

# Design of CASP - a Technology Independent Lightweight Signaling Protocol

H.Schulzrinne  
*Dept. of Computer Science*  
*Columbia University*  
*New York, USA*  
*hgs@cs.columbia.edu*

X.Fu  
*Telematics Group*  
*University of Göttingen*  
*Göttingen, Germany*  
*fu@cs.uni-goettingen.de*

C. Pampu  
*Information and Communication Mobile*  
*Siemens AG*  
*Berlin, Germany*  
*cornel.pampu@siemens.com*

C. Kappler  
*Information and Communication Mobile*  
*Siemens AG*  
*Berlin, Germany*  
*cornelia.kappler@siemens.com*

## Abstract

Existing signaling solutions are insufficient in terms of inter-domain and out-of-path signaling, mobility support and inter-working with policy and security mechanisms. The paper presents the Cross-Application Signaling Protocol (CASP) which is a general-purpose protocol for managing state information in network devices. This technology independent signaling protocol can be used for inter- and intra-domain QoS signaling, the configuration of middleboxes, for collecting measurement data and any other application where state management is required. It relies on existing transport protocols and consists of a messaging layer and a client layer. The messaging layer is application independent and is responsible for routing, session establishment and feature negotiation. In contrast to this application independent component of CASP, the client layer is the application-dependent part. As an example for a client the paper describes the QoS Resource Allocation Client for CASP and discusses requirements for extending CASP to include interdomain signaling. The discovery of next peers along the data path is handled by the Scout protocol, which is a specialized client protocol. Some of the basic mechanisms are derived from existing protocols. This way the design of this protocol relies on the experiences made in this area and is therefore one of the promising protocol candidates for the IETF NSIS WG.

## 1. Why a new signaling protocol now?

In the beginning of the Internet, all packets were equal and received the same treatment. As the Internet evolves, it is being used for an increasing number of different applications; now we know that flows originating from some applications or users need special treatment. Hence, flow-specific state needs to be established in network nodes. This implies flow-related signaling is gaining importance in the Internet. For example, signaling is necessary for installing QoS, NAT and firewall control, MPLS label distribution, collection of measurement data or VPN set-up. We define signaling as the establishment of state in nodes along the data path. There are other applications for such state-establishment protocols, such as depositing active-network code or measuring the performance of network elements (OAM).

It would be desirable to satisfy all these signaling needs with a single protocol. In fact, RSVP [RFC2205] has been extended so it can be employed for most of the above-mentioned signaling applications [TIST][RSVP-TE][RFC3175]. However, RSVP originally was designed to signal QoS for IntServ, and not for general-purpose signaling. Therefore, not surprisingly, it is not ideally suited for that purpose. Moreover, the multitude of RSVP extensions were designed in independent efforts and it is unclear whether they would collaborate in a single implementation.

For its original purpose, QoS signaling, RSVP is not widely deployed either. The reason is often believed to be excessive transport and processing overhead, although much of this could be reduced by yet other RSVP extensions [RFC2916]. For a detailed analysis see [RSVPanalysis]. However, despite the lack of current deployment in the Internet backbone, it becomes

increasingly important to be able to signal QoS, particularly in access networks. For example, UMTS, which in its core network uses the Internet protocol suite, is designed to deliver QoS-dependent real-time video and audio.

In order to advance the state of signaling, particularly regarding QoS signaling, in November 2001 the NSIS (Next Steps In Signaling) Working Group was chartered by the IETF [NSIS]. It is currently creating requirements and frameworks for a next-generation signaling protocol.

In this paper we describe a new general-purpose signaling protocol, CASP (Cross Application Signaling Protocol) [CASP] and its application to QoS signaling [CASPQoS]. Both CASP and CASP QoS have been proposed to the NSIS WG. We believe it to be superior to even an amended RSVP, because it has been designed from the beginning to accommodate general signaling applications. Particularly, as originally proposed by Braden [Braden], we define a two-layered signaling protocol. The lower message layer provides the functionality common to all signaling applications, whereas the upper client layer is specific to a particular signaling application, e.g. QoS.

The remainder of this paper is organized as follows: In Section 2 we discuss in more detail the state of the art in signaling. In Section 3, CASP is described, in Section 4, the CASP QoS client. Section 5 considers possible applicability to inter-domain QoS signaling, and Section 6 provides conclusions.

