NSIS
Internet-Draft
Expires: September 7, 2006

X. Fu
I. Juchem
C. Dickmann
Univ. Goettingen
H. Tschofenig
Siemens

March 6, 2006

Design Options of NSIS Diagnostics NSLP
draft-fu-nsis-diagnostics-nslp-01

Status of this Memo

Copyright Notice

Abstract

The Next Steps in Signaling protocol suite aims to provide a way to communicate with network intermediaries. As such, it is desirable to offer generic diagnostics function for NSIS users and system administrators to make the functionality provided by the network more transparent (e.g., to identify particular NSLPs, to determine to

which degree the network supports NSIS, GIST state or specific NSLP session information along a given path).

Instead of suggesting one specific solution we highlight the different design options of some simple, stateless diagnostics functions from a querying node to a responding node. These preliminary thoughts should help the working group to have a more structure discussion in this problem space.

Table of Contents

1. Introduction

The lack of a unified mechanism for diagnostics capabilities in the NSIS protocol suite represents a substantial problem, for both the end user and the system administrator. As the number of vendors and operators deploying (and possibly extending) NSIS is expected to increase, the problem of diagnosing the multitude of devices from different signaling functions, both for general signaling transport and for particular signaling applications, escalates. Although some NSLP specifications set out the details of how a device can enquire some sort of diagnostics data, the extent to which this diagnostics data is used and converted to meaningful information by the specific NSLPs varies considerably from one NSLP to another. For instance, in the current version of QoS-NSLP specification [1], Query messages are used for detecting path characteristics information from any QNE towards a QNI. In the case of NATFW NSLP, current specification [2] defines a Query and Diagnostics Request message used by authorized NATFW NEs for querying and diagnosing installed NATFW states and it is still in discussion whether to exclude this feature from the base specification of the NATFW NSLP. On the other hand, GIST [3] does not provide an explicit diagnostics function to allow authorized entities to diagnose the GIST level information; it simply discovers the next GIST peer and delivers NSLP messages as its payload.

It is expected that the additional administrative or development effort is required without diagnostic capabilities. RSVP's diagnostics functions (see [4]) allow any RSVP node to query a RSVP session's Path state and Resv state (if available). However they exhibit a lack of the the capability of performing authorized access, which may be desired to prevent the introduction of security vulnerabilities (see [5]).

This document describes some key design considerations towards a set of simple, generic and secure diagnostics functions. The following aspects seem to be major design decisions:

- Which information is going to be subject for diagnosis?

- o At which granularity of the information can be diagnosed?
- o Which transport mechanism should be used for delivering diagnostics messages, and whether to introduce/avoid GIST level and/or NSLP state
- o How to properly authorize a diagnostics operation and protect the diagnostics messages?

Subsequent sections will discuss these aspects in more details.

2. Information to be Diagnosed

The main purpose of diagnostics NSLP is to diagnose NSIS state information. One question is which state could and should be diagnosed, and how much granularity of the state should be possible to be diagnosed. Corresponding to the 2-layer architecture of NSIS, there are two types of NSIS states: GIST and NSLP state information. In case of GIST (or NTLP) state, we also distinguish between state information we want to collect for states corresponding to one single Session ID (SID), further labeled "SID-bound state information", and state information to be collected which is not directly corresponding to a SID (further called "General GIST state information". We will list a few state information elements below:
General GIST state information:

GIST state includes:
MA information: The MA information might include a list of the message associations a GIST node has currently in use. This involves the properties of the MA, e.g. the used protocol stack, the address of the corresponding peer and the list of MRS using the MA. In addition a general information about the supported protocol stacks (in respect of the GIST stack proposal) should be included in the diagnostics data.

GIST MRS information: An exhaustive list of the MRS table might cause the size of a diagnostics messages to increase dramatically, for example, when the core node supporting tens of thousands of sessions. It may also be not very helpful and also not secure without being scoped to a limited network domain. However, the total numbers of MRSs over an individual

MAs in a GIST node may be of interest to the entity performing the diagnostics function.

SID-bound state information:

For state information which is directly corresponding to one single GIST Session ID/NSLPid tupel, more detailed information can be collected. This includes, but does not limit to, the following information:

MA information: In contrast to the general GIST state case, the MA information for the SID-bound state information is limited to the message associations related to the specific SessionID/ NSLPID tupel.

MRS information: The MRS data might include the MRI and information about the upstream and downstream peers, as well as configuration values, such as refresh intervals.

