Signaling in IP-based Networks: Retrospect and Prospect
From QoS Resource Reservation to Software-Defined Networking

Xiaoming Fu, Senior Member, IEEE

Abstract—Various protocols have been developed to provide various signaling services in IP-based networks, such as for QoS reservations, or configuring middleboxes such as NATs and firewalls. This article presents a retrospective view on the development of IETF Next Steps in Signaling protocols, with a more flexible signaling framework than RSVP. The challenges for signaling in emerging network architectures like software-defined networking, network function virtualization and service function chaining, as well as the ongoing standardization efforts are also discussed.

1 INTRODUCTION

One fundamental principle of IP-based packet-switched networks like the Internet has been multiplexing a huge amount of small data packets and processing them in a best effort manner, without maintaining flow-based state in routers. With the emergence of various applications, this philosophy has been revised to accommodate the needs, by allow certain more complex functions to a designated set of flows, such as Quality of Service (QoS) resource reservation, Network Address Translators (NATs), stateful packet filtering firewalls, or to collect the network performance data for an end-to-end path. In order to enable these functions, a signaling protocol is essential to install and manipulate states in network nodes. Such states can be used for different application purposes, for example allocated available bandwidth shares, NAT pinholes, firewall rules, or Performance Enhancement Proxy (PEP) configurations.

Signaling protocols can be broadly categorized as either soft state or hard state based. Hard state signaling protocols require explicit state removal operations and thus may not be a best choice for packet-switched networks which are subject to packet losses. Instead, soft state signaling protocols are preferred in IP-based networks as installed state can be removed automatically if it does not receive a state refresh or update message after a given interval. In principle, for ensuring the soft state effectively installed and maintained, tradeoffs need to be made for setting proper timing interval settings in different environments [1].

Providing QoS – and hence signaling for QoS reservation state – has been envisioned as the key requirement to deliver multimedia applications over the Internet since 1980s. During the early days, the Internet Engineering Task Force (IETF) designed STREAM protocol Version 2 (ST-II) [2], an experimental hard-state signaling protocol which supports point-to-multipoint multicast communication. To overcome some shortcomings of ST-II, the Resource ReSerVation Protocol (RSVP) [3], [4] was then designed to provide support for multipoint-to-multipoint resource reservations in a soft state manner. Besides soft state, RSVP introduces a number of niche features for IP signaling, such as two-way state message exchange, independence of (multicast) routing, introduction of router alert option [5] for node discovery, and modularity through the introduction of opaque objects. It therefore quickly emerged at the forefront of academic and industrial interests and was regarded as the most promising candidate for a basic signaling protocol for IP-based networks.

Originally designed specifically for QoS resource reservation in couple with the Integrated Services model instead of general IP signaling, RSVP has been questioned for its complexity, security, scalability and modularity in meeting new requirements over the years [6]. To address these problems, the IETF Next Steps In Signaling (NSIS) working group (https://tools.ietf.org/wg/nsis) was formed in 2001 (and closed in 2011) to develop next signaling architecture and protocols. Meanwhile, the IETF developed further signaling protocols for other purposes, for example, Diameter [7] for providing AAA (authentication, authorization, and accounting) functionality for both wired and mobile network access, and RSVP-TE [8], LDP [9] and PCEP [10] for traffic engineering over MPLS.

These protocols, while some still in forms of experimental RFCs, could reinvigorate the drive for standards to provide various signaling services in IP-based networks, such as for QoS reservations, configuring middleboxes such as NATs and firewalls, or for managing states in emerging network architectures like software-defined networking (https://www.opennetworking.org/sdn-resources/sdn-definition), network function virtualization (https://tools.ietf.org/wg/nvo3) and service function chaining (https://tools.ietf.org/wg/sfc). A retrospect on these protocols would allow network designers, engineers and practitioners to better understand, develop and deploy value-added services to meet the dynamic needs.

The rest of the article is organized as follows. Section 2 discusses the general principles of signaling and challenges that existing protocols face. Section 3 presents the NSIS signaling framework and protocols as an example for sig-
naling protocol design, and how it addresses the signaling requirements envisioned during its development. Section 4 elaborates new challenges emerged from signaling in new architectures like software-defined networking (SDN) and service function chaining (SFC), and ongoing related standardization efforts.

