# GENERIC IP SIGNALING SERVICE PROTOCOL

Thanh Tra Luu, Nadia Boukhatem
*GET-Télécom ENST Paris ; LTCI-UMR 5141 CNRS. 46 rue Barrault, 75 013 Paris. Email : {luu, boukhatem} @enst.fr*

**Abstract:** The Next Steps In Signaling (NSIS) working group has been recently created to design a generic IP signaling protocol supporting various signaling applications. This paper presents the Generic In Signaling Service Protocol (GISP) we designed considering the current outputs of the NSIS working group. In particular, we focus on the state management and message fragmentation using a simple mechanism to detect the path MTU.

**Key words:** Signaling, transport, protocol design and implementation

## 1. INTRODUCTION

During the last few years, several IP signaling protocols have been defined to support different signaling applications such as resource reservation, label distribution and middlebox configuration. Recently, the IETF has created the NSIS (Next Steps in Signaling) Working Group to design and standardize a generic signaling protocol supporting a large variety of signaling applications and managing general-purpose states.

The NSIS WG is considering protocols for signaling information about a data flow along its path in the network. The NSIS signaling problem is very similar to that addressed by RSVP[1,2]. Thus, the NSIS WG explicitly intends to re-use, where appropriate, the RSVP protocol and generalize it to support various signaling applications rather than the single case of reservation resource application.

To achieve generalization, the NSIS protocol stack is decomposed into two layers: a generic lower layer responsible for transporting the signaling messages, and an upper layer, which is specific to each signaling application. The lower layer is called NTLP (NSIS Transport Layer Protocol) and the upper layer is called NSLP (NSIS Signaling Layer Protocol).

In this paper, we propose a specification of the NTLP layer, called GISP (Generic sIgnaling Service Protocol), considering the requirements[3], the guidelines[4] of the NSIS working group and the analysis of the existing RSVP implementations.

## 2. NSIS CONSIDERATIONS

The NSIS protocol is envisioned to support various signaling applications that need to install and/or manipulate state related to a data flow on the NSIS Entities (NEs) along the data flow path through the network.

Each signaling application has its own objective. For example, the QoS signaling application[5] is only used to reserve resources on the data path, the NAT/FW signaling application[6] is used to configure NAT/Firewall devices, etc. To achieve its objectives, the signaling application requires support to exchange signaling messages between the signaling entities to install/manipulate state in these entities.

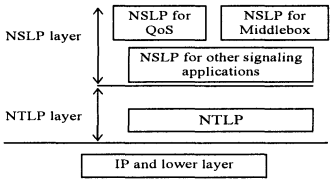


*Figure 1.* Protocol Signaling Structure

While the objective is tied to the signaling application itself, transporting the signaling messages and managing the state are the common functions of all or a large number of signaling applications. In order to achieve genericity, extensibility and flexibility, the NSIS framework proposed to split the protocol architecture in two layers to be able to support various signaling applications (figure 1).

The NSIS **Transport Layer** Protocol (NTLP) is responsible for transporting the signaling messages and supporting the state management. These functions are independent of any particular signaling application. Note that the transport layer has a peer-to-peer scope (or local scope). This means that NTLP only transports the signaling messages to its adjacent NSIS aware peers.

The NSIS **Signaling Layer** Protocol (NSLP) contains functionalities such as message formats and sequences, specific to a particular application. One NSLP is associated with only one signaling application. There can be an NSLP for QoS[5], an NSLP for middlebox configuration[6], etc. However, dif-

ferent NSLPs will use the common NTLP protocol to transport the signaling messages for their own signaling purposes. The NSLP has a scope, which is larger than the NTLP local scope. The NSLP signaling messages can be transported by the NTLP messages through many NTLP hops.

The main advantage of this layer model is its extensibility. Indeed, this facilitates the development of new signaling applications by using the transport and state management functions supported by the NTLP layer. However, this requires that the NTLP must be flexible and efficient enough to support existing and future signaling applications.