NSLP state:

It is likely possible to collect some information about the NSLP state corresponding to a particular session ID, if the authorization issue is addressed. However, it should not allow a diagnostics message from any querying node to query state belonging to other sessions or other NSLPs not being the present node (requestor). For NAT/FW NSLP, it may be interesting for a requestor to be able to diagnose the existence of NAT and FW devices (their IP addresses) and if possible, number or detailed information of NAT bindings or firewall entries, without a per-session basis. However, this is subject to authorization decisions in the protocol operation. Also, the QOS NSLP can be diagnosed for various information.

In addition, collecting general information such as GIST supporting information (e.g., GIST node IP addresses) or in the case of QoS NSLP, QOSM ID supporting information in the QoS NSLP supporting nodes may be possible.

Furthermore, if necessary, one can also calculate the processing delay information in GIST nodes, by putting timestamps to the

diagnostics messages when they traverse a GIST node. This simple extension, similar to traceroute, can be added to diagnostics messages as discussed in [6].

The final information we propose to be gathered from NSIS nodes is their support for different NSLPs. For a network administrator it will be helpful to collect which NSLPs are running on a certain node within the domain. This will help finding out whether an NSLP he is diagnosing is still/has been running on said peer. Also, it will be possible to detect still active NSLPs on nodes within the domain which should no longer be running for security reasons.

3. Information gathering and data transport options

This section will describe conditions and basic usage along with a proposed message format for GIST diagnostics client messages. It will also highlight practical usage for different scenarios.

3.1. Basic prerequisites

It is desirable that a diagnostics function does not install any new state. However sometimes this is inevitable, for example state in GIST level, especially MRS state even no new MAs will be introduced. It is possible that the diagnostics messages will be transported in a stateless mode and the response messages are directly addressed to

the requestor, if it is not necessary to maintain reverse routing information and modify/add results in the reverse direction. On the other hand, if a new MRS needs to be created in querying direction, then it should be removed in the response direction. Therefore, to avoid state creation on NTLP level, diagnostics messages should be kept as small as possible.

Diagnostics messages should be limited to fit into a D-mode (e.g., UDP) message in size, thus no larger than 64k and likely limited by the link MTU.

For simplicity and easy implementation we will continue NSIS' message object approach. This means that messages created by the diagnostic tool will consist of several objects which themselves could be compiled by adding various other message objects. By this, we also

enable easy extensibility for further information gathering of yet unknown NSLPs.

We will limit the diagnostic functions towards sysadmins diagnosing withing thei own domain only.

Messages by the diagnostics tool consist of one common header followed by a Query object and a list of Hop objects:

```
DIAGNOSTIC-message =
   Common header, Query object, [Hop object]*
```

The Query object specifies the requested information every GIST node should add. A Hop object is added by every GIST node on the path and contains the requested information. The Hop object itself consists of a Hop-Header and a list of data objects:

```
Hop-object =
   Hop header
   [IPv4 address object]*
   [IPv6 address object]*
   [General GIST information object]
   [SID-bound Response object]
   [NSLP state information object]
   [Available NSLPs object]
   [Additional information object]
```

The procedure for the diagnostic tools will be as follows:

1. At querying node, compose and send query message with designated destination - the final GIST node along the path
2. GIST at querying node forwards message to next GIST node towards the destination

3. Intermediate Diagnostic-NSLP-aware GIST nodes add queried information if message is on the downstream direction and forward to next peer
4. The destination GIST node adds queried information and forwards the message in the downstream direction

The message flow is depicted in Figure 1.

