NSIS: A New Extensible IP Signaling Protocol Suite

Xiaoming Fu, University of Göttingen; Henning Schulzrinne, Columbia University; Attila Bader, Ericsson
Dieter Hogrefe, University of Göttingen; Cornelia Kappler, Siemens
Georgios Karagiannis, University of Twente; Hannes Tschofenig, Siemens Corporate Technology
Sven Van den Bosch, Alcatel

ABSTRACT

In the last few years a number of applications have emerged that can benefit from network-layer signaling (i.e., the installation, maintenance, and removal of control state in network elements). These applications include path-coupled and path-decoupled quality of service management and resource allocation, as well as network debugging, NAT, and firewall control. These applications call for an extensible and securable signaling protocol. This article discusses some of the recent standardization efforts in the IETF for a new extensible IP signaling protocol suite (NSIS). We describe the design of the NSIS protocol suite, and compare it with RSVP, the current Internet QoS signaling protocol.

INTRODUCTION

Signaling in communication networks is defined as the exchange of information between nodes to establish, maintain, and remove control state in network nodes. The concept of signaling is not new. The industry recognized the need for a way to create and remove circuits, each associated with an end-to- end communications channel, for transporting information over long-haul networks. As a result they developed Signaling System 7 (SS7) for signaling in telephone networks. It nonetheless took until the last decade for network designers to use signaling to improve the ability of packet-switching networks to support emerging services, especially real-time services.

With the increasing diversity of services offered across the Internet, there is a new need for signaling over IP-based networks. Examples include reserving resources to provide quality of service (QoS) guarantees, configuring firewall pinholes and network address translator (NAT) bindings, and diagnosing path status. The Internet Engineering Task Force (IETF) developed the Resource Reservation Protocol (RSVP) [1, 2], but RSVP has been designed and applied to resource reservation for both integrated services (IntServ) and later differentiated services (DiffServ), rather than more general signaling services. This led the IETF in 2001 to form a new working group, Next Steps in Signaling (NSIS), to investigate a more flexible IP signaling architecture and protocols.

Due to the shortcomings of RSVP and its current extensions, the NSIS working group began to work on a new protocol suite in order to accommodate new signaling needs. As a result, an extensible IP signaling architecture [3] was developed, also referred to as NSIS. It consists of two layers: the lower layer provides a generic transport service for different signaling applications, which reside in the upper layer. The main part of the lower layer is the General Internet Signaling Transport (GIST) protocol [4]. Examples of upper layers (i.e., signaling applications) are QoS signaling, and firewall and NAT control. Furthermore, the working group decided to provide a mechanism to decouple next node discovery from signaling message delivery. This allows more flexibility, such as the ability to use standard transport layer and security protocols.

This article provides information about current standardization efforts for the NSIS protocol suite. The rest of the article is organized as follows. In the next section we provide an overview of RSVP and point out why it needs to be updated. We then briefly describe the NSIS signaling approach. There follows a discussion of GIST and application protocols, particularly the QoS signaling protocol [5]. We also present security aspects and the implementation status of NSIS protocols, and compare them to RSVP, in particular with respect to QoS signaling.

SOFT STATE SIGNALING AND THE RSVP SIGNALING PROTOCOL

Signaling protocols can use either a hard state or soft state approach. Hard state is installed in nodes upon receipt of a setup message and removed only upon receipt of an explicit tear-

0163-6804/05/$20.00 © 2005 IEEE

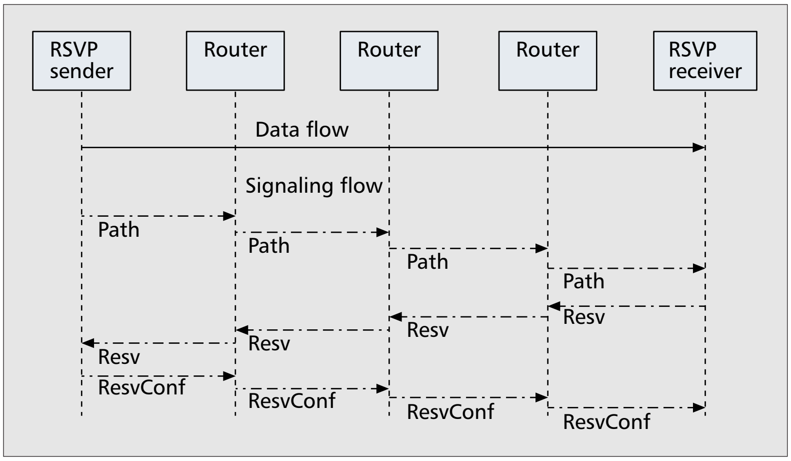


Figure 1. *RSVP basic protocol operation.*

down message. SS7 and ST-2 are examples of hard state protocols. In contrast, soft state refers to nonpermanent control state in network nodes that will expire unless refreshed. It was first formalized in RSVP. Signaling built on the soft state paradigm has been adopted by many other proposals, such as Boomerang and YESSIR [6].

RSVP has two main control messages: *Path* messages, which originate from the flow sender(s) and travel toward the receiver(s), discover RSVP aware routers, and establish routing state information to allow routing of messages in the reverse direction; and *Resv* messages, which travel in the reverse direction from the receiver(s) to the sender(s) and install reservation state. RSVP has been designed to support many-to-many multicast QoS reservations. A lot of effort went into addressing reservation merging and related killer reservation problems.