A NSIS signaling-aware entity (NE) can support many NSLPs. However, an NSLP can be installed on few NEs along the data path. For example, the NAT/Firewall signaling application[6] is only installed on the edge router of a private domain, whereas the QoS resource reservation signaling application[5] can be installed on all routers on the data path.

Actually, it is not reasonable to assume that all the equipments of an administrative domain will support the NSIS protocol. The NSIS protocol must therefore provide correct protocol operation even when there are non-NSIS-aware entities between two NSIS-aware entities. Furthermore, the NSIS protocol should support multiple signaling applications; it is very likely that a particular NSLP will only be implemented on a subset of the NSIS-aware nodes on a path. Therefore, in a heterogeneous environment, different kinds of nodes can be defined: non-NSIS-aware nodes, NTLP-only-aware nodes and NSIS-aware nodes supporting one or more NSLP.

## 2.1 NSIS transport layer functionalities

An overall NSIS signaling is the joint result of the NSLP and NTLP operations. As mentioned above, the NTLP has a peer-to-peer scope and operates only between adjacent signaling entities. Larger scope issues including end-to-end aspects are supported by the NSLP layer.

The functionalities of the NSIS protocol layer have been identified in the NSIS framework[4]. The NTLP is responsible for transporting the signaling messages between the peer NE nodes (upstream and downstream peer). The transport functions are the reliable message delivery, congestion control, bundling of small message, message fragmentation, and security protection.

Internet signaling requires the existence and management of state within the network for several reasons. Therefore, the NSLP should maintain the state for signaling sessions. However, the NTLP manages its own state to support the signaling application and it is unaware of the difference in state semantics between different types of signaling application.

## 2.2 Transport functions

Reliable and non-reliable message delivery: the NTLP should support mechanisms to confirm the reception of a signaling message if this is required.

Overload control: the NTLP should support overload control mechanisms. This allows the signaling protocol to coexist with other traffic flows and holds the performance of the signaling to degrade gracefully rather than catastrophically under overload conditions.

Messages bundling: the NTLP should support means to bundle signaling messages.

Fragmentation and assembling: the NTLP should be capable of fragmenting/assembling the signaling messages if the total length of message exceeds the link MTU (Maximum Transmission Unit).

Security protection: The NTLP only has a local scope. Thus, the NTLP only supports the security protection for the message transport between the adjacent NEs. The security protection that the NTLP should support includes integrity, anti-replay and confidentiality.

## 2.3 Signaling functions

Soft-state management: the NTLP should be capable of supporting soft-state management to support the signaling applications. The soft-state means that after being established on an entity, a state will be deleted on that entity if it is not periodically refreshed during a specific time period. The soft state is used to avoid the faults and the cases in which using explicit commands to delete an established state cannot be done (e.g. an intermediate router is shut down).

## 2.4 Other functions

Besides the transport and the signaling functions, the NTLP should also support other functions such as follows.

Multiplexing and demultiplexing signaling messages of different NSLPs.

The NTLP should be capable of sending notification reflecting the changes occurring in the network status (e.g. routing change, congestions) to the NSLP applications and other NSIS-aware entities.

# 3. GISP PROTOCOL

The GISP protocol is designed to run directly on the IP layer or UDP protocol. It is easier to create a signaling protocol without constraints by using IP or UDP as an underlying layer. However, all functions of the signaling protocol must be designed and implemented within the protocol to satisfy the requirements of signaling applications and the requirements of NSIS working group.

A GISP message consists of a common header, followed by a body consisting of a variable number of variable-length objects. The description of GISP objects can be found in our draft[7] with some minor changes. In this paper, we only focus the analysis on state management and message fragmentation.

The GISP is responsible for transporting signaling messages and supporting generic signaling functions. When the GISP is required to support a signaling session by its own NSLP upper layer or by receiving a signaling message, the GISP layer will establish a GISP state for that session. Once the GISP state of the signaling session is established, the GISP layer can filter, (de)multiplex the signaling messages, manage the state and notify NSLP applications about changes in the network (e.g. routing change, congestion).