2 IP Signaling Principles and Challenges of Prior Solutions

2.1 IP Signaling Principles

Generally, a signaling protocol for IP-based networks has to consider the following aspects:

- **Signaling application – the semantics of state**: State installed in routers shall serve for the ultimate application semantics, such as QoS reservations or firewall pinholes.
- **State management**: A signaling protocol needs to be able to create, refresh/update and remove the state, with proper state management mechanisms via actively generating signaling messages upon application needs, or internal state removal when the state timer expires. This requires setting the timers appropriately.
- **Message transport**: A signaling protocol needs to encapsulate and deliver the signaling messages in an appropriate manner.
- **Node discovery and network/route change detection**: When the nodes to be signaled are unknown, or nodes are changed during the lifetime of a signaling session (e.g., due to node failure or route changes), they need to be discovered by the signaling protocol.
- **Signaling session identification**: A signaling session needs to be identified uniquely to reflect the state to appropriately serve end-to-end application sessions, thus ensures that e.g., in case of node mobility, a same signaling session will not result in duplicated state.
- **Security considerations**: The aforementioned functions have to be performed in a secure manner to protect against misbehaving or malicious nodes.

2.2 Challenges of Prior Solutions

Signaling protocols developed before NSIS, represented by RSVP and its variants, although well able to offer soft state signaling functionality, faced a couple of challenges:

- **Signaling application**: In RSVP-based protocols, state installed in routers is closely coupled with the corresponding application semantics, in particular QoS reservations for Integrated Service model. Therefore, when signaling is desired for other applications such as middlebox configurations, these protocols are not easily adaptable.
- **State management**: RSVP introduces PATH and RESV two-way signaling exchange (with a PATH state remembering its previous RSVP hop) and soft state timers in routers for effective state management. When the network situation is highly dynamic, a fixed timer or refresh interval might not work well.
- **Message transport**: RSVP uses UDP or IP to encapsulate its messages (PATH and RESV). For the reliability of message delivery, [11] reintroduced hop-by-hop acknowledgement, which is a common feature of available transport protocols like TCP, DCCP or SCTP.

Node discovery and network/route change detection: RSVP introduces an IP router alert option in the PATH signaling messages for discovering unknown signaling nodes on the path. However, as a PATH signaling message carrying state semantics would always include discovery function, this brings difficulty from the security perspective.

**Signaling session identification**: An RSVP signaling session is identified by an RSVP sessionID destination address, destination port, protocol ID, which cannot uniquely identify a state for the end-to-end session it serves for. Due to node or session mobility, a same signaling session may result in duplicated states installed in (part of) the network, e.g., when Mobile IP is used for mobility management, or when port changes during the lifetime of a session.

**Security considerations**: RSVP uses keyed message digest to provide integrity protection of a signaling message. However, it assumes pre-shared keys are available and does not specify key management mechanism to create the desired security associations. In a network path with unknown nodes, it is rather difficult to have pre-shared keys available.

3 NSIS Signaling Framework and Protocols

3.1 Key Design Choices of NSIS Framework

The NSIS framework is based on the following key design choices [12].

**Separating General Transport Service from Signaling Applications**: To satisfy the need for different signaling applications, NSIS separates the message transport functionalities such as reliability, congestion control and integrity from signaling applications, leading to a two-layer architecture [13]:

- An NSIS Transport Layer Protocol or NTLP, where the key function element is GIST [14], [15], which provides the messaging function for the signaling application layer messages. Instead of reinventing the wheel, GIST reuses standard transport and security protocols.
- NSIS Signaling Layer Protocols or NSLPs, which provides application-specific signaling services, e.g., QoS NSLP for resource reservation [16] and NAT/firewall NSLP [17] for middlebox configuration.

**Separating Node Discovery from Message Transport**: One design choice in GIST design was to separate NSIS node discovery from message transport. Different from RSVP, where discovery and signaling message delivery are coupled in a single protocol procedure, a distinct and extensible discovery component is used in GIST which can rely on IP router alert options or other approaches, such as routing tables or DNS extensions. Furthermore, during the discovery process, transport protocol and security mechanisms may be negotiated between the neighbouring peers.