Setting up a reservation with RSVP works as follows. Upon receipt of a Path message, the RSVP module in each router records the previous RSVP router's address, creates or refreshes a Path state, and forwards the Path message toward the receiver(s). Upon receipt of the Path message at the final destination, a Resv message is sent backward. Every RSVP router receiving a Resv message creates or refreshes its Resv state, and forwards it to the previous RSVP router until it reaches the sender. Resv messages containing an optional confirmation-request object will cause each sender (or state merging point) to respond with a *ResvConf* message back to the receiver. Figure 1 illustrates an example signaling flow.

Originally, RSVP signaling was per-flow-based, and simply relied on periodical refreshes between routers for reliability of control message delivery. Later, reservation aggregation and hop-by-hop reliability were added [2, 7]; [2] also introduced message bundling and summary refresh mechanisms to reduce the RSVP refresh overhead.

The traffic engineering extension for RSVP, known as RSVP-TE [8], has been widely used for traffic engineering in IP networks with the development of multiprotocol label switching (MPLS). RSVP-TE uses RSVP in a special way. First, it is used in closed environments, typically within a single administrative domain, avoiding some of the security issues. Second, instead of signaling messages following the data path, RSVP-TE follows a *label switching path* for MPLS networks, which may be determined manually. The RSVP-TE signaling message then establishes the path later taken by the data traffic. Thus, the concept of path discovery is less relevant.

During the work of the NSIS working group, several key problems surfaced in the use of RSVP [6, 9]:

- RSVP was designed when node mobility was in its infancy and therefore does not support mobile nodes. Conversely, it expended a lot of effort to work well with IP multicast, which has not been widely deployed.
- RSVP's choice of transport mechanism imposes constraints on network architectures and signaling applications. For example, raw IP or User Datagram Protocol (UDP) is used for carrying RSVP messages across the network where Path messages are addressed end-to-end with an IP router-alert option. RSVP relies on nonadaptive retransmissions for reliability. Since fragmentation of messages is not allowed, the message length is limited to the maximum transport unit (MTU) size.
- A general need of signaling is to discover and signal to a chain of signaling-aware nodes in a hop-by-hop fashion along the data path. In RSVP, discovery and signaling message delivery are combined into a single protocol step. This design decision makes it difficult to offer proper security protection using existing security protocols as RSVP does not provide a solid security framework, especially for end-to-end addressed signaling messages (e.g., Path). Authentication and key management are not adequately addressed, and only manual configuration of crypto keys is supported. Authorization aspects are provided to some degree. but do not interwork with today's authentication, authorization, and accounting (AAA) infrastructure (e.g., DIAMETER and RADIUS) and in a roaming environment.

These problems make RSVP unsuitable as a general-purpose network-layer signaling protocol. Below, we discuss the design principles the NSIS effort applied to address these and other issues.

Design Principles of NSIS

As described earlier, the IETF NSIS working group is focusing on the development of a new signaling protocol suite for the manipulation of state along the data path, referred to as path-coupled signaling. (Path-decoupled signaling has also recently been investigated [10].)

An example NSIS signaling scenario is shown in Fig. 2. NSIS entities that communicate with each other are said to have a *peer* relationship. Each entity may store state information about its peers, but is not required to do so. One node, the NSIS initiator (NI), initiates signaling, while some nodes along the signaling path, called NSIS forwarders (NFs), intercept and then forward signaling messages, and the NSIS responder (NR) terminates the signaling. Figure 2 also shows that not all routers along the data path need to be NSIS-aware; nor do all NSIS nodes necessarily support all signaling applications. For a particular NSIS session, nodes not supporting the signaling application contained in an NSIS message are skipped. In this example, messages of signaling application type A will be delivered between R2 and the edge node, without being processed in R3.

The fundamental design choices for the NSIS protocol suite are summarized below.

Separating Signaling Message Transport from Signaling Applications

In order to meet the requirements for extensible, generic signaling, the design of the NSIS protocol suite separates the functionalities such as reliability, fragmentation, congestion control and integrity for signaling message transport from signaling applications. Thus, architecturally, NSIS consists of two protocol layers [3]:

- An NSIS Transport Layer Protocol (NTLP), primarily composed of a specialized messaging layer, denoted GIST [4], which is used to transport the signaling application layer messages. The GIST layer runs over standard transport and security protocols. Examples of such transport protocols are UDP, TCP, Stream Control Transmission Protocol (SCTP), and Datagram Congestion Control Protocol (DCCP).
- NSIS Signaling Layer Protocols(NSLPs), each running signaling application-specific functionality, including formats and processing rules of messages to be exchanged between NSLP peers. Examples of NSLPs include the QoS NSLP for resource reservation signaling [5], the NAT/firewall NSLP [11] for middlebox configuration, and a possible NSLP for configuration of metering entities [12].

The different layers are depicted in Fig. 3.

Decoupling of Discovery and Transport of Signaling Messages

Another key design choice made during the NSIS work is to decouple NSIS peer discovery from the signaling message transport mechanism. As mentioned earlier, RSVP combines discovery and signaling message delivery into a single protocol step, thereby preventing standard security protocols or transport layer protocols from being used. NSIS resolves this dilemma by introducing a discovery component in GIST that can rely on IP router alert options or other approaches, such as routing tables.