## 3.1 Signaling service functions - State management

In order to support NSLP signaling applications, the GISP must install and maintain its own states on the data path. This is the prime purpose of the NSIS protocol. In the NSIS framework, the NTLP state can be installed on all NEs or only some NEs on the data path. In the following part of the paper, we will present how the GISP establishes and manages the state on the data path. We will also describe how the GISP messages are defined to efficiently support the state management tasks for signaling applications.

The GISP protocol can operate in the stateful or stateless mode. In the stateful mode, the GISP must register and maintain the state concerning a particular data flow. In the stateless mode, the GISP processes the message without installing any state concerning that message. Note that, in the same signaling session, some signaling-aware nodes on the data path run in the stateful mode (i.e. establish the state) while the others run in the stateless mode (i.e. do not establish the state for this session)[8].

## 3.2 State management modes

As we discuss above, the GISP supports both stateful and stateless mode for signaling applications. The GISP defines five state management modes

(SM) to indicate how the state is establish on the signaling path. The SM value is specified by the SM field in the GISP message header (see our draft[7] for more details).

#### 3.2.1.1 SM=0.

The SM=0 can be used for directly sending signaling messages between two signaling-aware entities (SE) without interference of intermediate nodes. In other terms, the message is forwarded by the intermediate nodes without establishing state.

In figure 2, the signaling entity which initiates the session (signaling initiator) sends the message with the SM field = 0. Thus, the state is only established on the signaling initiator (SI) and the signaling responder (SR) which terminates the signaling path.

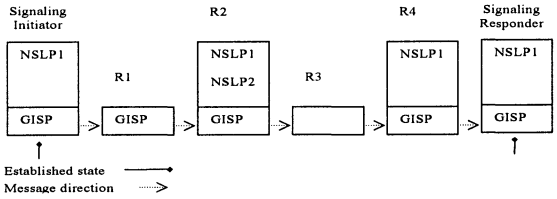


*Figure 2.* Establishing states in case SM=0

A message sent with SM=0 may have a router alert option (RAO) in its IP header to assure the hop by hop security or other purposes. The IP source and destination address are the address of the entity that sends and receives the message respectively.

#### 3.2.1.2 SM=1.

In case of SM=1, the GISP state is established on all signaling-aware nodes along the data path regardless of signaling applications. The SM=1 is therefore only used for the signaling applications requiring the signaling path must be seriously tied to the data path (e.g. resource reservation). A message sent with SM=1 has a RAO in its IP header to allow the next signaling nodes to examine the message.

In case a signaling-aware node does not support the NSLP specified in the message, this node only registers the content of message without analyzing it. If this node detects the changes in the network, it can send the registered message to establish rapidly the session on the new path. This provides a fast adaptation to route changes. Note that, the signaling application message content is always opaque to the GISP.

#### 3.2.1.3 SM=2.

In case of SM=2, the GISP state is established only on the SEs that support the NSLP specified in the message. The NSLP signaling application type is indicated by the NSLP-ID field in the GISP message. A message sent with SM=2 also has a RAO in its IP header to allow the next signaling nodes to examine the message.

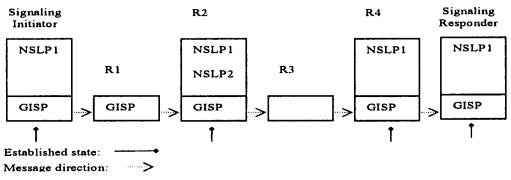


*Figure 3.* Establishing states in case SM=2

The SM=2 can be used for signaling applications which are not sensitive to routing changes as NAT/FW_NSLP[6]. In figure 3, the SI sends the signaling message with the SM=2 to the SR to establish a session for the NSLP1 signaling application. The intermediate nodes R2 and R4 support the NSLP1, and the state is only established on these nodes. The routing changes between the node R2 and the node R4 do not heavily influence the NAT/FW_NSLP application. Normally, the NAT/FW_NSLP is only installed on the edge nodes of private domain. Thus, the routing changes between two domains do not affect the signaling application.