**Session Identifier, Mobility Support and Security**: Unlike RSVP, an NSIS session identifier is a cryptographically random number, for identifying a signaling session and signaling state with probabilistic uniqueness. In case the source or destination IP address of an end host changes due
to mobility, the session identifier stays the same, ensuring the state being managed with a unique identifier.

Due to its layered design, NSIS has the built-in flexibility to offer different security features for different layers:

- **GIST** offers authentication and security association establishment between neighboring peers, assisted with an additional cookie mechanism to protect discovery process. To secure the message transport, existing security protocols such as IPSec (over IP), TLS (over TCP) and Datagram TLS (over UDP, DCCP or SCTP) may be used, depending on the negotiated transport protocol.

- The NSLP layer is responsible for signaling service-aware authorization, possibly in combination with an AAA protocol (e.g., Diameter).

3.2 General Internet Signaling Transport Protocol

GIST [14], [15] creates and maintains two types of states related to signaling transport: a per-flow message routing state for managing the processing of outgoing messages, and a message association state for managing per-peer state associated with connection mode messaging to a particular peer. The latter consists of signaling destination address, protocol and port numbers, as well as internal protocol configuration and state information. In addition to information about its neighbor NTLP peer, GIST also maintains certain message routing information such as the flow identifier, the NSLP type and session identifier, to uniquely identify the signaling application layer session for a flow.

GIST has two modes of operation: the **datagram mode**, which uses an unreliable unsecured datagram transport mechanism, with UDP as the initial choice; and the **connection mode**, which uses any stream or message-oriented transport protocol, with TCP as the initial choice and DCCP/SCTP as other options. It may employ network layer security associations such as IPSec, or a transport-layer security association such as TLS for TCP, and DTLS for DCCP or SCTP [18]. It is possible to mix these two modes along a chain of nodes, without coordination or manual configuration. This allows, for example, the use of datagram mode at the edges of the network and connection mode in the core of the network.

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

- **Query message**, typically embedded in datagram mode, is to discover the next hop signaling node. If it contains a NSLP ID (say QoS NSLP), the next (QoS) NSLP-aware node will respond. It contains a Query-cookie to initiate the “Query-Response-Confirm” three-way handshake process for denial-of-service protection.

- **Response message** is to respond the Query message.

- **Confirm message**, is to finalize the GIST state establishment.

- **Data message** is to deliver NSLP signaling message using the message routing state without changing GIST message associate state.

- **Error message** to report an error detected at GIST level.

- **MA-Hello message** is sent over connection-mode to keep the GIST message association state alive.

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

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

GIST operates on soft state and refreshes of the routing state are done locally; it assumes NSLPs will refresh their own state in an end-to-end manner. Since it is possible that not all NSLPs are supported in a single NSIS node, in route change cases with GIST MA-keep-alive messages and state expiration mechanism GIST can detect the route change, update its own routing state consistently, and inform interested NSLP instances at affected nodes.

### 3.3 QoS NSLP

The basic operation of QoS NSLP [16], [19], [20] over the GIST service can be explained using an example, assume a QoS NSLP RESERVE message is created at the NSIS initiating host NI. The GIST module first constructs a GIST-query message, namely a UDP datagram, possibly including the QoS NSLP RESERVE message as payload. The message is addressed to the flow destination and labeled with a router alert option, similar to RSVP. The next downstream NSIS peer which supports the QoS NSLP, assuming R1, recognizes it and passes the QoS NSLP RESERVE message to its QoS NSLP daemon. The GIST daemon at R2 also recognizes the upstream NSIS peer who wants to learn about its downstream peer, and thus answers with a GIST response message. Upon the receipt of this response, the upstream NSIS peer (here, NI) creates a message association with the downstream peer (here, R1), using, say, TCP. All subsequent NSIS messages between these two peers can now be sent via this message association. QoS NSLP defines four messages:

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

QoS NSLP supports signaling for flexible QoS models, e.g., ITU-T Y.1541 QoS model [20], Resource Management in DiffServ (RMD) QoS model [21], and Integrated Service’s controlled-load QoS model [22].

