TOC

Network Working Group	J. Manner
Internet-Draft	TKK
Intended status: Informational	R. Bless
Expires: September 10, 2009	Univ. of Karlsruhe
	J. Loughney
	Nokia
	E B. Davies, Ed.
	Folly Consulting
	March 09, 2009

Using and Extending the NSIS Protocol Family
draft-ietf-nsis-ext-02.txt

Status of this Memo

Copyright Notice

Abstract

This document gives an overview of the Next Steps in Signaling (NSIS) framework and protocol suite created by the NSIS working group during the period 2001-2009 together with suggestions on how the industry can make use of the new protocols, and how the community can exploit the extensibility of both the framework and existing protocols to address future signaling needs.

Table of Contents

---

1. Introduction and History

TOC

The Transport Area Directors held a Next Steps in Signaling (NSIS) birds of a feather session on Wednesday 21st March 2001 at the 50th IETF meeting in Minneapolis. The goal of the session was to discuss and gather an initial set of requirements for a next generation Internet signaling protocol suite as it was felt that the current RSVP-based solutions have short-comings, e.g., with respect to mobility or QoS interoperability. The NSIS Working Group was officially formed later that year, in November 2001 and had its first meeting at the IETF 52 in Salt Lake City in December 2001.

The initial charter of NSIS was focused on QoS signaling as the first use case, taking the Resource ReSerVation Protocol (RSVP) as the background for the work. In May 2003, middlebox traversal was added as an explicit second use case. The requirements for the new generation of signaling protocols are documented in [RFC3726] (Brunner, M., “Requirements for Signaling Protocols,” April 2004.) and an analysis of existing signaling protocols can be found in [RFC4094] (Manner, J. and X. Fu, “Analysis of Existing Quality-of-Service Signaling Protocols,” May 2005.).

The design of NSIS is based on a two-layer model, where a general signaling transport layer provides services to an upper signaling layer. The design was influenced by Bob Braden's Internet Draft entitled "A Two-Level Architecture for Internet Signaling" [I-D.braden-2level-signal-arch] (Braden, R. and B. Lindell, “A Two-Level Architecture for Internet Signaling,” November 2002.).

This document gives an overview of what the NSIS framework and protocol suite is at the time of writing (2008), provides help and guidelines to the reader as to how NSIS can be used in an IP network, and how the protocol suite can be enhanced to satisfy new use cases.

2. The NSIS Architecture

TOC

The design of the NSIS protocol suite reuses ideas and concepts from RSVP but essentially divides the functionality into two layers. The lower layer, the NSIS Transport Layer Protocol (NTLP), is in charge of transporting the higher layer protocol messages to the next signaling node on the path. This includes discovery of the next hop NSIS node, which may not be the next routing hop, and different transport and security services depending on the signaling application requirements. The General Internet Signaling Transport (GIST) [I-D.ietf-nsis-ntlp] (Schulzrinne, H. and M. Stiemerling, “GIST: General Internet Signalling Transport,” June 2009.) has been developed as the protocol that fulfills the role of the NTLP. The NSIS suite supports both IP protocol versions, IPv4 and IPv6.

The actual signaling application logic is implemented in the higher layer of the NSIS stack, the NSIS Signaling Layer Protocol (NSLP). While GIST is only concerned with transporting NSLP messages hop-by-hop between pairs of signalling nodes, the end-to-end signaling functionality is provided by the NSLP protocols if needed - not all NSLP protocols need to perform end-to-end signaling, even the current protocols have features to confine the signaling to a limited path. Messages transmitted by GIST on behalf of an NSLP are identified by a unique NSLP identifier (NSLPID) associated with the NSLP. Two NSLP protocols are currently standardized: one concerning Quality of Service signaling and one to enable NAT/Firewall traversal.

NSIS is primarily designed to provide the signaling needed to install state on nodes that lie on the path that will be taken by some end-to-end flow of data packets in order to facilitate or enhance some characteristic of the data flow. This is achieved by routing signaling messages along the same path (known as "path-coupled signaling") and intercepting the signaling message at NSIS capable nodes. Parameters carried by the signaling message drive the operation of the relevant transport or signaling application. In particular, the messages will carry Message Routing Information (MRI) that will allow the NSIS nodes to identify the data flow to which the signaling applies. Generally, the intercepted messages will be reinjected into the network after processing by the NSIS entities and routed further towards the destination, possibly being intercepted by additional NSIS nodes before arriving at the flow endpoint.

As with RSVP, it is expected that the signaling message will make a complete round trip either along the whole end-to-end path or a part of it if the scope of the signaling is limited. This implements a two-phase strategy in which capabilities are assessed and provisional reservations are made on the outbound leg; these provisional reservations are then confirmed and operational state installed on the return leg. Unlike RSVP, signaling is normally initiated at the source of the data flow making it easier to ensure that the signaling follows the expected path of the data flow, but can also be receiver initiated as in RSVP.

A central concept of NSIS is the Session Identifier (SID). Signaling application states are indexed and referred to through the SID within the NSLP layer. This decouples the state information from IP addresses, allowing dynamic IP address changes for signaling flows, e.g., due to mobility: changes in IP addresses do not force complete tear down and re-initiation of a signaling application state, merely an update of the state parameters in the NSLP(s), especially the MRI.

At the NTLP (GIST) layer the SID is not meaningful by itself, but is rather used together with the NSLP identifier (NSLPID) and the Message Routing Information (MRI). This 3-tuple is used by GIST to index and manage the signaling flows. Changes of routing or dynamic IP address changes, e.g., due to mobility, will require GIST to modify the Messaging Associations (MAs) that are used to channel NSLP messages

between adjacent GIST peers in order to satisfy the NSLP MRI for each SID.
The following design restrictions were imposed for the first phase of the protocol suite. They may be lifted in future and new functionality may be added into the protocols at some later stage.

*Initial focus on MRMs for path-coupled signaling: GIST transports messages towards an identified unicast data flow destination based on the signaling application request, and does not directly support path-decoupled signaling, e.g., QoS signaling to a bandwidth broker or other off-path resource manager. The framework also supports a "Loose-End" message routing method used to discover GIST nodes with particular properties in the direction of a given address, for example the NAT/FW NSLP uses this method to discover a NAT along the upstream data path.

*No multicast support: Introducing support for multicast was deemed too much overhead, considering the currently limited support for global IP multicast. Thus, the current GIST and the NSLP specifications consider unicast flows only.

The key documents specifying the NSIS framework are:

*Requirements for Signaling Protocols [RFC3726] (Brunner, M., “Requirements for Signaling Protocols,” April 2004.)