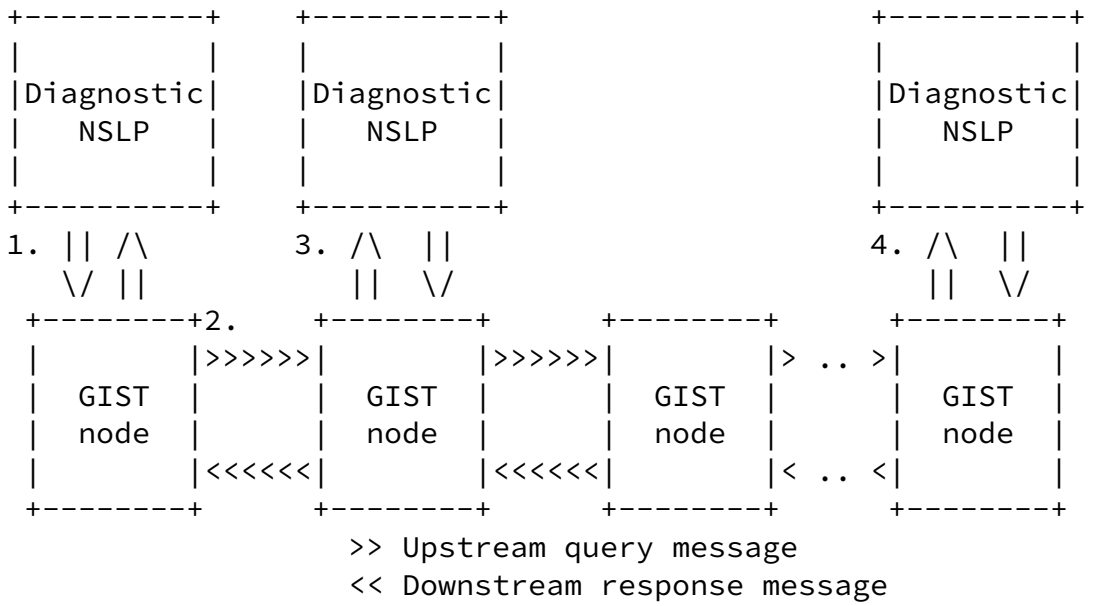
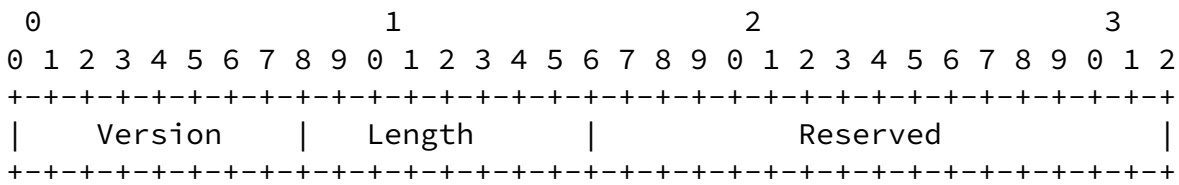


Figure 1: Diagnostic NSLP message flow

3.1.1. Basic message objects

3.1.1.1. Common Header

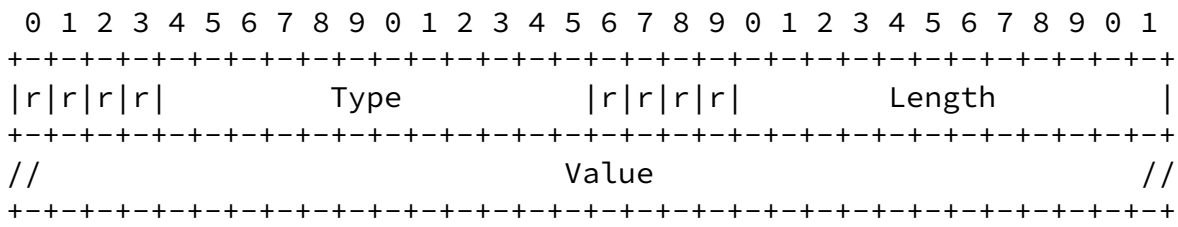


Version : Identifier for diagnostic NSLP
Length : Overall message length

3.1.1.2. Standard Object Format

Each object begins with a fixed header giving the object Type and object Length. This is followed by the object Value, which is a whole number of 32-bit words long.

0 1 2 3



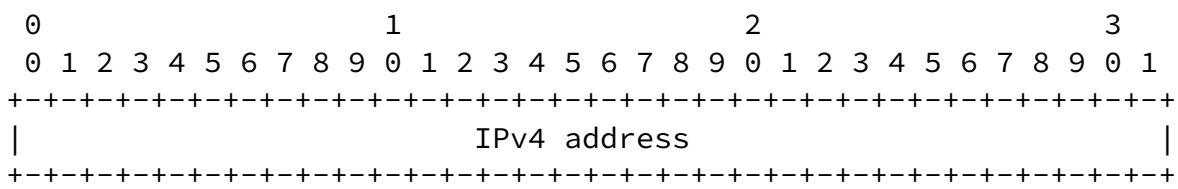
Type 00 : Query object
Type 01 : Hop object
Type 02 : IPv4 address object
Type 03 : IPv6 address object
Type 04 : General GIST information object
Type 05 : SID-bound Response object
Type 06 : NSLP state information object
Type 07 : Available NSLPs object
Type 08 : Additional information object object

3.1.1.3. Hop object

The hop object is just a container for the information objects requested in the Query object.