Introduction of a Session Identifier

In NSIS, similar to RSVP, a data flow is defined as a unidirectional sequence of packets between the same endpoints that all follow a unique path through the network. They are identified by a flow identifier (e.g., a 5-tuple or a DSCP field). Unlike RSVP, NSIS offers a *session identifier*. A session identifier is a cryptographically random number used to probabilistically uniquely identify a signaling session and signaling state, independent of the flow identifier. A session may map to a specific flow, but for some scenarios signaling applications may create more flexible flow-session relationships:

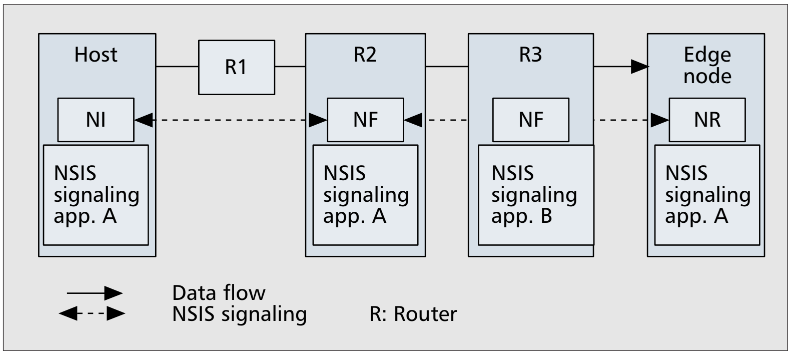


Figure 2. *An NSIS signaling scenario between a host and an edge node.*

- Mobility: During handover, the source or destination IP address of an end host, and hence the flow identifier, may change. This does not affect its installed reservation if the associated session can be remapped to the updated flow identifier.
- Multihoming: In this case, multiple flow identifiers can be mapped to the same session.
- Tunneling and IPv4/v6 traversal: When NSIS signaling messages traverse NSIS-aware IPv4/v6 borders or other tunneling devices, while the session identifier will remain the same, the flow identifiers may be remapped into a different one, depending on the signaling application when entering the region.

Typically, a session carries opaque per-flow information specific to its NSLP. This information might be related to resource reservation, or some other control function in routers and middleboxes along the path.

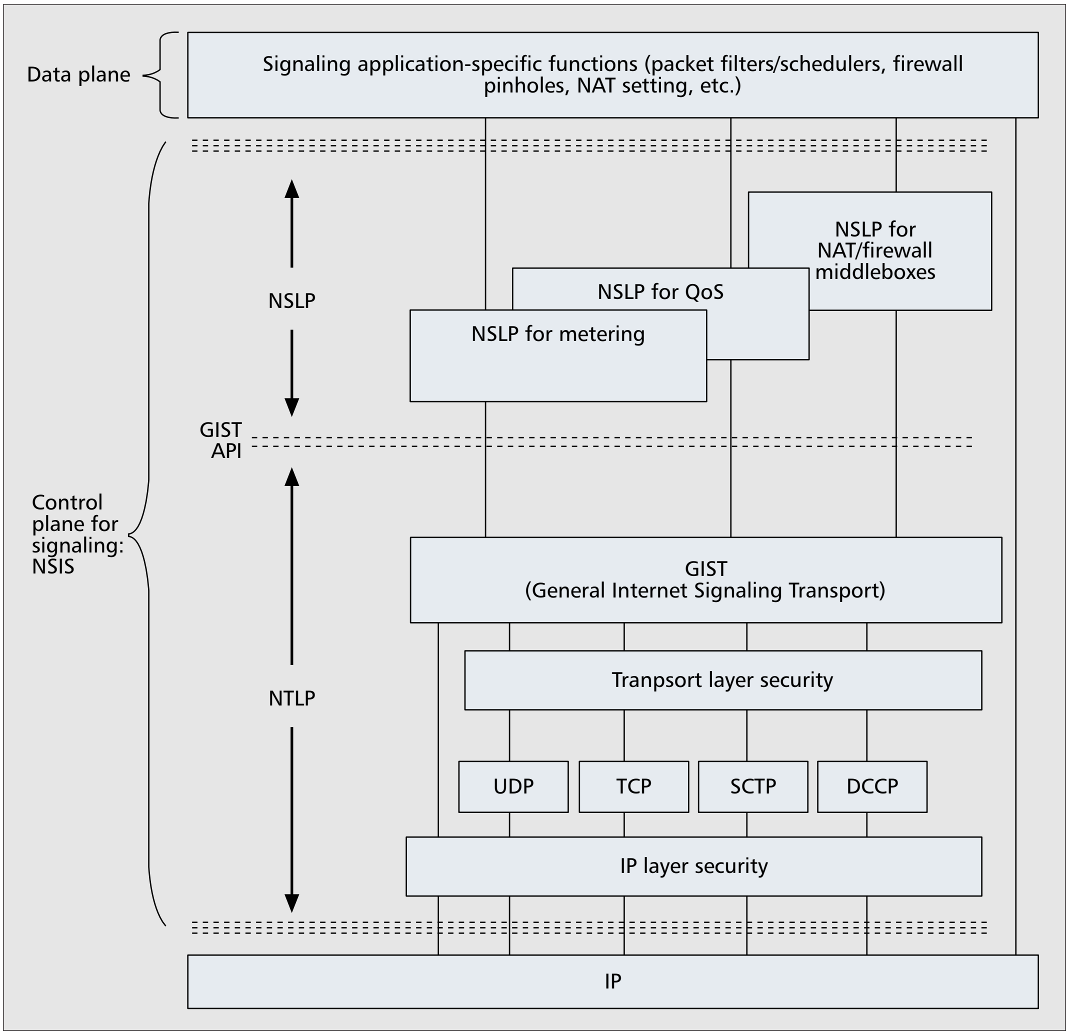
Support for Signaling to Hosts, Networks, and Proxies

NSIS signaling is applicable in different parts of the Internet and may be triggered in different ways. This is required to allow the signaling to be initiated and terminated in different parts of the network: at end hosts, at domain boundaries (edge nodes), and at interior routers. The NSIS protocol suite thus supports many different signaling exchanges, including end-to-end signaling where the exchange is performed between end hosts, edge-to-edge signaling where the boundary nodes of a domain might communicate directly, and end-to-edge signaling, such as in host-to-network scenarios.