*Next Steps in Signaling: Framework [RFC4080] (Hancock, R., Karagiannis, G., Loughney, J., and S. Van den Bosch, “Next Steps in Signaling (NSIS): Framework,” June 2005.)

*Security Threats for NSIS [RFC4081] (Tschofenig, H. and D. Kroeselberg, “Security Threats for Next Steps in Signaling (NSIS),” June 2005.)

The protocols making up the suite specified by the NSIS working group are documented in:

*The General Internet Signaling Transport protocol [I-D.ietf-nsis-ntlp] (Schulzrinne, H. and M. Stiemerling, “GIST: General Internet Signalling Transport,” June 2009.)

*Quality of Service NSLP (QoS NSLP) [I-D.ietf-nsis-qos-nslp] (Manner, J., Karagiannis, G., and A. McDonald, “NSLP for Quality-of-Service Signaling,” January 2010.)

*The QoS specification template [I-D.ietf-nsis-qspec] (Ash, G., Bader, A., Kappler, C., and D. Oran, “QoS NSLP QSPEC Template,” November 2008.)

- NAT/Firewall traversal NSLP [I-D.ietf-nsis-nslp-natfw] (Stiemerling, M., Tschofenig, H., Aoun, C., and E. Davies, “NAT/Firewall NSIS Signaling Layer Protocol (NSLP),” April 2010.)

The next three sections provide a brief survey of GIST, the QoS NSLP, and the NAT/Firewall NSLP.

3. The General Internet Signaling Transport

TOC

The General Internet Signaling Transport (GIST) [I-D.ietf-nsis-ntlp] (Schulzrinne, H. and M. Stiemerling, “GIST: General Internet Signalling Transport,” June 2009.) provides a signaling transport and security services to NSIS Signaling Layer Protocols (NSLP) and the associated signaling applications. GIST does not define new IP transport protocols or security mechanisms but rather makes use of existing protocols, such as TCP, UDP, TLS and IPsec. Applications can indicate the desired reliability, e.g., unreliable or reliable, and GIST then uses the most appropriate transport protocol to achieve the goal. If applications request also security, GIST uses TLS. The GIST layered protocol stack is shown in Figure 1 (The NSIS protocol stack).

```
            +-----+ +--------+ +-------+
            |     | |        | |       |
            | QoS | | NAT/FW | |  ...  |      NSLP
            |     | |        | |       |
            +-----+ +--------+ +-------+

-------------------------------------------------------------------
            +--------------------------+
            |                          |
            |          GIST            |      NTLP
            |                          |
            +--------------------------+

-------------------------------------------------------------------
            +--------------------------+
            |           TLS            |
            +--------------------------+
            +--------------------------+
            | TCP | UDP | SCTP | DCCP  |
            +--------------------------+
            +--------------------------+
            |         IPsec            |
            +--------------------------+
            +--------------------------+
            |    IPv4     |    IPv6    |
            +--------------------------+
```

Figure 1: The NSIS protocol stack

GIST divides up the end-to-end path to be taken by the data flow into a number of segments between pairs of NSIS aware peer nodes located along the path. Not every router or other middlebox on the path needs to be NSIS aware: each segment of the signaling path may incorporate several routing hops. Also not every NSIS aware node necessarily implements every possible signaling application. If the signaling for a flow requests services from a subset of the applications, then only nodes that implement those services are expected to participate as peers, and even some of these nodes can decline to operate on a particular flow if, for example, the additional load might overload the processing capability of the node. These characteristics mean that incremental deployment of NSIS capabilities is possible both with the initial protocol suite, and for any future NSLP applications that might be developed. The following paragraphs describe how a signaling segment is setup offering the transport and security characteristics needed by a single NSLP.

When an NSLP application wants to send a message to its next peer, GIST starts the process of discovering the next signaling node by sending a Query message towards the destination of the related data flow. This Query carries the NSLP identifier (NSLPID) and Message Routing Information (MRI) among others. The MRI contains enough information to control the routing of the signaling message and identify the associated data flow. The next GIST node on the path receives the message and if it is running the same NSLP, it provides the MRI to the NSLP application and requests it to make a decision on whether to peer with the querying node. If the NSLP application chooses to peer, GIST sets up a Message Routing State (MRS) between the two nodes for the future exchange of NSLP data. State setup is performed by a three-way handshake that allows for negotiation of signaling flow parameters and provides counter-measures against several attacks like denial-of-service by using cookie mechanisms and a late state installation option.

If a transport connection is required and needs to provide for reliable or secure signaling, like TCP or TLS/TCP, a Messaging Association (MA) is established between the two peers. An MA can be re-used for signaling messages concerning several different data flows, i.e., signaling messages between two nodes are multiplexed over the same transport connection. This can be done when the transport requirements (reliability, security) of a new flow can be met with an existing MA, i.e., the security and transport properties of an existing MA are equivalent or better than what is requested by the new MA.

For path-coupled signaling, we need to find the nodes on the data path that should take part in the signaling of an NSLP and invoke them to act on the arrival of such NSLP signaling messages. The basic concept is that such nodes along a flow's data path intercept the corresponding signaling packets and are thus discovered automatically. It was originally envisaged that GIST would place a Router Alert Option (RAO) in Query message packets to ensure that they are intercepted by NSIS aware nodes as in RSVP.

Late in the development of GIST serious concerns were raised in the IETF about the security risks and performance implications of extensive usage of the RAO [I-D.rahman-rtg-router-alert-dangerous] (Rahman, R. and D. Ward, “Use of IP Router Alert Considered Dangerous,” October 2008.), as well as discovery of evidence that several existing implementations of RAO were inconsistent with the standards and would not support the NSIS usage. There were also concerns that extending the need for RAO recognition in the fast path of routers that are frequently implemented in hardware would delay or deter implementation and deployment of NSIS. An alternative mechanism was therefore standardized.

The approved version of GIST specifies that NSIS nodes recognise UDP packets directed to a specific destination port and containing a GIST specific "magic number" as the first 32 bits of the UDP payload as

Query messages that need to be intercepted. It is recognised that this interception method is not the most efficient possible and GIST provides for the use of alternatives, such as the RAO, for specific NSLPs as a part of its extensibility design. Further intentional bypassing of signaling nodes can be accomplished either in GIST or in the NSLP.

Since GIST carries information about the data flow inside its messages (in the MRI), NAT gateways must be aware of GIST in order to let it work correctly. GIST provides a special object for NAT traversal so that the actual translation is disclosed if a GIST-aware NAT gateway provides this object.