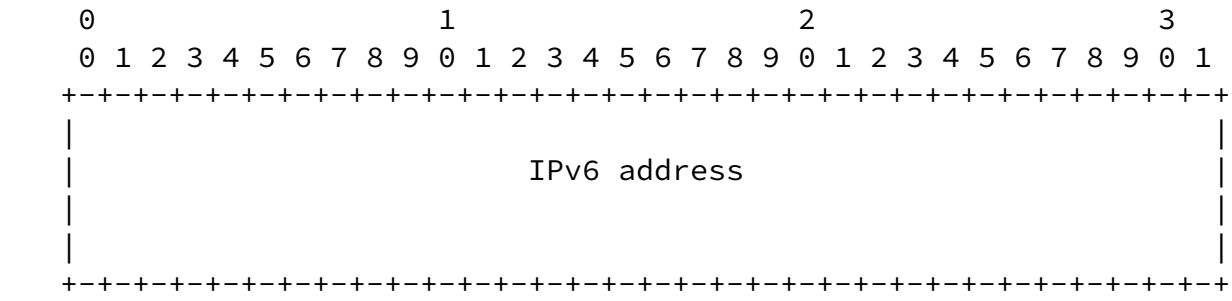
3.1.1.4. IPv4 address object

Every Hop Object may contain any number of IPv4 address objects.
Length: 4 bytes



3.1.1.5. IPv6 address object

Every Hop Object may contain any number of IPv6 address objects.
Length: 16 bytes



3.2. NTLP state information

As described earlier, we distinguish between general GIST state information and SID-bound state information. When collecting general GIST state information, one has to be careful to limit the amount of gathered information to comply to the basic prerequisites, especially in terms of expected message sizes. For example, when querying for larger sized information the possible maximum amount of nodes the query collects data from and replies it to the querying node, has to be taken into account to not exceed the maximum allowed message size and thereby risk the loss of data. Therefore, mechanisms to prevent overlimits of message size caused by the computation of (informative data size + header sizes) * amount of queried nodes, need to be present. For example, intermediate peers that identify further information to exceed the maximum size could intercept the query process by sending already collected information back to the querying node and trigger a re-querying. The querying node can now start querying information from the intercepting node towards the destination.

Hence we propose an extension to the message flow as shown in Figure 1 by introducing the possibility for every Diagnostic NSLP to act as a query intercepter:

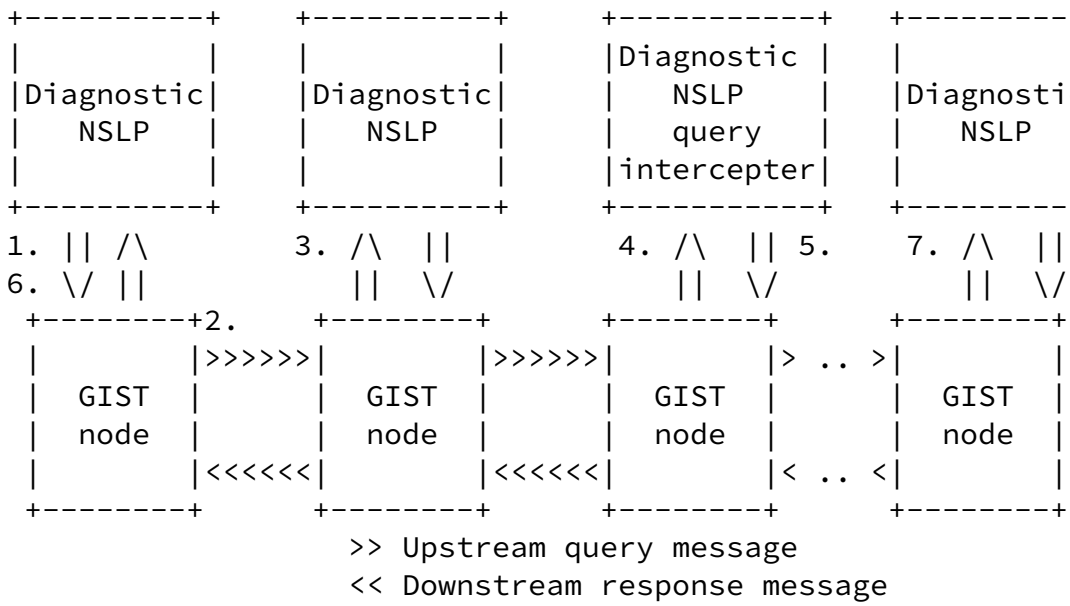


Figure 2: Extended message flow with intercepting node

Here we introduce 3 new actions:

1. At querying node, compose and send query message with designated destination - the final GIST node along the path
2. GIST at querying node forwards message to next GIST node towards the destination
3. Intermediate Diagnostic-NSLP-aware GIST nodes add queried information if message is on the downstream direction and forward to next peer
4. One peer discovers that adding the requested information would exceed the maximum message size and intercepts the querying process.
5. The intercepting node sends the already collected informational data downstream towards the querying node.
6. The querying node extracts the stores the data already collected along the path up to the intercepting node. It then sends a new query message directed at the destination node from the intercepting node on to collect data further along the path.
7. The destination GIST node adds queried information and forwards the message in the downstream direction

3.2.1. General GIST state information

This will be discussed in a later version

3.2.2. SID-bound state information

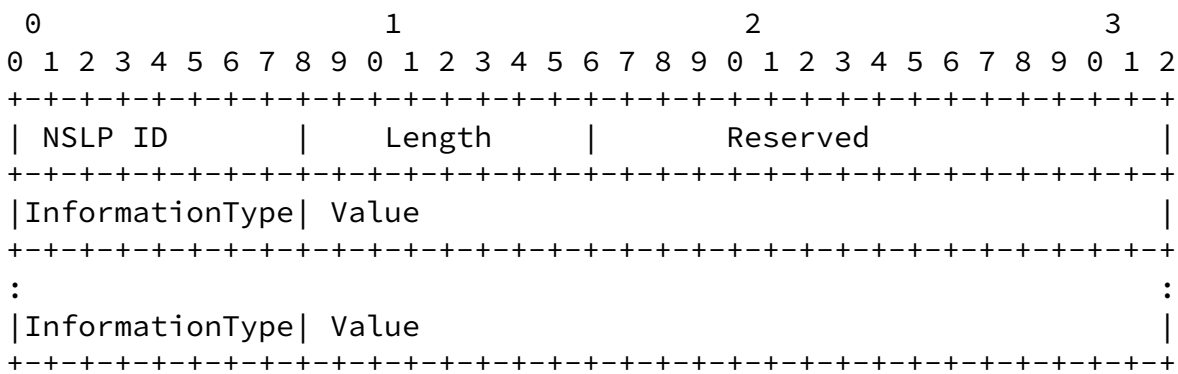
This will be discussed in a later version

3.3. NSLP state information

Which information to collect about states linked to a specific SID depends on the NSLP being queried. However, some basic information common to all NSLPs could still be gathered, for example the total amount of states connected to a specific SID. Querying of supported NSLP-IDs on each GIST node should be limited to Administrators only.

3.3.1. NSLP state information object

NSLP state information object.
Length: Variable



NSLP ID: Identifier for queried NSLP
Length : Overall message length
InformationType : Identifier for information that is to be queried on GIST nodes up to destination
Value : Value of queried information

Information Type could be any data queried at a GIST node which could be used for diagnosis. Example: Total amount of states maintained by a QoS NSLP, Value would then contain that amount. When acting as a query message, only the InformationType fields will be present and contain the identifier. Intermediate GIST nodes and the destination GIST node will then enter the values accordingly. This would also allow for querying multiple types of information.

GIST nodes not able to provide the requested information enter a value of 0 if they couldn't provide the information due to non availability and -1 if an error occured into the specific value field.