GIST: General Internet Signaling Transport Protocol

The NSIS Transport Layer Protocol, as noted earlier, forms the fundamental building block of the NSIS protocol suite. The main task of NTLP is to deliver signaling messages for various NSLPs from the NI toward the NR, typically the flow source and destination, respectively. The NI and NR can, however, also be represented by proxies (e.g., to support end systems that do not

GIST has two modes of operation: the datagram mode, which uses an unreliable unsecured datagram transport mechanism, with UDP as the initial choice; and the connection mode, which uses any stream or message-oriented transport protocol, with TCP as the initial choice.



Data plane
Signaling application-specific functions (packet filters/schedulers, firewall pinholes, NAT setting, etc.)
Control plane for signaling: NSIS
NSLP
NSLP for NAT/firewall middleboxes
NSLP for QoS
NSLP for metering
GIST API
GIST (General Internet Signaling Transport)
NTLP
Tranpsort layer security
UDP
TCP
SCTP
DCCP
IP layer security
IP

Figure 3. *Logical components in an NSIS-aware node.*

themselves have NSIS capabilities). NTLP is implemented by the GIST protocol.

Instead of building a new transport protocol, GIST reuses existing transport and security protocols to provide a universal message transport service. Like RSVP, GIST is a soft state protocol. It creates and maintains two types of states related to signaling transport: a per-flow message routing state for managing the processing of outgoing messages, and a message association state for managing per-peer state associated with connection mode messaging to a particular peer. The latter consists of signaling destination address, protocol and port numbers, as well as internal protocol configuration and state information. In addition to information about its neighbor NTLP peer, GIST also maintains certain message routing information such as the flow identifier, NSLP type, and session identifier to uniquely identify the signaling application layer session for a flow.

GIST has two modes of operation: the *datagram mode*, which uses an unreliable unsecured datagram transport mechanism, with UDP as the initial choice; and the *connection mode*, which uses any stream or message-oriented transport protocol, with TCP as the initial choice. It may employ network layer security associations such as IPsec, or a transport-layer security association such as TLS. It is possible to mix these two modes along a chain of nodes, without coordination or manual configuration. This allows, for example, the use of datagram mode at the edges of the network and connection mode in the core of the network.

We explain the operation of GIST using the example in Fig. 2, where A is QoS NSLP, while B is another NSLP. Assume a QoS signaling message is processed by GIST at the NI, the host. The GIST module first constructs a GIST-query message, a UDP datagram, possibly including a QoS NSLP payload. The message is addressed to the flow destination and labeled with a router alert option, similar to RSVP. The next downstream NSIS peer that supports the QoS NSLP (R2) recognizes this message, and passes the NSLP payload and flow identifier to its QoS NSLP process. It also recognizes the upstream NSIS peer who wants to learn about its downstream peer, and thus answers with a GIST response message. Upon receipt of this response, the upstream NSIS peer creates a message association with the downstream peer (here, R2), using, say, TCP. All subsequent NSIS messages between these two peers can now be sent via this message association.

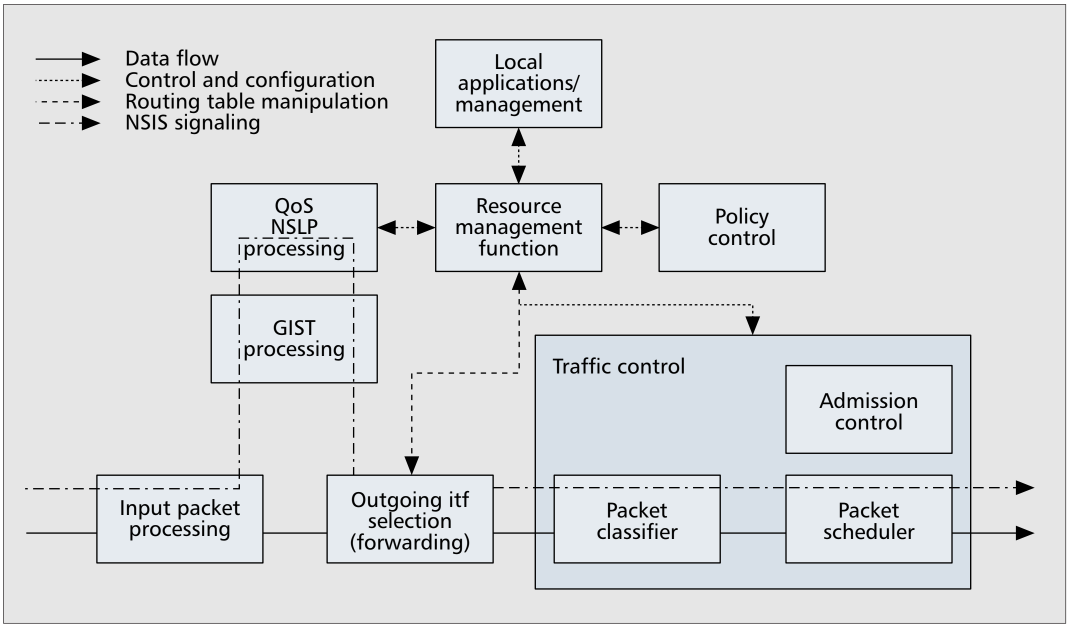


Figure 4. *QoS NSLP model of operation.*

Since it is possible that not all NSLPs are supported in a single NSIS node, in route change cases, GIST cannot carry out the complete path update processing for both NSLP and GIST states. Rather, GIST can detect the route change, update its own routing state consistently, and inform interested signaling applications at affected nodes.

