examples include network monitoring and testing, and tunnel endpoint discovery.

7. Security Considerations

This document describes a framework for signaling protocols that 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 4.7. We have assumed that, apart from being resistant to denial of service attacks against itself, the main role of the NTLP will be to provide message protection over the scope of a single peer relationship, between adjacent signaling application entities. (See Section 3.2.3 for a discussion of the case where these entities are separated by more than one NTLP hop.) These functions can ideally be provided by an existing channel security mechanism, preferably using an external key management mechanism based on mutual authentication. Examples of possible mechanisms are TLS, IPsec and SSH. However, there are interactions between the actual choice of security protocol and the rest of the NTLP design. Primarily, most existing channel security mechanisms require explicit identification of the peers involved at the network and/or transport level. This conflicts with those aspects of path-coupled signaling operation (e.g., discovery) where this information is not even implicitly available because peer identities are unknown; the impact of this 'next-hop problem' on RSVP design is discussed in the security properties document [6] and also influences many parts of the threat analysis [2]. Therefore, this framework does not mandate the use of any specific channel security protocol; instead, this has to be integrated with the design of the NTLP as a whole.

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 authorization requirements of the signaling application. Authorization is a complex topic, for which a very brief overview is provided in Section 3.3.7.

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 authorization aspects. This complicates the analysis of basing signaling application security on NTLP protection.

An additional concern for signaling applications is the session identifier security issue (Sections 4.6.2 and 5.2). The purpose of this identifier is to decouple session identification (as a handle for network control state) from session "location" (i.e., the data flow endpoints). The identifier/locator distinction has been extensively discussed in the user plane for end-to-end data flows, and is known to lead to non-trivial security issues in binding the two together again. Our problem is the analogue in the control plane, and is at least similarly complex, because of the need to involve nodes in the interior of the network as well.

Further work on this and other security design will depend on a refinement of the NSIS threats work begun in [2].

8. References

8.1. Normative References

[1] Brunner, M., "Requirements for Signaling Protocols", RFC 3726, April 2004.

[2] Tschofenig, H. and D. Kroeselberg, "Security Threats for Next Steps in Signaling (NSIS)", RFC 4081, June 2005.

[3] Chaskar, H., "Requirements of a Quality of Service (QoS) Solution for Mobile IP", RFC 3583, September 2003.

[4] Swale, R., Mart, P., Sijben, P., Brim, S., and M. Shore, "Middlebox Communications (midcom) Protocol Requirements", RFC 3304, August 2002.

8.2. Informative References

[5] Manner, J. and X. Fu, "Analysis of Existing Quality of Service Signaling Protocols", Work in Progress, December 2004.

[6] Tschofenig, H., "RSVP Security Properties", Work in Progress, February 2005.

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

[8] Katz, D., "IP Router Alert Option", RFC 2113, February 1997.

[9] Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, October 1999.

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

[11] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[12] Tschofenig, H., "NSIS Authentication, Authorization and Accounting Issues", Work in Progress, March 2003.

[13] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, April 2001.

[14] Ji, P., Ge, Z., Kurose, J., and D. Townsley, "A Comparison of Hard-State and Soft-State Signaling Protocols", Computer Communication Review, Volume 33, Number 4, October 2003.

[15] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

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

[17] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, November 2000.

[18] Hinden, R., "Virtual Router Redundancy Protocol (VRRP)", RFC 3768, April 2004.

[19] Heijenk, G., Karagiannis, G., Rexhepi, V., and L. Westberg, "DiffServ Resource Management in IP-based Radio Access Networks", Proceedings of 4th International Symposium on Wireless Personal Multimedia Communications WPMC'01, September 9 - 12 2001.

[20] Manner, J., Lopez, A., Mihailovic, A., Velayos, H., Hepworth, E., and Y. Khouaja, "Evaluation of Mobility and QoS Interaction", Computer Networks Volume 38, Issue 2, p. 137-163, 5 February 2002.

[21] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[22] Liebsch, M., Ed., Singh, A., Ed., Chaskar, H., Funato, D., and E. Shim, "Candidate Access Router Discovery (CARD)", Work in Progress, May 2005.

[23] Kempf, J., "Problem Description: Reasons For Performing Context Transfers Between Nodes in an IP Access Network", RFC 3374, September 2002.

[24] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[25] Nordmark, E., "Stateless IP/ICMP Translation Algorithm (SIIT)", RFC 2765, February 2000.

[26] Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489, March 2003.

[27] Terzis, A., Krawczyk, J., Wroclawski, J., and L. Zhang, "RSVP Operation Over IP Tunnels", RFC 2746, January 2000.

[28] Bosch, S., Karagiannis, G., and A. McDonald, "NSLP for Quality-of-Service signaling", Work in Progress, February 2005.

[29] Stiemerling, M., "A NAT/Firewall NSIS Signaling Layer Protocol (NSLP)", Work in Progress, February 2005.

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

[31] Westberg, L., Csaszar, A., Karagiannis, G., Marquetant, A., Partain, D., Pop, O., Rexhepi, V., Szabo, R., and A. Takacs, "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.

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

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

Appendix A. Contributors

Several parts of the introductory sections of this document (in particular, in Sections 3.1 and 3.3) are based on contributions from Ilya Freytsis, then of Cetacean Networks, Inc.

Bob Braden originally proposed "A Two-Level Architecture for Internet Signalling" as an Internet-Draft in November 2001. This document served as an important starting point for the framework discussed herein, and the authors owe a debt of gratitude to Bob for this proposal.

Appendix B. Acknowledgements

The authors would like to thank Bob Braden, Maarten Buchli, Eleanor Hepworth, Andrew McDonald, Melinda Shore, and Hannes Tschofenig for significant contributions in particular areas of this document. In addition, the authors would like to acknowledge Cedric Aoun, Attila Bader, Anders Bergsten, Roland Bless, Marcus Brunner, Louise Burness, Xiaoming Fu, Ruediger Geib, Danny Goderis, Kim Hui, Cornelia Kappler, Sung Hycuk Lee, Thanh Tra Luu, Mac McTiffin, Paulo Mendes, Hans De Neve, Ping Pan, David Partain, Vlora Rexhepi, Henning Schulzrinne, Tom Taylor, Michael Thomas, Daniel Warren, Michael Welzl, Lars Westberg, and Lixia Zhang for insights and inputs during this and previous framework activities. Dave Oran, Michael Richardson, and Alex Zinin provided valuable comments during the final review stages.

Authors' Addresses

Robert Hancock
Siemens/Roke Manor Research
Old Salisbury Lane
Romsey, Hampshire SO51 0ZN
UK

EMail: robert.hancock@roke.co.uk

Georgios Karagiannis
University of Twente
P.O. BOX 217
7500 AE Enschede
The Netherlands

EMail: g.karagiannis@ewi.utwente.nl

John Loughney
Nokia Research Center
11-13 Itamerenkatu
Helsinki 00180
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

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.