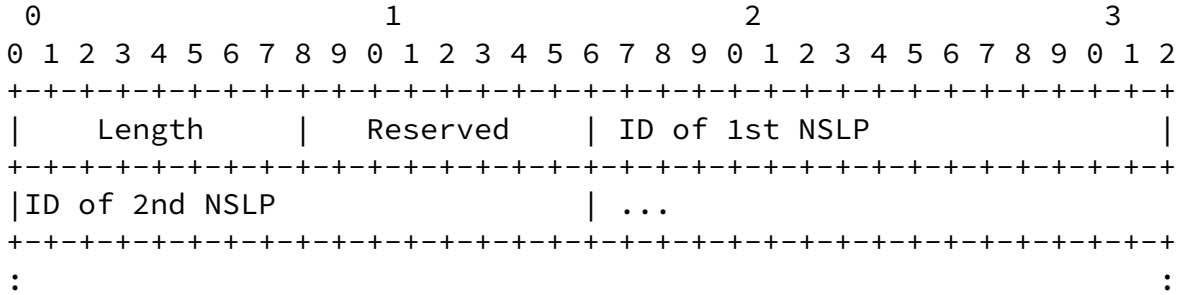
3.4. Query available NSLPs

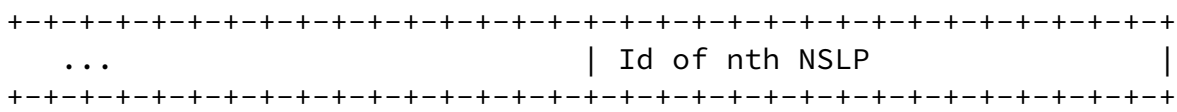
This scenario can be expected to be most widely used. Also, it should be the most easiest one to deploy and implement aswell as gather the requested information. The diagnostics NSLP simply queries for other NSLP IDs on the GIST nodes, collects them and adds their values to the RESPONSE object together with the total amount of IDs and finally its IP address.

3.4.1. Available NSLPs object

Available NSLPs object lists the NSLP IDs supported at the given GIST node.

Length: Variable (depends on the number of NSLPs supported on the GIST node)





Length : Overall message length
ID of Xth NSLP : NSIS ID of NSLP existing and supported by GIST node

3.5. Additional information

Additional information can be gathered by computing the minimum, maximum and average delay of a roundtrip with values collected by running an instance of the PING Deamon. Other possible data to be queried for can include the software version of GIST, the OS running GIST, amount of known peers and other useful information.

4. Security Considerations

Authorization and protection of the diagnostics messages seem to be two outstanding issues, among the various issues identified in [7].

On one hand, one may desire that only the state installer can query the session-specific state. For general information diagnostics, measures may be desired for allowing administrators only be able to make the operations across their own domains, or neighboring trusted domains.

On the other hand, the diagnostics messages carry senstive information that needs to be integrity protected. Some measures may be utilized such as reusing the secure MAs (if available) between the neigboring GIST nodes, or add NSLP level security mechanisms such as CMS.

A future version of this document will add more security relevant considerations.

5. Summary and Open Issues

We have discussed in this document how diagnostics functions for NSIS implementations as a common NSLP could be designed. Our intention is to keep it as simple and secure as possible.

Further possible additions to the diagnostics function could be diagnostics support for tunnelling and mobility devices.

A new NSLP ID needs to be defined if a common NSLP for diagnostics functions is devised.

6. IANA Considerations

A future version of this document will provide details about an IANA consideration.

7. Acknowledgments

The authors would like to thank Sebastian Willert, Henning Peters, Luis Cordeiro and Bernd Schloer for the discussion and implementation of the early ideas. Moreover, Robert Braden, Scott Bradner, Robert Hancock, John Loughney, Jukka Manner, Andrew McDonald, David Oran and Martin Stiemerling provided valuable comments.

8. References

8.1. Normative References

[1] Manner, J., "NSLP for Quality-of-Service signalling", draft-ietf-nsis-qos-nslp-09 (work in progress), February 2006.

[2] Stiemerling, M., "NAT/Firewall NSIS Signaling Layer Protocol (NSLP)", draft-ietf-nsis-nslp-natfw-09 (work in progress), February 2006.

[3] Schulzrinne, H. and R. Hancock, "GIST: General Internet Signaling Transport", draft-ietf-nsis-ntlp-09 (work in

progress), February 2006.

8.2. Informative References

[4] Terzis, A., Braden, B., Vincent, S., and L. Zhang, "RSVP Diagnostic Messages", RFC 2745, January 2000.

[5] Tschofenig, H. and R. Graveman, "RSVP Security Properties", RFC 4230, December 2005.

[6] Juchem, I., "A stateless Ping tool for simple tests of GIMPS implementations", draft-juchem-nsis-ping-tool-02 (work in progress), July 2005.

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

Authors' Addresses

Xiaoming Fu
University of Goettingen

Institute for Informatics
Lotzestr. 16-18
Goettingen 37083
Germany

Email: fu@cs.uni-goettingen.de

Ingo Juchem
University of Goettingen
Institute for Informatics
Lotzestr. 16-18
Goettingen 37083
Germany

Email: ijuchem@cs.uni-goettingen.de

Christian Dickmann
University of Goettingen
Institute for Informatics
Lotzestr. 16-18
Goettingen 37083
Germany

Email: mail@christian-dickmann.de

Hannes Tschofenig
Siemens
Otto-Hahn-Ring 6
Munich, Bayern 81739
Germany

Email: Hannes.Tschofenig@siemens.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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