A GIST message consists of a common header and a sequence of type-length-value (TLV) objects. The common header indicates whether it is a datagram mode or connection mode message, whether it is headed upstream or downstream, as well as the NSLP type and hop counter to avoid message loops. In addition, GIST uses query and response cookies for protection against denial-of-service attacks.

GIST query messages are retransmitted with exponential backoff if a corresponding response is not received on time. Other NSLP messages encapsulated in datagram mode are not retransmitted; they rely on initial query messages that are eventually resent. Whenever possible, reuse of existing reliable transport and security protocols is recommended via the connection mode in GIST. Connection mode is necessary for larger data objects, when fast state setup in the face of packet loss is desirable, or where channel security is required. A querying node can choose to refresh the message routing state by resending a GIST query. However, whether to maintain messaging association is determined by local policy. For example, a node may choose to retain the association if there are flows still in place that might generate messages using it. Advanced features are described below.

Message association negotiation: GIST messages can include a stack proposal object, so a node can propose and negotiate about the stack forming the message association (i.e., which combinations of transport and security protocols are used).

Rerouting: Since it is possible that not all NSLPs are supported in a single NSIS node, in route change cases, GIST cannot carry out the complete path update processing for both NSLP and GIST states. Rather, GIST can detect the route change, update its own routing state consistently, and inform interested signaling applications at affected nodes.

Interaction with NAT and IP tunneling: GIST messages carry IP addresses and port numbers in their payloads (to specify the flow for which they are signaling), and typically NSLPs do not carry data that is affected by NATs. Once a NAT or tunneling device is GIST-aware, it can modify GIST datagram mode messages. Subsequent connection mode messages are not affected by NATs because the NSIS peers address each other directly.

The QoS Signaling Application Protocol in NSIS

Model of Operation

This section presents a logical model for the operation of QoS NSLP and associated provisioning mechanisms within a single node. The model is shown in Fig. 4.

From the perspective of a single node, the request for QoS may result from a local application request, initiated by a user application or network management (e.g., initiating a tunnel to handle an aggregate), or processing an incoming QoS NSLP message. The *incoming message* case requires NSIS messages to be captured during input packet processing and handled by GIST. Only messages related to QoS are passed up to the QoS NSLP processing module.

QoS NSLP can signal for any QoS model (e.g., IntServ or DiffServ). A controlled load service QoS model over NSIS is described in [13]. It can also carry, for example, Third Generation Partnership Project (3GPP)-specific QoS parameters. Reservation-specific parameters (e.g., available bandwidth or token bucket sizes), encapsulated in a QSPEC object [14], are carried from one QoS NSLP node to another. QSPEC parameters provide a common language to be reused in several QoS models, ensuring some degree of interoperability.

The QoS NSLP node may need to indicate that the required resources have been configured. Therefore, it may generate an acknowledgment message in one direction, and may propagate the resource request further along the path towards the data receiver.

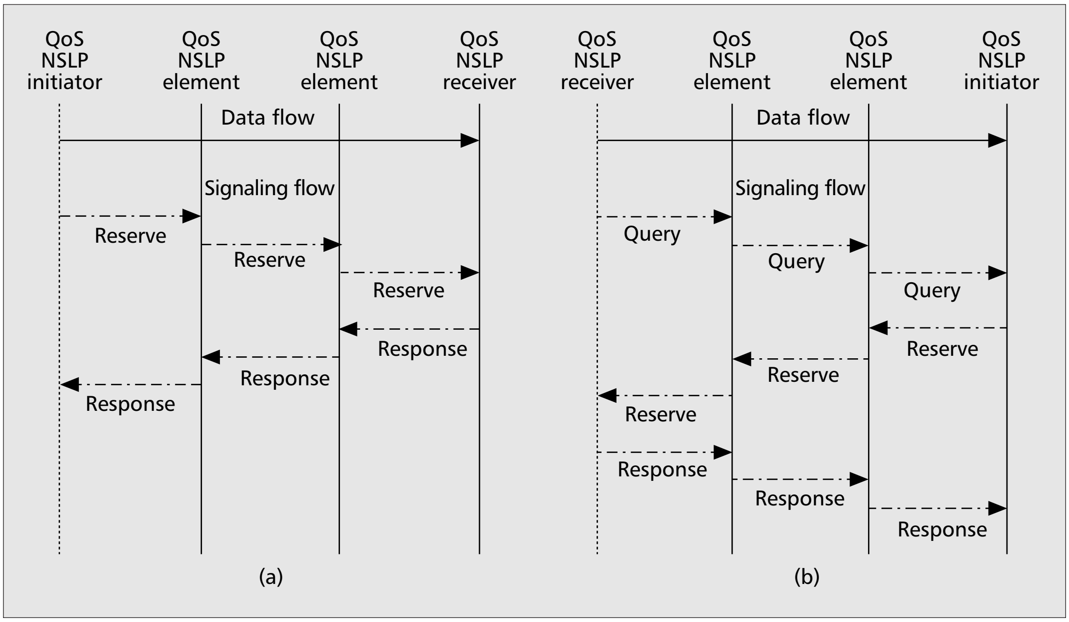


Figure 5. *Basic a) sender-initiated and b) receiver-initiated protocol operation.*

In each QoS NSLP node, the QoS request, specifically the QSPEC, is handled by a resource management function (RMF). The local QoS model describes how the RMF should interpret the QSPEC, and how to grant and configure the resource. The grant processing involves two additional local decision modules, *policy control* and *admission control*. Finally, the QoS NSLP node may need to indicate that the required resources have been configured. Therefore, it may generate an acknowledgment message in one direction, and may propagate the resource request further along the path toward the data receiver.