#### 3.2.1.4 SM=3.

The case of SM=3 looks like to SM=2. However, if the state is not yet established, the GISP must wait the decision of the NSLP signaling application to know whether the state will be established. In some cases, the GISP cannot decide to establish the state by itself (e.g. lack of information).The mode SM=3 allows the NSLP signaling application decide to establish the state or not.

#### 3.2.1.5 SM=4.

In case of SM=4, the message is not involved in state establishment. It is used by the GISP to exchange information about the states that have been established (refresh, delete states) or other information (congestion control, security control...).

### 3.2.1 State refresh mechanism

In the original RSVP version[1], an established state is refreshed by resending the Path and Resv message on the data path. Every Path (or Resv) must

be totally examined even though the session is established. This increases the cost of message processing.

The RSVP extension[2] has improved the refresh mechanism. The RSVP messages are categorized into two types: trigger and refresh message. Trigger messages are the messages that advertise state or any other information not previously transmitted. Refresh messages transmit previously advertised state and contain exactly the same objects and same information as a previously transmitted message, and are sent over the same path. Every trigger and refresh messages uses a MESSAGE_ID (message identifier), which uniquely identifies the message. Note that a MESSAGE_ID has a hop-by-hop scope and concerns two specific adjacent RSVP-aware routers.

The MESSAGE_ID identifying the refresh message has the same value as the MESSAGE_ID of the trigger message that was previously sent.

When an established state needs refreshing, the RSVP only sends the MESSAGE_ID of that state. This avoids sending the whole message to refresh the state. If the trigger message content needs to be modified, the RSVP sends a new trigger message with new incrementally changed MESSAGE_ID. As a result, the MESSAGE_IDs values of established sessions on two adjacent nodes are not necessarily consecutive. Therefore, the RSVP must send each MESSAGE_ID to refresh a specific session.

In the GISP, we define a new mechanism to reduce the refresh cost. This mechanism is also applicable to RSVP to reduce the refresh cost. Each signaling-aware entity (SE) has a list of established sessions with a particular adjacent SE. We call this list Socket_ID list. The index numbers of each entry in the list are called Socket_ID. Each Socket_ID is associated with a Session_ID value to identify a specific established session between two adjacent SEs. The GISP will choose the Socket_ID for each new state and this value is not reused until the state is deleted. When establishing a new state, the GISP chooses the first unused Socket_ID (i.e. there is no established session corresponding to this Socket_ID) in the list. As a result, the Socket_ID of established states are consecutive. Therefore, the GISP only sends the first and the last Socket_ID values to refresh all the sessions having the Socket_ID values in this range. The figure 4 illustrates this mechanism.

Suppose that the node A sends a trigger message for Session_ID=K for establishing a session on the node B (see figure 4). When receiving the trigger message, the node B looks for the first empty Socket_ID (Socket_ID=2) and sends this value back to the node A. When A wants to refresh the states that are established on the node B (Session_ID I, K and A), it sends the socket blocks (Socket_ID=1, Socket_ID=3). If an established session is deleted, the socket entry value is set empty (null). This null socket entry will be filled with a new session.

Note that, the A's socket list is a mirror of B's socket list. The node A can deduce the first null socket in the socket list of the node B. However, the value of the first null Socket_ID must be sent to A to make the GISP more robust. In some cases, a signaling-aware can loose the information about the established states (e.g when the node reboots). The node A can use the Socket_ID sent back from the node B to synchronize the two lists and detects the errors.