As with RSVP, all the state installed by NSIS protocols is "soft-state" that will expire and be automatically removed unless it is periodically refreshed. NSIS state is held both at the signaling application layer and in the transport layer, and is maintained separately. NSLPs control the lifetime of the state in the application layer by setting a timeout and sending periodic "keep alive" messages along the signaling path if no other messages are required. The MAs and the routing state are maintained semi-independently by the transport layer, because MAs may be used by multiple NSLP sessions, and can also be recreated "on demand" if the node needs to reclaim resources. The transport layer can send its own "keep alive" messages across a MA if no NSLP messages have been sent, for example if the transport layer decides to maintain a heavily used MA even though there is no current NSLP session using it. State can also be deleted explicitly when no longer needed.

If there is a change in the route used by a flow for which NSIS has created state, NSIS needs to detect the change in order to determine if the new path contains additional NSIS nodes that should have state installed. GIST may use a range of triggers in order to detect a route change. It probes periodically for the next peer by sending a GIST Query, thereby detecting a changed route and GIST peer. GIST monitors routing tables, the GIST peer states, and notifies NSLPs of any routing changes. It is then up to the NSLPs to act appropriately, if needed, e.g., by issuing a refresh message. The periodic queries also serve to maintain the soft-state in nodes as long as the route is unchanged.

In summary, GIST provides several services in one package to the upper layer signaling protocols:

- Signaling peer discovery: GIST is able to find the next hop node that runs the NSLP being signaled for.

- Multiplexing: GIST reuses already established signaling relationships and messaging associations to peers if the signaling flows traverse the same next signaling hop.

- Transport: GIST provides transport with different attributes, namely reliable/unreliable and secure/unsecure.

- *Confidentiality: If security is requested, GIST uses TLS to provide an encrypted and integrity protected message transport to the next signaling peer.

- *Routing changes: GIST detects routing changes, but instead of acting on its own, it merely sends a notification to the local NSLP. It is then up to the NSLP to act.

- *Fragmentation: GIST uses either a known Path MTU for the next hop or limits its message size to 576 bytes. If fragmentation is required it automatically establishes an MA and sends the signaling traffic over a reliable protocol, e.g., TCP.

- *State maintenance: GIST establishes and then maintains the soft-state that controls communications through MAs between GIST peers along the signaling path, according to usage parameters supplied by NSLPs and local policies.

4. Quality of Service NSLP

TOC

The Quality of Service (QoS) NSIS Signaling Layer Protocol (NSLP) establishes and maintains state at nodes along the path of a data flow for the purpose of providing some forwarding resources for that flow. It is intended to satisfy the QoS-related requirements of RFC 3726 [RFC3726] (Brunner, M., “Requirements for Signaling Protocols,” April 2004.). No support for QoS architectures based on bandwidth brokers or other off-path resource managers is currently included. The design of the QoS NSLP is conceptually similar to RSVP, RFC 2205 [RFC2205] (Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, “Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification,” September 1997.), and uses soft-state peer-to-peer refresh messages as the primary state management mechanism (i.e., state installation/refresh is performed between pairs of adjacent NSLP nodes, rather than in an end-to-end fashion along the complete signaling path). The QoS NSLP extends the set of reservation mechanisms to meet the requirements of RFC 3726 [RFC3726] (Brunner, M., “Requirements for Signaling Protocols,” April 2004.), in particular support of sender or receiver-initiated reservations, as well as, a type of bi-directional reservation and support of reservations between arbitrary nodes, e.g., edge-to-edge, end-to-access, etc. On the other hand, there is currently no support for IP multicast.

A distinction is made between the operation of the signaling protocol and the information required for the operation of the Resource Management Function (RMF). RMF-related information is carried in the QSpec (QoS Specification) object in QoS NSLP messages. This is similar to the decoupling between RSVP and the IntServ architecture, RFC 1633

[RFC1633] (Braden, B., Clark, D., and S. Shenker, “Integrated Services in the Internet Architecture: an Overview,” June 1994.). The QSpec carries information on resources available, resources required, traffic descriptions and other information required by the RMF. A template for Qspec objects is defined in [I-D.ietf-nsis-qspec] (Ash, G., Bader, A., Kappler, C., and D. Oran, “QoS NSLP QSPEC Template,” November 2008.). This provides a number of basic parameter objects that can be used as a common language to specify components of concrete QoS models. The objects defined in [I-D.ietf-nsis-qspec] (Ash, G., Bader, A., Kappler, C., and D. Oran, “QoS NSLP QSPEC Template,” November 2008.) provide the building blocks for many existing QoS models such as those associated with RSVP and Differentiated Services. The extensibility of the template allows new QoS model specifications to extend the template language as necessary to support these specifications.

The QoS NSLP supports different QoS models, because it does not define the QoS mechanisms and RMF that have to be used in a domain. As long as a domain knows how to perform admission control for a given QSpec, QoS NSLP actually does not care how the specified constraints are enforced and met, e.g., by putting the related data flow in the topmost of four DiffServ classes, or by putting it into the third highest of twelve DiffServ classes. The particular QoS configuration used is up to the network provider of the domain. The QSpec can be seen as a common language to express QoS requirements between different domains and QoS models.

In short, the functionality of the QoS NSLP includes:

- Conveying resource requests for unicast flows
- Resource requests (QSpec) that are decoupled from the signaling protocol (QoS NSLP)
- Sender- and receiver-initiated reservations, as well as, bi-directional
- Soft-state and reduced refresh (keep-alive) signaling
- Session binding, session X can be valid only if session Y is too
- Message scoping, end-to-end, edge-to-edge or end-to-edge (proxy mode)
- Protection against message re-ordering and duplication
- Group tear, tearing down several session with a single message
- Support for re-routing, e.g., due to mobility
- Support for request priorities and preemption

- *Stateful and stateless nodes: stateless operation is particularly relevant in core networks where large amounts of QoS state could easily overwhelm a node

- *Reservation aggregation

The protocol also provides for a proxy mode to allow the QoS signaling to be implemented without needing all end hosts to be capable of handling NSIS signaling.

The QSpec Template supports situations where the QoS parameters need to be fine-grained, specifically targeted to an individual flow in one part of the network (typically the edge or access part) but might need to be more coarse-grained, where the flow is part of an aggregate (typically in the core of the network).