QoS NSLP state is soft state, setting a state timer in QoS NSLP nodes and using RERERVE end-to-end refreshes/updates to update the latest network situation.

### 3.4 NSLP for NAT/Firewall Signaling

The main goal of NSIS NAT/FW NSLP [17] signaling is to enable communications between two endpoints in presence of NATs or firewall middleboxes, assuming these middleboxes support NSIS NAT/FW NSLP. The NAT/FW NSLP protocol is used to dynamically install necessary policy rules in all NAT/FW NSLP-aware middleboxes along the path, so that application data can traverse from the sender to the receiver. To do this, the end host application firstly triggers the local NSIS entity to signal along the data path. If it supports NAT/FW NSLP, an NAT/FW NSLP CREATE message will be initiated to establish policy rules and NAT bindings in all middleboxes along the path; these middleboxes would be discovered by the GIST next-hop discovery process.

The NAT/FW NSLP employs four types of signaling messages, most notably the CREATE and the EXTERNAL messages:

- The CREATE message is used for creating, changing, refreshing, and deleting NATFW NSLP signaling sessions, i.e., open the data path from data source to data receiver. It will be created by the sender, processed by every middlebox on the path and finally reaches the destination.
- The EXTERNAL message is sent from the source address to an external address and is intercepted by the edge-firewall or edge-NAT and not forwarded to the destination address. This allows signaling at the edge-middlebox without introducing long end-to-end signaling delays.
- The RESPONSE message provides response to CREATE and EXTERNAL request messages.
- The asynchronous NOTIFY message is used by NAT/FW NSLP peers to alert other NAT/FW NSLP peers about specific events (especially failures).

Like QoS NSLP, NAT/FW NSLP signaling sessions are maintained on a soft state basis.

### 3.5 NSIS Extensibility

NSIS is extensible in three main aspects [23]. First, NSIS supports any type of signaling applications, which is achieved by the two-layer splitting. When a new signaling application is desired, a new NSLP can be defined. Second, GIST provides the flexibility in adding and choosing new node discovery mechanisms. This may be achieved by adding new discovery entries. Third, GIST allows extending transport protocols and security protocols.

In this section we elaborate via an example of the third aspect, through the introduction of new message routing methods and incorporating its stack proposal information in the GIST three-way handshake.

The standard GIST specification [14] only specifies the use of UDP, TCP and TLS over TCP for GIST message transport. However, some NSLP context information has a validation lifetime, therefore, the GIST message transport must accommodate flexible retransmission, so stale NSLP
messages that are held up by congestion can be dropped. Together with the head-of-line blocking issue and other issues with TCP, these considerations argue that implementations of GIST should support the Stream Control Transport Protocol (SCTP) [24] as an optional transport protocol for GIST, especially if deployment over the public Internet is contemplated. In addition, the use of Datagram TLS [25] is also encouraged to secure unreliable communication channels e.g., when partial reliability SCTP or DCCP is used.

To extend SCTP or DCCP or datagram TLS over them as a possible GIST message (secure) transport mechanism between two neighboring nodes, one can introduce a new stack proposal for the negotiation of GIST transport protocol stack, in addition to the new protocol IDs assigned by IANA. The detailed description of such extension is given in [18].

4 Signaling for Emerging Network Services: New Challenges and Opportunities

The current network infrastructure is composed of many network elements, which are relatively static. New service provisioning paradigms, new mobile networks, data center networks and cloud architectures require more flexible network infrastructure and elastic service function deployment models. However, a new network protocol/function takes months to years to get standardized, designed, and learned how to operate. Moreover, these functions do not adapt well to elastic service environments enabled by virtualization.

To respond to this, Network Functions Virtualization (NFV) and Software Defined Networking (SDN) architectures are being developed to help operators more effectively manage resources and significantly reduce overall deployment and upgrading costs. This way, network functions like middleboxes and network service function chaining can be run on off-the-shelf servers by using modern virtualization technologies.