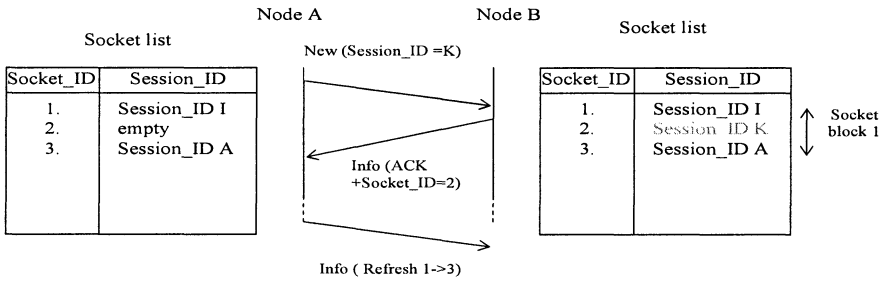


*Figure 4.* States refreshing in the NTLP

This mechanism aims to reduce the payload used to refresh the state between two nodes. Note that this mechanism allows refreshing a list of session rather than refreshing each individual session. All the sessions in a list are refreshed at the same time. It saves the CPU resources to manage the timeout for each session.

In addition, the GISP proposal allows two adjacent NSIS-aware nodes have more than one Socket_ID list. Each list has a different timeout value. If a particular session is more stable, we can place it in a list having a longer timeout. This allows the signaling application change the refresh period for a specific state in the same manner as the staged refresh timer mechanism[11] but still avoid managing the soft state for each individual state.

### 3.2.2 The GISP message

In the GISP protocol, two states are associated with a session: forward state and backward state. The forward state concerns the information sent by the SI along the downstream path. The backward state concerns the information sent by the SR along the reverse data path. The forward state is usually created before the backward. The backward state is not always created. If the forward state of a session is deleted, the backward state of that state is also deleted. However, if the backward state is deleted, the forward state of a session still exists.

The GISP supports three message types that are used to establish and remove the states (forward and backward) on the data path. They are also used to exchange signaling information between the SEs.

#### 3.2.3.1 New message.

A New message is used to establish the forward state for the downstream path between two signaling entities. The SI will send a New message along the data downstream path. The New messages can be sent with the Router Alert IP (RAO) option in their IP headers to allow the GISP on intermediates nodes to examine signaling messages.

When the New message is sent with a RAO option in the IP header, intermediate SEs will examine it. The GISP is responsible for examining signaling message to know if it must establish a forward state to support the signaling application. The decision depends on the SM field of message and the signaling application that the GISP supports.

In the RSVP implementation, when receiving a Path message, a node must detect if the session has been established or not in the binary search tree. This wastes the CPU resource and increases the session establishment latency. In order to improve the message processing, the GISP uses the New message to establish new sessions. The New message is used once in a session. If the state on the path changes, the New message cannot be used to re-establish the session. In this case, the GISP sends a Mod message to re-establish the session.

#### 3.2.3.2 Mod message.

The Mod message has the same structure as the New message. It is used to modify a forward state of a session. If the sender wants to modify the signaling session, he sends a Mod message along the data path toward the receiver. The GISP on the data path will examine the message and decide to deliver it to the NSLP layer or not.

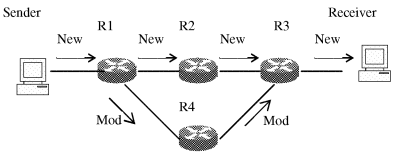


*Figure 5.* Establishing forward states by using New and Mod message

The sender or the intermediate nodes can periodically send the Mod message to refresh and detect a route change in the network. When a SE receives a Mod message that is was previously received, it considers this message as a refresh notification from the previous node.

In case a SE node detects a route change (ex. receive a route change notification from the routing table), it sends a Mod message to establish the state on the new path. This is illustrated in figure 5. When the router R1 detects a

route change, it sends a Mod message on the new data path until the message reaches the node R3.

The Mod message is also used to establish the backward state if the receiver requires this. In this case, the Mod message will be sent hop by hop along the reverse data path. Note that the backward state is established if and only if the forward state is established. Thus, the SE always verifies the existence of the forward state before establishing the backward state. That is the reason why the New message cannot be used to establish backward states.

#### 3.2.3.3 Info message.

The Info message can be sent hop by hop along the data path or the reverse data path. It is responsible for exchanging information between adjacent SEs nodes. The information can concern state management (refresh, delete an established state), reliable message delivery (acknowledge a message) and other information (congestion control, error notification, etc).