---

5. NAT/Firewall Traversal NSLP

TOC

The NAT/Firewall Traversal NSLP [I-D.ietf-nsis-nslp-natfw] (Stiemerling, M., Tschofenig, H., Aoun, C., and E. Davies, “NAT/Firewall NSIS Signaling Layer Protocol (NSLP),” April 2010.) lets end-hosts interact with NAT and firewall devices in the data path. Basically it allows for a dynamic configuration of NATs and/or firewalls along the data path in order to enable data flows to traverse these devices without being obstructed. For instance, firewall pinholes could be opened on demand by authorized hosts. Furthermore, it is possible to block unwanted incoming traffic on demand, e.g., if an end-host is under attack.

Configurations to be implemented in NAT and firewall devices signalled by the NAT/Firewall NSLP take the form of a (Pattern, Action) pair, where the pattern specifies a template for packet header fields to be matched. The device is then expected to apply the specified action to any passing packet that matches the template. Actions are currently limited to ALLOW (forward the packet) and DENY (drop the packet). The template specification allows for a greater range of packet fields to be matched than those allowed for in the GIST MRI.

Basically NAT/Firewall signaling starts at the data sender (NSIS Initiator) before any actual application data packets are sent. Signaling messages may pass several NAT/Firewall NSLP-aware middleboxes (NSIS Forwarder) on their way downstream and usually hit the receiver (being the NSIS Responder). A proxy mode is also available for cases where the NAT/Firewall NSLP is not fully supported along the complete data path. NAT/Firewall NSLP is based on a soft-state concept, i.e., the sender must periodically repeat its request in order to keep it active.

Additionally, the protocol also provides functions for receivers behind NATs. The receiver may request an external address that is reachable from outside. The reserved external address must, however, be communicated to the sender out-of-band by other means, e.g., by application level signaling. After this step the data sender may initiate a normal NAT/Firewall signaling in order to create firewall pinholes.

The protocol also provides for a proxy mode to allow the NAT/Firewall signaling to be implemented without needing all end hosts to be capable of handling NSIS signaling.

6. Deploying the Protocols

TOC

First of all, NSIS implementations must be available in at least some of the corresponding network nodes (i.e., routers, firewalls, or NAT gateways) and end-hosts. That means not only GIST support, but also the NSLPs and their respective control functions (such as a resource management function for QoS admission control etc.) must be implemented. NSIS is capable of incremental deployment and an initial deployment does not need to involve every node in a network domain. This is discussed further in Section 6.2 (Incremental Deployment and Workarounds).

Another important issue is that applications may need to be made NSIS-aware, thereby requiring some effort on the applications programmer's behalf. Alternatively, it may be possible to implement separate applications to control, e.g., the network QoS requests or firewall pinholes, without needing to update the actual applications that will take advantage of NSIS capabilities.

6.1. Obstacles

TOC

Although GIST is no longer dependent on RAO (there is known to be network equipment with broken implementations of the RAO deployed), if NSIS is to be deployed in routers with hardware-based forwarding engines it is not guaranteed that the hardware will be able to divert Query packets identified by a well-known UDP port into the slow path, which will make deployment of NSIS dependent on hardware replacement rather than software upgrade. However, the removal of dependence on RAO makes it more likely that NSIS Query packets can be forwarded through nodes that are not NSIS aware.

NAT gateways and firewalls may also hinder initial deployment of NSIS protocols as they may either filter signaling traffic or perform NSIS-unaware address translations.

6.2. Incremental Deployment and Workarounds

TOC

NSIS is specifically designed to be incrementally deployable. It is not required that all nodes on the signaling and data path are NSIS aware. To make any use of NSIS at least two nodes on the path need to be NSIS aware. However, it is not essential that the initiator and receiver of the data flow are NSIS aware. Both the QoS and NAT/Firewall NSLPs provide "proxy modes" in which nodes adjacent to the initiator and/or receiver can act as proxy signaling initiator or receiver. An initiator proxy can monitor traffic and, hopefully, detect when a data flow of a type needing NSIS support is being initiated. The proxies can act more or less transparently on behalf of the data flow initiator and/or receiver to set up the required NSIS state and maintain it while the data flow continues. This capability reduces the immediate need to modify all the data flow end points before NSIS is viable.

7. Security Features

TOC

Basic security functions are provided at the GIST layer, e.g., protection against some blind or denial-of-service attacks, but note that introduction of alternative MRMs may provide attack avenues that are not present with the current emphasis on the path-coupled MRM. Conceptually it is difficult to protect against on-path attacker and man-in-the-middle attacks when using path-coupled MRMs, because a basic functionality of GIST is to discover yet unknown signaling peers. Transport security can be requested by signaling applications and is realized by using TLS between signaling peers, i.e., authenticity and confidentiality of signaling messages can be assured between peers. GIST allows for mutual authentication of the signaling peers (using TLS means like certificates) and can verify the authenticated identity against a database of nodes authorized to take part in GIST signaling. It is, however, a matter of policy that the identity of peers is verified and accepted upon establishment of the secure TLS connection. While GIST is handling authentication of peer nodes, more fine grained authorization may be required in the NSLP protocols. There is currently an ongoing work to specify common authorization mechanisms to be used in NSLP protocols [I-D.manner-nsis-nslp-auth] (Manner, J., Stiemerling, M., Tschofenig, H., and R. Bless, “Authorization for NSIS Signaling Layer Protocols,” July 2008.), thus allowing, e.g., per-user and per-service authorization.

TOC

8. Extending the Protocols

This section discusses the ways that are available to extend the NSIS protocol suite. The Next Steps in Signaling (NSIS) Framework [RFC4080] (Hancock, R., Karagiannis, G., Loughney, J., and S. Van den Bosch, “Next Steps in Signaling (NSIS): Framework,” June 2005.) describes a two-layer framework for signaling on the Internet, comprising a generic transport layer with specific signaling layers to address particular applications running over this transport layer. The model is designed to be highly extensible so that it can be adapted for different signaling needs.

It is expected that additional signaling requirements will be identified in future. The two layer approach allows for NSLP signaling applications to be developed independently of the transport protocol. Further NSLPs can therefore be developed and deployed to meet these new needs using the same GIST infrastructure thereby providing a level of macro-extensibility. However, the GIST protocol and the two signaling applications have been designed so that additional capabilities can be incorporated into the design should additional requirements within the general scope of these protocols need to be accommodated.

The NSIS framework is also highly supportive of incremental deployment. A new NSLP need not be available on every NSIS aware node in a network or along a signaling path in order to start using it. Nodes that do not (yet) support the application will forward it without complaint until it reaches a node where the new NSLP application is deployed.