## 2. State of the Art in Signaling

Signaling protocols are typically used in connection-oriented networks to establish, modify, status query and release connections. Examples of classical signaling protocols include Signaling System 7 (SS7) [SS7] in telephony networks, and ATM Forum User Network Interface (UNI) [ATM] or ITU Q.2931 [ITU] signaling protocols in ATM networks. Recently M3UA has been developed for the transport of SS7 MTP 3 user signaling messages over IP using the Stream Control Transmission Protocol (SCTP) [RFC3331].

In the network core, signaling protocols are needed to provide packet-switched networks with a similar behavior as in circuit-switched networks. In today's Internet, a typical packet-switched network, data is transmitted without reserving any bandwidth whatsoever. If one of the links is congested, arriving packets have to wait or may even be lost in associated routers. Without a signaling protocol, the routers do not have the knowledge of what treatment should be provided to these packets, even if the packets are for real-time communication, or other special purposes.

To provide Quality of Service (QoS) in the Internet, at first S-Stream Protocol version 2 (ST2) [RFC1819] was

designed. Later, Resource reReservation Protocol (RSVP) [RFC2205] was developed and standardized in 1997, originally designed to establish and maintaining resource reservations for end-to-end real-time sessions over the Internet based on the Integrated Services architecture [RFC2210].

RSVP attracted great research and industry interest as it introduced a number of intriguing features into the signaling protocol. To meet new challenges in Internet signaling, a number of extensions of RSVP have been proposed, including RSVP Refresh Reduction extension [RFC2961], RSVP Diagnosis messages [RFC2745], extensions to be used in IP tunnels [RFC2746], 802.x networks [RFC2814] and Differentiated Services networks [RFC2996], RSVP Traffic Engineering Extensions (RSVP-TE) [RSVP-TE] in Multi-Protocol Label Switched (MPLS) [RFC3031] networks, and the extension of these protocols for Generalized MPLS (GMPLS RSVP-TE) [GMPLS] [Berger][Mannie], which supports Synchronous Optical Network (SONET), Synchronous Digital Hierarchy (SDH) and Dense Wavelength Division Multiplexed (DWDM) networks [Mannie].

However, due to a number of reasons, RSVP and its extensions, even when used for QoS resource reservation, do not meet the requirements of Internet signaling, and are difficult to be deployed in the global Internet. For example, the built-in multicast support adds considerable overhead, but very often is unnecessary [Fu02]; the limitation of signaling message size and overhead to deal with reliability also limit its use [McDonald03][Pan03]. To tackle some problems with RSVP, other protocols like YESSIR [YESSIR], Boomerang [Boomerang], BGRP [BGRP] were developed but did not raise community interest.

In order to meet the demand for a generic Internet signaling protocol, as already mentioned above, a new IETF working group, Next Steps in Signaling (NSIS), was founded to investigate the architecture and protocol design of the next (generic) signaling protocol for the Internet based on existing experiences. Specifically, NSIS is to develop the requirements, architecture and protocols for the next IETF steps on signaling QoS. Emphasis is put on evaluating whether it is possible to amend RSVPv1. Up to that point NSIS output includes requirements – shortly due for WG last call – [NSIS Req], a framework for QoS signaling [NSISframe] and the above-mentioned RSVP analysis [RSVPanalysis].

## 3. CASP – A Framework for Signaling

CASP consists of a generic transport (messaging) layer and any number of client (application) protocols. The messaging layer is responsible for delivering signaling messages from the initiator to the responder, typically the

data source and the data sink, respectively. (The initiator and responder can also be represented by proxies close by, e.g., to support end systems that do not themselves have CASP capabilities.) The client layers perform the actual signaling function, e.g., reserve resources or open firewall ports. There is also one special client, the scout protocol, described below.

A CASP message is handed from one CASP node to another. Typically, nodes are connected by a reliable transport protocol, such as TCP or SCTP. We chose a reliable transport since signaling requires many of its functions, such as reliability, congestion control, flow control and fragmentation. This may seem surprising since it is often assumed that signaling applications have low data rates, with small, infrequent packets. However, while this is often true, not all signaling applications are that well-behaved all the time. For example, authentication tokens and user authorization certificates for AAA can easily push the message size to several kilobytes. A CA-signed certificate including the principal's public key weighs in at about 5 kB, without signed data. Such large messages will likely require fragmentation and may make congestion control advisable. Also, end systems may decide to rapidly probe for available resources if the network is busy. Since CASP nodes may need to perform time-consuming AAA operations, the processing time for each request can vary, so that flow control is needed to keep a neighboring node from overwhelming it with requests.