Moreover, the Info message allows to GISP to detect route changes (as the Mod message) and discover the path MTU (Maximum Transmission Unit).

The Info message does not carry the NSLP message content to refresh an established state; it only carries the Session_ID of the established session. This reduces the signaling payload in the network. When receiving the message, the GISP detects if the Session_ID corresponds to one of the established states. If there is a corresponding state, the GISP considers this message as a refresh notification for that state. Otherwise, the GISP will send back a notification to require sending a Mod message to establish the state.

In the following section, we present how the Info message is used to detect the path MTU.

## 3.3 Transport service functions

The GISP directly runs on the IP or on UDP, which do not meet the requirements in the NSIS WG. Thus, the GISP must implement all the generic transport service functions specified in the NSIS requirements[3,4]. The GISP has its own transport functions such as follows: reliable message delivery, message bundling, message fragmentation and the path MTU discovery, congestion control, security transport

### 3.3.1 Reliable message delivery

The reliable message delivery is considered as a significant service for the signaling applications in the NSIS framework. This allows the signaling applications to receive the confirmation of signaling message reception. A

NSLP signaling application can invoke the reliable delivery service for its own purpose. This is always optional to use.

The GISP supports mechanisms to confirm of the reception of a signaling message if it is required. If a message is required to transport reliably, the GISP sets the ACK flag in the MSG_SEQ object, which contains the sequential number of the message. When a SE receives this message, it will send back a MSG_ACK that contains the sequential number of the MSG_SEQ object.

### 3.3.2 Message bundling.

The GISP supports the message bundling for the signaling applications. The GISP can send more than one NSLP message in a GISP message. The GISP puts each NSLP messages in a GISP object and transports these objects in an only one GISP message. When receiving a bundle message, the GISP decapsulates it and process each NSLP message as it was received individually.

### 3.3.3 The message fragmentation and the path MTU discovery

For NSLPs that generate large messages, it will be necessary to fragment them efficiently within the network. The GISP is required to support the message fragmentation. However, using the IP fragmentation can be incompatible.

Note that in the RSVP, the IP source address and the IP destination address of an RSVP downstream trigger are the address of the sender and receiver respectively. If an intermediate router does not fragment the message (e.g. Don't Fragment bit is set or IPv6 packet) a "Packet Too Big" ICMP message will be sent to the sender address rather than to the previous node that sent the message. Thus, the previous node cannot know why the message is lost.

The GISP protocol uses a simple mechanism to detect MTU between signaling aware nodes. The GISP only uses this mechanism to send a trigger message (New or Mod) if the message length is larger than 576 bytes (IPv4) or 1280 bytes (IPv6). After sending the trigger message, the GISP will send an Info called Info Detect message. The Info Detect message only carries the information about the SE that sent this message (e.g. IP address) and Session_ID of the session. Because the Session_ID length is 16 bytes, the Info Detect message length is always smaller than 576 bytes (IPv4) or 1280 bytes (IPv6).

In case an intermediate node drops the trigger message, the next SE node only receives the Info Detect message. This node waits for a short time in-

terval before sending back an Info message, called Info Failure message. This Info Failure message includes the next SE's address and the link MTU value of the interface through which the Info Detect message was received.

When the SE that sent the trigger message receives the Info Failure message, it knows the address of the SE and the link MTU of the interface through the Info Detect message was received. Thus, the GISP can use the path MTU discovery mechanisms for IP to detect the path MTU which are defined in RFC1191 and RFC1981. With this path MTU value, the GISP will fragment the trigger message and send individual fragments towards the next SE. Theses fragments will be assembled at the next SE and sent to the signaling application layer.

### 3.3.4 The congestion control

The RSVP does not implement any kind of congestion control algorithm. In case the RSVP traffic is low, the co-existence between RSVP and other TCP-like protocol is acceptable. Nevertheless, when the RSVP traffic is quite high, its flows are very aggressive and starves other TCP and TCP-like flows.