One key functionality of parameter objects carried in NSIS protocols is the so-called "Extensibility flags (AB)". All the existing protocols (and any future ones conforming to the standards) can carry new experimental objects, where the AB-flags can indicate whether a receiving node must interpret the object, or whether it can just drop them or pass them along in subsequent messages sent out further on the path. This functionality allows defining new objects without forcing all network entities to understand them.

8.1. Overview of Administrative Actions Needed When Extending NSIS

TOC

Generally, NSIS protocols can be extended in multiple ways, many of which require the allocation of unique code point values in registries maintained by IANA on behalf of the IETF. This section is an overview of the administrative mechanisms that might apply. The extensibility rules are based upon the procedures by which IANA assigns values: "Standards Action" (as defined in [IANA]), "IETF Action", "Expert Review", and "Organization/Vendor Private", defined below. The appropriate procedure for a particular type of code point is defined in one or other of the NSIS protocol documents, mostly

[I-D.ietf-nsis-ntlp] (Schulzrinne, H. and M. Stiemerling, “GIST: General Internet Signalling Transport,” June 2009.). In addition to registered code points, all NSIS protocols provide code points that can be used for experimentation, usually within closed networks, as explained in [RFC3692] (Narten, T., “Assigning Experimental and Testing Numbers Considered Useful,” January 2004.). There is no guarantee that independent experiments will not be using the same code point!

8.2. GIST

TOC

GIST is extensible in several aspects. In this list, references to document sections refer to the GIST specification [I-D.ietf-nsis-ntlp] (Schulzrinne, H. and M. Stiemerling, “GIST: General Internet Signalling Transport,” June 2009.).

*Use of different Message Routing Methods: currently only two message routing methods are supported (Path-coupled MRM and Loose-End MRM), but further MRMs may be defined in the future. See Section 3.8. One possible additional MRM under development is documented in [I-D.bless-nsis-est-mrm] (Bless, R., “An Explicit Signaling Target Message Routing Method (EST-MRM) for the General Internet Signaling Transport (GIST) Protocol,” July 2008.). This MRM would direct signaling towards an explicit target address other than the (current) data flow destination and is intended to assist setting up of state on a new path during 'make-before-break' handover sequences in mobile operations. Note that alternative routing methods may require modifications to the firewall traversal techniques used by GIST and NSLPs.

-New MRMs require allocation of a new MRM-ID either by standards action or expert review[I-D.ietf-nsis-ntlp] (Schulzrinne, H. and M. Stiemerling, “GIST: General Internet Signalling Transport,” June 2009.).

*Use of different transport protocols or security capabilities: the initial handshake allows a negotiation of the transport protocols to be used. Currently, a proposal to add DCCP and DTLS to GIST exists [I-D.manner-nsis-gist-dccp] (Manner, J., “Generic Internet Signaling Transport over DCCP and DTLS,” June 2007.). See Sections 3.2 and 5.7. GIST expects alternative capabilities to be treated as selection of an alternative protocol stack. Within the protocol stack, the individual protocols used are specified by MA Protocol IDs which are allocated from an IANA registry if new protocols are to be used. See Sections 5.7 and 9.

  - Use of an alternative transport protocol or security capability requires allocation of a new MA-Protocol-ID either by standards action or expert review[I-D.ietf-nsis-ntlp] (Schulzrinne, H. and M. Stiemerling, “GIST: General Internet Signalling Transport,” June 2009.).

*Use of alternative security services: Currently only TLS is specified for providing secure channels with MAs. Section 3.9 suggests that alternative protocols could be used, but the interactions with GIST functions would need to be carefully specified. See also Section 4.4.2.

  - Use of an alternative security service requires allocation of a new MA-Protocol-ID either by standards action or expert review[I-D.ietf-nsis-ntlp] (Schulzrinne, H. and M. Stiemerling, “GIST: General Internet Signalling Transport,” June 2009.).

*Query mode packet interception schemes: GIST has standardized a simple scheme using a well-known UDP port number plus a "magic number" at the start of the UDP payload. Alternative schemes, possibly including a reversion to the original proposal to use RAO mechanisms[I-D.hancock-nsis-gist-rao] (Hancock, R., “Using the Router Alert Option for Packet Interception in GIST,” November 2008.), can be specified as extensions. See Sections 5.3.2 and 5.3.2.5. Each NSLP needs to specify membership of an "interception class" whenever it sends a message through GIST. A packet interception scheme can support one or more interception classes. In principle, a GIST instance can support multiple packet interception schemes, but each interception class needs to be associated with exactly one interception scheme in a GIST instance and GIST instances that use different packet interception schemes for the same interception class will not be interoperable.

Defining an alternative interception class mechanism for incorporation into GIST should be considered as a very radical step and all alternatives should be considered before taking this path. The main reason for this is that the mechanism will necessarily require additional operations on every packet passing through the affected router interfaces. A number of considerations should be taken into account:

  - Although the interception mechanism need only be deployed on routers that actually need it (probably for a new NSLP), deployment may be constrained if the mechanism requires modification to the hardware of relevant routers and/or needs to await modification of the software by the router vendor.

- Any packet fields to be examined should be normally close to the start of the packet so that additional memory accesses are not needed to retrieve the values needed for examination.

- The logic required to determine if a packet should be intercepted needs to be kept simple to minimise the extra per-packet processing.

- The mechanism should be applicable to both IPv4 and IPv6 packets.

- Packet interception mechanisms potentially provide an attack path for Denial of Service attacks on routers, in that packets are diverted into the "slow path" and hence can significantly increase the load on the general processing capability of the router. Any new interception mechanism needs to be carefully designed to minimize the attack surface.

Packet interception mechanisms are identified by an "interception class" which is supplied to GIST through the Application Programming Interface for each message sent.

- New packet interception mechanisms will generally require allocation of one or more new Interception-class-IDs. This does not necessarily need to be placed in an IANA registry as it is primarily used as a parameter in the API between the NSLPs and GIST and may never appear on the wire, depending on the mechanism employed; all that is required is consistent interpretation between the NSLPs and GIST in each applicable node. However, if, as is the case in the RAO proposal [I-D.hancock-nsis-gist-rao] (Hancock, R., “Using the Router Alert Option for Packet Interception in GIST,” November 2008.), the scheme distinguishes between multiple packet interception classes by a value carried on the wire (different values of RAO parameter for the RAO proposal), an IANA registry may be required to provide a mapping between interception classes and on-the-wire values as discussed in Section 6 of [I-D.hancock-nsis-gist-rao] (Hancock, R., “Using the Router Alert Option for Packet Interception in GIST,” November 2008.).