Conceptually, SDN separates a centralized and more complex control plane from fast, dumb data plane functions (of whitebox switches), allowing creating intelligent networks that are open, programmable and application aware. SDN decouples the network control and forwarding functions, enabling the network control to become directly programmable under logically centralized controllers. One or more SDN controllers can be responsible for the control of one SDN network. In order to enable an end-to-end service, negotiation and configuration of SDN controllers effectively applicable for the service along its data path, usually called SDN east-west interface, is needed.

NFV defines the architecture to host network features and functions on general-purpose servers instead of specific-purposed network devices. NFV decouples hardware from software that transforms dedicated network functions into software-based virtualized network functions that can operate in a common, standard execution environment, supporting a large variety of services, ranging from performance accelerations, firewalls, DPI to load balancing. A multitude of such service functions can be further chained dynamically depending on the operator’s policy, security and performance goals. Note SDN and NFV/SFC are independent of each other, although they can be used in a combined manner.

In this SDN/NFV environment, a flexible signaling protocol to create dynamically, maintain and remove control plane state is necessary. In addition, support of the placement/steering of service functions and application workloads in the network would be also desirable. Challenges for such a protocol to signal SDN/NFV control plane state include:

- Scalability vs. performance: SDN/NFV signaling operations tend to have transactional semantics which are typically heavyweight and require additional computation and communication overhead. How to enable the signaling entities to support ever-increasing requests and support different services with an existing SDN/NFV infrastructure.
- Interoperability: each SDN/NFV solution provider can have its own information model for configuring its SDN/NFV state, which makes it difficult to achieve interoperability of SDN/NFV signaling from different vendors.
- Complexity: the cost for supporting congestion control, reliability and VM live migration in the SDN/NFV environment may be heavyweight, affecting the choice of related transport mechanisms and state management mechanisms.
- Security and accountability: signaling shall be prevented from denial of service and other types of malicious attacks, and accountable for the associated resource usage.
- The ability to network diagnostics and middlebox state monitoring is more and more important.

The development of signaling for SDN/NFV network environments in different standardization fora is still ongoing. Among others, ITU-T SG-13 [26] and IRTF SDN RG (https://irtf.org/sdnrg) are currently defining signaling requirement and framework for SDN networks. Part of the job of IETF NVO3 (https://tools.ietf.org/wg/nvo3/) and SFC (https://tools.ietf.org/wg/sfc/) working groups as well as ETSI NFV ISG (http://www.etsi.org/nfv) is to specify signaling framework and protocols for SFC/NFV services. ONF (https://www.opennetworking.org) specifies the signaling protocol for the south-bound interface between the SDN controller and whitebox switches. Open source projects (e.g., OpenDaylight) are also producing community-based documents, including signaling.

While the development and adoption of SDN/NFV signaling protocols will depend on a number of factors, we hope the experience and lessons learned from the IETF NSIS protocol development may help the developers of new signaling protocols with some prospects and potentially useful options.

Acknowledgments

The author would like to thank colleagues in IETF NSIS Working Group, in particular Henning Schulzrinne, Martin Stiemerling, Hannes Tschofenig, Jukka Manner, John Loughney, Robert Hancock, Cornelia Kappler, Rüdiger Geib, Christian Dickmann, Elwyn Davies and Charles Shen...
well as the support of EU FP6 ENABLE and FP7 CleanSky projects.

References


Xiaoming Fu (M’02, SM’09) received his PhD from Tsinghua University, China in 2000, and is currently a chaired professor of computer science and head of Computer Networks Group at the University of Göttingen, Germany. His research interests include network architectures, protocols, and applications. He is an IEEE Distinguished Lecturer and an associate editor of IEEE Communications Magazine, IEEE Transactions on Network and Service Management, Elsevier Computer Networks, and Computer Communications. He was secretary (2008-2010) and vice chair (2010-2012) of IEEE Communications Society Technical Committee on Computer Communications (TCCC), and chair (2011-2013) of the Internet Technical Committee (ITC) of IEEE ComSoc and Internet Society. He is the recipient of IEEE LANNAN 2013 Best Paper Award, ACM ICN 2014 Best Paper Award and 2005 University of Göttingen Foundation Award for Exceptional Publications by Young Scholars.