In this paper, we only show the two main difficulties to support the congestion control in the GISP. The solution for congestion control in the GISP is still under investigation. Firstly, the GISP has two ways to address a signaling message as the RSVP:

Peer-to-peer: the message is addressed to an adjacent signaling-aware. In this case, the message is sent directly between two signaling nodes without the interference of the intermediate nodes.

End-to-end: the message is addressed to the flow destination directly, and intercepted by intervening signaling-aware nodes.

In the first way, the signaling messages are sent between two explicit nodes, the GISP can reuse the congestion control mechanism for TCP as in the RFC2581[9]. However, in the second way, the GISP does not know which SE on the data path will intercept the message. Thus, the congestion control for the TCP cannot be used.

Secondly, the GISP supports the reliable and unreliable message delivery for the signaling applications. It is not always required to send back the acknowledgement of a message. In the congestion control mechanisms[9,12], the acknowledgement of packet is used to analyze the congestion in the network. If a packet sender does not receive the acknowledgement of a specified packet, it considers the packet is lost and there is congestion in the network. However, the GISP cannot consider that this message is lost if it does not receive the acknowledgement of that message since the acknowledgement is not always required.

### 3.3.5 Security protection

The GISP only supports the security protection of the message transport between two adjacent NEs. Each SE must establish security associations (SA) with the other adjacent SEs before sending the GISP message. The GISP uses existing protocols to establish the SAs (e.g IKE). To support the message authentication, integrity and anti-replay security service, the GISP reuses the security mechanism proposed in the RSVP[10].

## 4. CONCLUSION

The NSIS WG has proposed a generic framework for the design of an IP signaling protocol. The NSIS protocols architecture is decomposed in the NSLP and NTLP layers. We have designed GISP protocol to be the generic NTLP layer taken into account the requirements of the NSIS WG and based on the analysis of the existing RSVP implementations.

In this paper, we presented the main characteristics of GISP protocol and proposed a new refresh mechanism to reduce the refresh payload and the CPU resource used for the soft-state management. We also proposed a mechanism to discover the path MTU between two signaling-aware nodes. This allows GISP to determine the appropriate length of the signaling messages and decreasing the load of the header message processing.

## REFERENCES

1. R. Braden et al., "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specifica tion", RFC 2205, September 1997.
2. L. Berger et al., "RSVP Refresh Overhead Reduction Extensions", RFC2961, April 2001
3. M. Brunner et al., "Requirements for Signaling Protocols", RFC 3726, April 2004.
4. R. Hancock, Next Steps in Signaling: Framework, draft-ietf-nsis-fw-05, Internet Draft, Work in progress, October 2003
5. S. Van den Bosch et al., NSLP for Quality-of-Service signaling, draft-ietf-nsis-qos-nslp-02, Internet Draft, work in progress, February 2004.
6. M. Stiemerling et al., "A NAT/Firewall NSIS Signaling Layer Protocol (NSLP)", Internet draft, work in progress, draft-ietf-nsis-nslp-natfw-01, February 2004
7. Thanh Tra Luu et al., "NTLP Considerations and Implementation", Internet Draft, work in progress, draft-luu-ntlp-con-imp-01, May 2004
8. A. Bader et al.,"RMD (Resource Management in Diffserv) QoS-NSLP model", Internet Draft, Work in progress, draft-bader-RMD-QoS-model-00, February 2004
9. M. Allman et al., TCP Congestion Control, RFC 2581, April 1999
10. F. Barker et al., RSVP Cryptographic Authentication, RFC 2747, January 2000
11. Ping Pan et al., "Staged refresh timers for RSVP", Global Telecommunications Conference, 1997. GLOBECOM '97., IEEE , Volume: 3 , 3-8 Nov. 1997 Pages:1909 - 1913 vol.3
12. Datagram Congestion Control Protocol working group http://www.ietf.org/html.charters/dccp-charter.html