*Use of alternative NAT traversal mechanisms: the mechanisms proposed for both legacy NAT traversal (Section 7.2.1) and GIST-aware NAT traversal (Section 7.2.2) can be extended or replaced. As discussed above, extension of NAT traversal may be needed if a new MRM is deployed. Note that there is extensive discussion of

NAT traversal in the NAT/Firewall NSLP specification [I-D.ietf-nsis-nslp-natfw] (Stiemerling, M., Tschofenig, H., Aoun, C., and E. Davies, “NAT/Firewall NSIS Signaling Layer Protocol (NSLP),” April 2010.).

- Additional error identifiers: Making extensions to any of the above items may result in new error modes having to be catered for. See Section 9 and Appendix A Sections A.4.1 - A.4.3.

  - Additional error identifiers require allocation of new error code(s) and/or subcode(s), and may also require allocation of Additional Information types. These are all allocated on a first-come, first-served basis by IANA [I-D.ietf-nsis-ntlp] (Schulzrinne, H. and M. Stiemerling, “GIST: General Internet Signalling Transport,” June 2009.).

- Generally: the AB-flags enable the community to specify new objects applicable to GIST, that can be carried inside a signaling session without breaking existing implementations. The AB-flags can also be used to indicate in a controlled fashion that a certain object must be understood by all GIST nodes, which makes it possible to probe for the support of an extension. One such object already designed is the "Peering Information Object (PIO)" [I-D.manner-nsis-peering-data] (Manner, J., Liuhto, L., Varis, N., and T. Huovila, “Peering Data for NSIS Signaling Layer Protocols,” February 2008.) that allows a QUERY message to carry additional peering data for the recipient for making the peering decision.

  - New objects require allocation of a new Object Type ID either by standards action or provision of another type of specification [I-D.ietf-nsis-ntlp] (Schulzrinne, H. and M. Stiemerling, “GIST: General Internet Signalling Transport,” June 2009.).

- Major modifications could be made by adding additional GIST message types and defining appropriate processing. It might be necessary to define this as a new version of the protocol. A field is provided in the GIST Common Header containing the version number. GIST currently has no provision for version or capability negotiation that might be needed if a new version was defined.

  - New GIST Message Types require allocation of a new GIST Message Type ID either by standards action or expert review [I-D.ietf-nsis-ntlp] (Schulzrinne, H. and M. Stiemerling, “GIST: General Internet Signalling Transport,” June 2009.).

Finally, and more generally, as asserted in Section 1 of the GIST specification, the GIST design could be extended to cater for multicast flows and for situations where the signaling is not tied to an end-to-end data flow, but it is not clear whether this could be done in a totally backwards compatible way, and is not considered within the extensibility model of NSIS.

8.3. QoS NSLP

TOC

The QoS NSLP provides signaling for QoS reservations on the Internet. The QoS NSLP decouples the resource reservation model or architecture (QoS model) from the signaling. The signaling protocol is defined in Quality of Service NSLP (QoS NSLP) [I-D.ietf-nsis-qos-nslp] (Manner, J., Karagiannis, G., and A. McDonald, “NSLP for Quality-of-Service Signaling,” January 2010.). The QoS models are defined in separate specifications and the QoS NSLP can operate with one or more of these models as required by the environment where it is used. It is anticipated that additional QoS models will be developed to address various Internet scenarios in the future. Extensibility of QoS models is considered in Section 8.4 (QoS Specifications).
The QoS NSLP specifically mentions the possibility of using alternative Message Routing Methods (MRMs), apart from the general ability to extend NSLPs using new objects with the standard "AB" extensibility flags to allow them to be used in new and old implementations.
There is already work to extend the base QoS NSLP and GIST to enable new QoS signaling scenarios. One such proposal is the Inter-Domain Reservation Aggregation aiming to support large-scale deployment of the QoS NSLP [I-D.bless-nsis-resv-aggr] (Doll, M. and R. Bless, “Inter-Domain Reservation Aggregation for QoS NSLP,” July 2007.). Another current proposal seeks to extend the whole NSIS framework towards path-decoupled signaling and QoS reservations [I-D.cordeiro-nsis-hypath] (Cordeiro, L., Curado, M., Monteiro, E., Bernardo, V., Palma, D., Racaru, F., Diaz, M., and C. Chassot, “GIST Extension for Hybrid On-path Off-path Signaling (HyPath),” February 2008.).

8.4. QoS Specifications

TOC

The QoS Specification template (QSpec) is defined in [I-D.ietf-nsis-qspec] (Ash, G., Bader, A., Kappler, C., and D. Oran, “QoS NSLP QSPEC Template,” November 2008.). This provides the language in which the requirements of specific QoS models are described.
Introduction of new QoS models requires IETF action, with the published document defining the specific elements within the QSpec used in the

new model. See [I-D.ietf-nsis-qspec] (Ash, G., Bader, A., Kappler, C., and D. Oran, “QoS NSLP QSPEC Template,” November 2008.) for details. The introduction of new QoS models is designed to enable deployment of NSIS-based QoS control in specific scenarios. One such example is the Integrated Services Controlled Load Service for NSIS [I-D.kappler-nsis-qosmodel-controlledload] (Kappler, C., Fu, X., and B. Schloer, “A QoS Model for Signaling IntServ Controlled-Load Service with NSIS,” April 2010.).

A key feature provided by defining the QSpec template is support of a common language for describing QoS requirements and capabilities, which can be reused by any QoS models intending to use the QoS NSLP to signal their requirements for traffic flows. The commonality of the QSpec parameters ensures a certain level of interoperability of QoS models and reduces the demands on hardware that has to implement the QoS control. Optional QSpec parameters support the extensibility of the QoS NSLP to other QoS models in the future; new QSpec parameters can be defined in the document that defines a new QoS model. See Sections 4.4 and 7 of [I-D.ietf-nsis-qspec] (Ash, G., Bader, A., Kappler, C., and D. Oran, “QoS NSLP QSPEC Template,” November 2008.).

The QSPEC consists of a QSPEC version number, QSPEC objects plus specification of processing and procedures that can be used to build many QoS models. If changes are made to the QSPEC that are not backwards compatible, a new QSPEC version number has to be assigned. Note that a new QSPEC version number is not needed just because new additional QSPEC parameters are specified; new versions will be needed only if the existing functionality is modified. It is required that later QSPEC versions be backward compatible with earlier QSPEC versions. The template includes version negotiation procedures that allow the originator of an NSLP message to retry with a lower QSPEC version if the receiver rejects a message because it does not support the QSPEC version signaled in the message.