PROTOCOL BASICS

QoS NSLP is a soft state protocol. It defines four messages:

- The *RESERVE message* is the only message that manipulates QoS NSLP reservation state. It is used to create, refresh, modify, and remove such state.
- The *QUERY message* requests information about the data path without making a reservation. This functionality can be used to "probe" the network for path characteristics, receiver-initiated reservations, or support of certain QoS models.
- The *RESPONSE message* provides information about the result of a previous QoS NSLP message.
- The *NOTIFY message* can be used to convey information to a QoS NSLP node. It differs from a RESPONSE message in that it is sent asynchronously and need not refer to any particular state or previously received message. The information conveyed by a NOTIFY message is typically related to error conditions.

QoS NSLP messages are sent NSIS-peer-to-NSIS-peer. In contrast to RSVP, it supports both sender-initiated and receiver-initiated reservations. The messaging flows for basic sender-initiated and receiver-initiated reservations are shown in Fig. 5.

ADVANCED MECHANISMS

QoS NSLP further supports a number of advanced mechanisms that allow it to be used in more complex signaling scenarios:

- *Summary refreshes* allow an abbreviated form of refreshing the RESERVE message.
- *Message scoping* allows the use of local policy to decide whether or not to forward a message.
- *Session binding* enforces a relation between different sessions. This information may be used for local optimization in case of bidirectional or aggregate reservations; QoS NSLP supports aggregation facilities similar to [7] as well as other application scenarios such as signaling for local QoS parameters and models.
- *Route change detection*: QoS NSLP is able to detect route changes and automatically adapt to the new routes.
- *Reduced state*: QoS NSLP does not mandate that each QoS NSLP node store QoS reservation state. It supports a "reduced-state" operation, where reservation states with a coarser granularity (e.g., per-class) are used, or a stateless operation where no QoS NSLP state is created. An example of this operation is resource management in DiffServ (RMD) [15], which can be used to signal DiffServ to core routers.

SECURITY CONSIDERATION FOR NSIS

Securing the NSIS protocol suite introduces some challenges, since NSIS aims to support a large number of scenarios, involves a series of signaling entities, and needs to accommodate devices from high-performance servers in corporate networks to mobile devices, a variety of cryptographic mechanisms (symmetric and asymmetric cryptography, different authentication protocols), and existing network architectures, such as the PacketCable or 3GPP architectures.

Since the NSIS protocol suite is split into two layers, the NSIS security solution needs to offer

	RSVP	NSIS
Protocol structure	Single layer	Two layers
Transport	IP or UDP	Reliable (TCP,STCP)/datagram (UDP,DCCP)
Reservation initiator	Receiver	Sender or receiver
States	Soft + expl. release	Soft + expl. release
QoS models	IntServ/DiffServ	IntServ/DiffServ/other
Scope of signaling	End-to-end	End-to-end/host-to-edge/edge-to-edge
Multicast	Yes	No
Mobility	No	Yes
Bi-directional	No	Yes
Aggregation	Yes	Yes
Summary refresh	Yes	Yes
Priority/preemption	Yes	Yes

Table 1. *Summary of the basic features of RSVP and NSIS (QoS) signaling.*

For QoS authorization, the decision might be related to the ability of the user to pay for the treatment. Making an authorization decision to create a NAT binding might depend on the traffic direction either from the private network to the Internet or vice versa.

security protection for both layers. Security protection for GIST in connection mode needs to offer the following properties:

1 Authentication of the two neighboring protocol peers
2 Security association establishment to provide integrity, confidentiality, and replay protection for signaling messages exchanged between these entities
3 Denial of service protection
4 Authorization of the signaling peers
5 Some security protection for the discovery mechanism

It is difficult to design a new security protocol addressing all these issues. Existing security protocols such as TLS or Internet Key Exchange version 2 (IKEv2)/IPsec already provide a number of these features, addressing issues 1, 2, and 3, but at a cost of considerable setup latency. This setup latency can be amortized across many messages and across sessions if signaling peers use the GIST connection mode. If NSIS sessions are established only between nodes that support the same NSLP, peers can verify peer identity and authorize those peers.

Authorization at the GIST layer aims to ensure that a GIST responder only establishes communication with a legitimate GIST initiator. However, in most cases it is quite difficult for GIST alone to make such a decision, and it typically needs to consult with the NSLP layer. Still worse, it is even more difficult to ensure that the GIST initiator sends signaling messages to the "right" GIST peer (i.e., one that supports a specific NSLP); this requires authorization information to be provided along with the authentication and key exchange process (e.g., as part of authorization certificates). Furthermore, to deal with adversaries redirecting signaling messages, additional security mechanisms have been integrated into the discovery exchange, such as cookies [4].

Most authorization decisions will, however, be executed at the NSLP. For QoS authorization, the decision might be related to the ability of the user to pay for the treatment. Making an authorization decision to create a NAT binding might depend on the traffic direction either from the private network to the Internet or vice versa.

In many cases, an individual NSIS router will be unable to make an authorization decision, particularly in mobile environments, so it may contact the AAA infrastructure to delegate the decision. To avoid reauthorization at different protocol layers (e.g., at the application layer using SIP and again in NSIS), it is possible to combine these independent protocol runs with the help of authorization tokens.

Comparison between RSVP and NSIS (QoS) Signaling

The basic differences between RSVP and NSIS, particularly with respect to QoS signaling, are summarized in Table 1. Some of them are further elaborated below.

Transport of signaling messages: As explained earlier, RSVP messages are transported unreliably by UDP or directly over IP. In NSIS, the message transport mechanism (NTLP) is separated from the signaling application layer (NSLP), thus allowing different signaling applications. Different applications and sessions can share the same message associations, so it is not necessary to create message associations for each session. NTLP uses existing transport protocols, including UDP and TCP.