*Creation of a new, incompatible version of an existing Qspec requires allocation of a new QSPEC version number by standards action. See [I-D.ietf-nsis-qspec] (Ash, G., Bader, A., Kappler, C., and D. Oran, “QoS NSLP QSPEC Template,” November 2008.).

*Completely new QSPECs can also be created. Such new QSPECs require allocation of a QSPEC type by standards action. Values are also available for local or experimental use during development. See [I-D.ietf-nsis-qspec] (Ash, G., Bader, A., Kappler, C., and D. Oran, “QoS NSLP QSPEC Template,” November 2008.).

*Additional QSPEC procedures can be defined requiring allocation of a new QSPEC procedure number by standards action or through a another specification document. Values are also available for local or experimental use during development. See

[I-D.ietf-nsis-qspec] (Ash, G., Bader, A., Kappler, C., and D. Oran, “QoS NSLP QSPEC Template,” November 2008.).

- Additional QSPEC parameters and associated error codes can be defined requiring a specification document. Values are also available for local or experimental use during development. See [I-D.ietf-nsis-qspec] (Ash, G., Bader, A., Kappler, C., and D. Oran, “QoS NSLP QSPEC Template,” November 2008.).

8.5. NAT/Firewall NSLP

TOC

The NAT/Firewall signaling can be extended broadly in the same way as the QoS NSLP by defining new parameters to be carried in NAT/Firewall NSLP messages. See Section 7 of [I-D.ietf-nsis-nslp-natfw] (Stiemerling, M., Tschofenig, H., Aoun, C., and E. Davies, “NAT/Firewall NSIS Signaling Layer Protocol (NSLP),” April 2010.). No proposals currently exist to fulfill new use cases for the protocol.

8.6. New NSLP protocols

TOC

Designing a new NSLP is both challenging and easy.
New signaling applications with associated NSLPs can be defined to work in parallel or replace the applications already defined by the NSIS working group. Applications that fit into the NSIS framework will be expected to use GIST to provide transport of signaling messages and appropriate security facilities which relieves the application designer of many "lower level" problems. GIST provides many important functions through its service layer API, and allows the signaling application programmer to offload, e.g., the channel security, transport characteristics and signaling node discovery to GIST.
Yet, on the other hand, the signaling application designer must take into account that the network environment can be dynamic, both in terms of routing and node availability. The new NSLP designer must take into account at least the following issues:

- Routing changes, e.g., due to mobility: GIST sends Network Notifications when something happens in the network, e.g., peers or routing paths change. All signaling applications must be able to handle these notifications and act appropriately. GIST does not include logic to figure out what the NSLP would want to do due to a certain network event. Therefore, GIST gives the notification to the application, and lets it make the right decision.

- GIST indications: GIST will also send other notifications, e.g., if a signaling peer does not reply to refresh messages, or a certain NSLP message was not successfully delivered to the recipient. Again, NSLP applications must be able to handle these events, too. Appendix B in the GIST specification discusses the GIST-NSLP API and the various functionality required, but implementing this interface can be quite challenging; the multitude of asynchronous notifications than can from GIST increases the implementation complexity of the NSLP.

- Lifetime of the signaling flow: NSLPs should inform GIST when a flow is no longer needed using the SetStateLifetime primitive. This reduces bandwidth demands in the network.

- NSLP IDs: NSLP messages may be multiplexed over GIST MAs. The new NSLP needs to use a unique NSLPID to ensure that its messages are delivered to the correct application by GIST. A single NSLP could use multiple NSLPIDs, for example to distinguish different classes of signaling nodes that might handle different levels of aggregation of requests or alternative processing paths. Note that unlike GIST, the NSLPs do not provide a protocol versioning mechanism. If the new NSLP is an upgraded version of an existing NSLP, then it should be distinguished by a different NSLPID.

  - A new generally available NSLP requires IESG approval for the allocation of a new NSLP ID [I-D.ietf-nsis-ntlp] (Schulzrinne, H. and M. Stiemerling, “GIST: General Internet Signalling Transport,” June 2009.).

- Source IP address: It is sometimes challenging to find out at the NSLP, what will the source IP address be, especially when a node has multiple interfaces. Moreover, the logic in specifying the source IP address may differ if the node processing an NSLP message is the source of the signaling flow, or an intermediate node on the signaling. Thus, the NSLP must be able to find out the right source IP address from its internal interfaces, and its location on the signaling.

- New MRMs: GIST defines currently two Message Routing Methods, and leave the door open for new ideas. Thus, it is possible that a new NSLP also requires a new MRM, path-decoupled routing being one example.

- Cooperation with other NSLPs: Some applications might need resources from two or more different classes in order to operate successfully. The NSLPs managing these resources could operate cooperatively to ensure that such requests were coordinated to avoid wasting signaling bandwidth and prevent race conditions.

The API between GIST and NSLPs (see Appendix B in [I-D.ietf-nsis-ntlp] (Schulzrinne, H. and M. Stiemerling, “GIST: General Internet Signalling Transport,” June 2009.)) is very important to understand. The abstract design in the GIST specification does not specify the exact messaging between GIST and the NSLPs but gives an understanding of the interactions, especially what kinds of asynchronous notifications from GIST the NSLP must be prepared to handle: the actual interface will be dependent on each implementation of GIST.

Messages transmitted by GIST on behalf of an NSLP are identified by a unique NSLP identifier (NSLPID). NSLPIDs are 16 bit unsigned numbers taken from a registry managed by IANA and defined in Section 9 of the GIST specification [I-D.ietf-nsis-ntlp] (Schulzrinne, H. and M. Stiemerling, “GIST: General Internet Signalling Transport,” June 2009.).

A range of values (32704-32767) is available for Private and Experimental use during development, but any new signaling application that expects to be deployed generally on the Internet needs to be defined either in a standards track RFC or, possibly, an experimental RFC. Such an RFC would request allocation of unique NSLPID value(s) from the IANA registry. There is additional discussion of NSLPIDs in Section 3.8 of the GIST specification.

9. Security Considerations

TOC

This document provides information to the community. It does not raise new security concerns.

10. IANA Considerations

TOC

This memo includes no request to IANA.

11. Acknowledgements

TOC

This document combines work previously published as two separate drafts: "What is Next Steps in Signaling anyway - A User's Guide to the NSIS Protocol Family" written by Roland Bless and Jukka Manner, and "NSIS Extensibility Model" written by John Loughney.