Reservation model: The RSVP reservation model is receiver-initiated, and its signaling

The current NSIS next peer discovery mechanism relies on the router alert option, which may be deprecated by some operators especially in border routers. Therefore, alternative discovery mechanisms may need to be developed, for example by extending routing protocols or creating a new DNS namespace.

extends from flow sender to flow receiver. NSIS QoS NSLP supports both sender- and receiver-initiated reservations. Proxy operation is supported; that is, NSIS messages can be initiated and terminated in nodes other than the source or end of the data flow. Thus, the scope of NSIS signaling can be end-to-end, edge-to-edge, host-to-edge, or edge-to-host.

Multicast: Unlike RSVP, NSIS does not support multicast, reducing complexity for the majority of user applications which are unicast. However, the basic NSIS protocol model is likely extensible to IP multicast.

Bidirectional reservation: The QoS NSLP supports bidirectional reservations by binding the sessions in both directions. There is no such support in RSVP.

QoS models: NSIS QoS NSLP allows signaling any QoS model.

Mobility Support: By identifying signaling sessions by a random session identifier, rather than by a flow identifier including the IP address, NSIS can support mobility more easily [16].

Security: Whereas security was added to RSVP after its basic design, NSIS has been designed with security in mind from the beginning, integrating standard security protocols, such as TLS or IPsec/IKEv2. These protocols offer features such as flexible authentication methods, negotiation of crypto algorithms, extensively verified protocols, and denial of service protection.

Similar to RSVP, NSIS can also bypass non-NSIS nodes, as explained earlier. Furthermore, as NSIS discovery is based on NSLP type, an NSLP protocol message of type x skips those NSIS nodes not supporting x as non-NSLPx clouds, eliminating the need to maintain state in those nodes and reducing signaling latency.

Implementation Status

From the beginning of the NSIS discussions, implementations were used to validate the feasibility of the designs proposed. Students at the University of Kentucky and University of Göttingen completed implementations of early drafts, with modular interfaces of generic signaling services for NSLPs. We know of implementation activities at Siemens Roke Manor Research, NEC, Nokia, Alcatel, and the University of Coimbra; Ericsson, the University of Karlsruhe, the University of Twente, and Samsung are working toward independent implementations of QoS NSLP.

The University of Göttingen demonstrated its open source GIST implementation at the NSIS interim meeting in May 2005. The first interoperability tests took place in July 2005.

Conclusion

The development of an extensible IP signaling protocol suite in the IETF NSIS working group has attracted the interest of researchers and industry for both its intrinsic new features and its practical applications for broader signaling purposes. Unlike RSVP, NSIS assumes a two-layer extensible signaling architecture, and reuses existing transport and security mechanisms. It utilizes a session identifier independent of the flow identifier for state management and a discovery component to determine the next NSIS peer, and integrates security from the start.

The working group plans to submit the protocol specifications (e.g., [4, 5, 11]) to the IESG as proposed standards in the second half of 2005, after early implementations have been demonstrated to interwork.

We believe that the extensibility, security, and mobility features of the NSIS protocols will speed their deployment. However, more detailed investigations into mobility issues, as well as formal validation and verification of these protocols will be necessary. In addition, NSIS protocols will likely need to interwork with the existing RSVP protocol.

The current NSIS next peer discovery mechanism relies on the router alert option, which may be deprecated by some operators, especially in border routers. Therefore, alternative discovery mechanisms may need to be developed, for example, by extending routing protocols or creating a new DNS namespace.

The current QoS NSLP mechanism does not address how users may pay a price premium for guaranteed access to resources. Integration with AAA or micropayment mechanisms integrated into NSLP need to be explored.

Acknowledgment

This article builds on the contributions of a large number of individuals to the NSIS protocol suite in the IETF NSIS working group; in particular, the authors would like to acknowledge John Laughney and Robert Hancock. The RFCs and Internet drafts should be consulted for proper attribution.

References

[1] R. Braden *et al.*, “Resource Reservation Protocol (RSVP) — Version 1 Functional Specification,” RFC 2205, Sept. 1997; http://www.rfceditor. org/rfc/rfc2205.txt

[2] L. Berger *et al.*, “RSVP Refresh Overhead Reduction Extensions,” RFC 2961, Apr. 2001; http://www.rfc-editor.org/rfc/rfc2961.txt

[3] R. Hancock *et al.*, “Next Steps in Signaling: Framework,” RFC 4080, June 2005; http://www.rfc-editor.org/rfc/rfc4080.txt

[4] H. Schuzrinne and R. Hancock, “GIMPS: General Internet Messaging Protocol for Signaling,” Internet draft, work in progress, July 2005; http://www.ietf.org/internet-drafts/draft-ietf-nsis-ntlp-07.txt

[5] J. Manner *et al.*, “NSLP for Quality-of-Service signaling,” Internet draft, work in progress, July 2005; http://www.ietf.org/internet-drafts/draftietf- nsis-qos-nslp-07.txt

[6] J. Manner and X. Fu, “Analysis of Existing Quality-of-Service Signaling Protocols,” RFC 4094, May 2005; http://www.rfceditor. org/rfc/rfc4094.txt

[7] F. Baker *et al.*, “Aggregation of RSVP for IPv4 and IPv6 Reservations,” RFC 3175, Sept. 2001; http://www.rfc-editor.org/rfc/rfc3175.txt

[8] D. Awduche *et al.*, “RSVP-TE: Extensions to RSVP for LSP tunnels,” RFC 3209, Dec. 2001; http://www.rfc-editor.org/rfc/rfc3209.txt