Max Laier, Nuutti Varis and Lauri Liuhto have provided reviews of "User's Guide" draft and valuable input.

The "Extensibility Model" borrowed some ideas and some text from RFC3936 (Kompella, K. and J. Lang, “Procedures for Modifying the

Resource reSerVation Protocol (RSVP),” October 2004.) [RFC3936], Procedures for Modifying the Resource ReSerVation Protocol (RSVP); Robert Hancock provided text for the original GIST section, since much modified and Claudia Keppler have provided feedback on this draft, while Allison Mankin and Bob Braden suggested that this draft be worked on.

12. References

TOC

12.1. Normative References

TOC

[I-D.ietf-nsis-nslp-natfw]	Stiemerling, M., Tschofenig, H., Aoun, C., and E. Davies, “NAT/Firewall NSIS Signaling Layer Protocol (NSLP),” draft-ietf-nsis-nslp-natfw-25 (work in progress), April 2010 (TXT).
[I-D.ietf-nsis-ntlp]	Schulzrinne, H. and M. Stiemerling, “GIST: General Internet Signalling Transport,” draft-ietf-nsis-ntlp-20 (work in progress), June 2009 (TXT).
[I-D.ietf-nsis-qos-nslp]	Manner, J., Karagiannis, G., and A. McDonald, “NSLP for Quality-of-Service Signaling,” draft-ietf-nsis-qos-nslp-18 (work in progress), January 2010 (TXT).
[I-D.ietf-nsis-qspec]	Ash, G., Bader, A., Kappler, C., and D. Oran, “QoS NSLP QSPEC Template,” draft-ietf-nsis-qspec-21 (work in progress), November 2008 (TXT).
[RFC3726]	Brunner, M., “Requirements for Signaling Protocols,” RFC 3726, April 2004 (TXT).
[RFC4080]	Hancock, R., Karagiannis, G., Loughney, J., and S. Van den Bosch, “Next Steps in Signaling (NSIS): Framework,” RFC 4080, June 2005 (TXT).
[RFC4081]	Tschofenig, H. and D. Kroeselberg, “Security Threats for Next Steps in Signaling (NSIS),” RFC 4081, June 2005 (TXT).

12.2. Informative References

TOC

[I-D.bless-nsis-est-mrm]	Bless, R., “An Explicit Signaling Target Message Routing Method (EST-MRM) for the General Internet Signaling Transport (GIST) Protocol,” draft-bless-nsis-est-mrm-01 (work in progress), July 2008 (TXT).

[I-D.bless-nsis-resv-aggr]	Doll, M. and R. Bless, “Inter-Domain Reservation Aggregation for QoS NSLP,” draft-bless-nsis-resv-aggr-01 (work in progress), July 2007 (TXT).
[I-D.braden-2level-signal-arch]	Braden, R. and B. Lindell, “A Two-Level Architecture for Internet Signaling,” draft-braden-2level-signal-arch-01 (work in progress), November 2002 (TXT).
[I-D.cordeiro-nsis-hypath]	Cordeiro, L., Curado, M., Monteiro, E., Bernardo, V., Palma, D., Racaru, F., Diaz, M., and C. Chassot, “GIST Extension for Hybrid On-path Off-path Signaling (HyPath),” draft-cordeiro-nsis-hypath-05 (work in progress), February 2008 (TXT).
[I-D.hancock-nsis-gist-rao]	Hancock, R., “Using the Router Alert Option for Packet Interception in GIST,” draft-hancock-nsis-gist-rao-00 (work in progress), November 2008 (TXT).
[I-D.kappler-nsis-qosmodel-controlledload]	Kappler, C., Fu, X., and B. Schloer, “A QoS Model for Signaling IntServ Controlled-Load Service with NSIS,” draft-kappler-nsis-qosmodel-controlledload-11 (work in progress), April 2010 (TXT).
[I-D.manner-nsis-gist-dccp]	Manner, J., “Generic Internet Signaling Transport over DCCP and DTLS,” draft-manner-nsis-gist-dccp-00 (work in progress), June 2007 (TXT).
[I-D.manner-nsis-nslp-auth]	Manner, J., Stiemerling, M., Tschofenig, H., and R. Bless, “Authorization for NSIS Signaling Layer Protocols,” draft-manner-nsis-nslp-auth-04 (work in progress), July 2008 (TXT).
[I-D.manner-nsis-peering-data]	Manner, J., Liuhto, L., Varis, N., and T. Huovila, “Peering Data for NSIS Signaling Layer Protocols,” draft-manner-nsis-peering-data-01 (work in progress), February 2008 (TXT).
[I-D.rahman-rtg-router-alert-dangerous]	Rahman, R. and D. Ward, “Use of IP Router Alert Considered Dangerous,” draft-rahman-rtg-router-alert-dangerous-00 (work in progress), October 2008 (TXT).
[RFC1633]	Braden, B., Clark, D., and S. Shenker, “Integrated Services in the Internet Architecture: an Overview,” RFC 1633, June 1994 (TXT, PS, PDF).
[RFC2205]	Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, “Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification,” RFC 2205, September 1997 (TXT, HTML, XML).
[RFC3692]	

Narten, T., “Assigning Experimental and Testing Numbers Considered Useful,” BCP 82, RFC 3692, January 2004 (TXT).

[RFC3936] Kompella, K. and J. Lang, “Procedures for Modifying the Resource reSerVation Protocol (RSVP),” BCP 96, RFC 3936, October 2004 (TXT).

[RFC4094] Manner, J. and X. Fu, “Analysis of Existing Quality-of-Service Signaling Protocols,” RFC 4094, May 2005 (TXT).

Authors' Addresses

TOC

Jukka Manner
Helsinki University of Technology (TKK)
P.O. Box 3000
Espoo FIN-02015 TKK
Finland
Phone: +358 9 451 2481
Email: jukka.manner@tkk.fi
URI: http://www.netlab.tkk.fi/~jmanner/

Roland Bless
Institute of Telematics, Universitaet Karlsruhe (TH)
Zirkel 2
Karlsruhe 76128
Germany
Phone: +49 721 608 6413
Email: bless@tm.uka.de
URI: http://www.tm.uka.de/~bless

John Loughney
Nokia
955 Page Mill Road
Palo Alto 94303
USA
Phone: +1 650 283 8068
Email: john.loughney@nokia.com

Elwyn Davies (editor)
Folly Consulting
Soham,
UK
Phone:
Fax:
Email: elwynd@folly.org.uk

URI:	http://www.folly.org.uk