[9] H. Tschofenig and R. Graveman, “RSVP Security Properties,” Internet draft, work in progress, Feb. 2005; http://www.ietf.org/internet-drafts/draft-ietf-nsis-rsvp-sec-properties-06.txt

[10] R. Hancock *et al.*, “A Problem Statement for Path-Decoupled Signaling in NSIS,” Internet draft, work in progress, July 2005; http://www.ietf.org/internetdrafts/draft-hancock-nsis-pds-problem.txt

[11] M. Stiemerling, H. Tschofenig, and C. Aoun, "NAT/Firewall NSIS Signaling Layer Protocol (NSLP)," Internet draft, work in progress, July 2005; http://www.ietf.org/internet-drafts/draftietf- nsis-nslp-natfw-07.txt

[12] F. Dressler *et al.*, "NSLP for Metering Configuration Signaling," Internet draft, work in progress, July 2005; http://www.ietf.org/internetdrafts/ draft-dressler-nsis-metering-nslp-02.txt

[13] C. Kappler and X. Fu, "A QoS Model for Signaling IntServ Controlled- Load Service with NSIS," Internet draft, work in progress, July 2005; http://www.ietf.org/internet-drafts/draft-kappler-nsisqosmodel- controlled-load-02.txt

[14] J. Ash, A. Bader, and C. Kappler, "QoS-NSLP QSPEC Template," Internet draft, work in progress, July 2005: http://www.ietf.org/internet-drafts/draft-ietf-nsis-qspec-05.txt

[15] A. Bader *et al.*, "QOSM — The Resource Management in Diffserv QOS Model," Internet draft, work in progress, June 2005; http://www.ietf.org/internet-drafts/draft-ietf-nsis-rmd-03.txt

[16] S. Lee *et al.*, "Applicability Statement of NSIS Protocols in Mobile Environments," Internet draft, work in progress, July 2005; http://www.ietf.org/internet-drafts/draft-ietf-nsis-applicabilitymobility- signaling-02.txt

BIOGRAPHIES

XIAOMING FU (fu@cs.uni-goettingen.de) received a Ph.D. degree from Tsinghua University, China, in 2000. He was a research staff member at Technical University Berlin before joining the University of Göttingen as a lecturer and researcher. His research interests include network architectures, mobile networks, protocol analysis, and design. He is a co-author of RFC 4094 and over 30 research papers. Currently, he serves as TPC member for ICDCS '06 and ICC '06, and an Expert of ETSI STFs for IPv6 Interoperability.

HENNING SCHULZRINNE (hgs@cs.columbia.edu) received degrees from Darmstadt (Germany) University of Technology, the University of Cincinnati, and the University of Massachusetts in Amherst. He held research positions at GMD Fokus, Berlin, and Bell Laboratories before joining the faculty of Columbia University, New York. He is currently chairing the Department of Computer Science. His research interests encompass real-time network services, ubiquitous and mobile computing, and network reliability. He is a co-author of numerous RFCs, including RTP, RTSP, SIP, and GIST.

ATTILA BADER (attila.bader@ericsson.com) obtained his Ph.D. degree in physics in 1997 from Kossuth University, Debrecen, Hungary. Since 2001 he has worked for Ericsson Research in ATM and IP transport solutions for mobile networks. He has participated in IETF and 3GPP QoS standardization work. He is interested in QoS in 3G and next-generation mobile networks as well as in optical networking.

DIETER HOGREFE received his Diploma degree and Ph.D. from the University of Hannover, Germany. His research activities are directed toward computer networks and software engineering. After years of research work in Siemens, German Telecom, and the University of Hamburg, he holds full professorships at the Universities of Bern and Lübeck. Since 2002 he is a professor of telematics at the University of Göttingen. He is chairman of the ETSI Technical Committee on Methods for Testing and Specification (MTS).

CORNELIA KAPPLER (cornelia.kappler@siemens.com) studied physics at Munich University, Harvard University, and the University of Toronto. In 1995 she received a Ph.D. from the University of Toronto. Later she switched fields into communication networks, working for NEC Networking Laboratories, Berlin, Germany, and, since 2000, for Siemens Communications. Her current research interests focus on mobile networks, particularly 4G networks, and signaling protocols.

GEORGES KARAGIANNIS (karagian@cs.twente.nl) holds Ph.D. and M.Sc. degrees from the University of Twente, the Netherlands. From 1994 to 1998 he worked as a researcher at the same university and then joined Ericsson Eurolab Netherlands. Since April 2003 he has been an assistant professor at the University of Twente. His research interests are in the fields of fixed, mobile, and wireless (inter)networking, end-to-end QoS signaling and provisioning, mobility and routing in logical access (overlay) networks, and performance evaluation.

HANNES TSCHOFENIG received a Diploma degree from the University of Klagenfurt, Austria. He joined Siemens Corporate Technology in 2001, where he is currently a senior research scientist. His research focuses on security issues, especially in mobile communications. He is chair of the IETF Emergency Context Resolution with Internet Technologies (ECRIT) working group and secretary of the NSIS working group. He is a co-author of RFC 3726, RFC 4081, and a number of Internet drafts.

SVEN VAN DEN BOSCH (sven.van_den_bosch@ alcatel.de) received his Master's degree and Ph.D. in electrical engineering from Ghent University, Belgium, in 1995 and 1999, respectively. After completing his studies, he started working as a research engineer on developing a traffic engineering methodology within Alcatel's Network Strategy Group. His research interests also include next-generation signaling and QoS, with a focus on the IETF NSIS work. He currently works as a product manager for remote home network management products.

As NSIS discovery is based on the NSLP type, an NSLP protocol message of type x skips those NSIS nodes not supporting x as non-NSLPx clouds, eliminating the need to maintain state in those nodes and reducing signaling latency.