RFC 2916 [RFC2916] introduced an acknowledgement mechanism into RSVP. This simple retry-until-acknowledged mechanism ensures that Path and Resv messages are delivered to the next hop, but it falls far short of a "modern" transport protocol. For example, it does not support RTT discovery, selective acknowledgement, windowing or duplicate acknowledgement detection. By using an existing transport protocol, CASP can benefit from the improvements in these protocols, both for general Internet use and more specialized environments, such as high-loss-rate wireless links.

Transport connections in CASP are simple reliable channels, shared between any number of CASP sessions, sequentially and in parallel. Nodes can tear down transport connections without affecting the CASP session. Sharing a transport session amortizes the connection setup overhead across many sessions and improves the round-trip-time estimate. We anticipate that the number of CASP peers for each node is likely going to be measured in the tens to hundreds; in any event, modern operating systems support thousands of simultaneous TCP or SCTP connections. The use of a reliable transport also makes it possible to use TLS for channel confidentiality and integrity.

CASP supports both soft-state and hard-state operation by using configurable timers. In soft-state mode, the CASP state is removed after a set interval unless it is refreshed. A reliable transport mechanism improves the operation of the soft-state mechanism, rather than interfering with it. For RSVP without hop-by-hop reliability, the timeout interval has to be set to a multiple of the refresh interval  $R$ , typically  $K=3$  times as long, to avoid timing out state due to lost refresh packets. With reliable transmission, this interval can be reduced to just slightly larger than the nominal refresh interval. For hard-state operation, the refresh interval is set to a very large number, so that only an explicit state removal message will get rid of the session.

A CASP session is established between the initiator and the responder, along a chain of CASP nodes. At each node, the CASP server determines the next node along the data path, checks if there is an existing transport connection to that node, or establishes one if not, and then forwards the message downstream. The node then remembers the upstream node and associates it with a session identifier. If the responder wants to send a message to the initiator, it simply marks it as a response and hands it to the nearest node. Each node uses cryptographically-random session identifier chosen by the initiator to find the upstream node. If there is no state, the message is routed based on its destination IP address. This behavior ensures that all messages for a session traverse the same set of CASP nodes, in both directions. The forward direction visits a subset of the routers along the data path, but the reverse direction may well differ from the reverse data path, due to asymmetric routing common in the Internet.

Typically, either the initiator or the responder can send messages, but we also allow any node to inject a "forward" or "backward" message, i.e., directed to either the responder or the initiator. Such mid-stream generation is useful for local repair, but does pose special security problems to make sure that only authorized entities issue such messages. The use of channel security can address some of these concerns.

Not all CASP nodes need to support all client layers. Indeed, it is likely that certain client layers will only be supported by specialized nodes. For example, a firewall control protocol may only be used by the two firewalls in the initiator and responder networks. All other nodes along the path simply forward the message, without processing it. We are currently discussing whether certain common flow information should be exposed to all nodes, so that NATs, for example, can inspect and modify the information.

Since CASP is meant to support a variety of applications, it needs to be extensible, both at the messaging and the client layer. A CASP message consists of a sequence of message objects, possibly nested. Each

message object indicates whether it is mandatory or optional. A CASP node that does not understand a mandatory object rejects it with an error indication instead of forwarding it. Unknown, but not mandatory, objects are simply copied in the outgoing message. In addition, CASP supports feature discovery, so that a node can inquire about the capabilities along the path. A single capability can include one or more different message objects.

As mentioned above, CASP nodes need to discover the next hop. RSVP solves the problem by using a Path message that is addressed to the data destination and is marked by a router alert option, so that intermediate routers intercept and process the message. In RSVP, the Path message also has application-functionality, as it announces the flow characteristics to the data sink. In CASP, we strive to support a larger variety of next-hop discovery mechanisms. We distinguish between *active*, *passive* and *directory-based* discovery. For active discovery, each node sends out a UDP CASP “scout” message addressed to the CASP responder IP address, again marked with an IP router alert option. The next CASP node intercepts the scout message and responds to the IP source address. This is then used by the previous hop to establish a connection to that node. (This mode of operation will work through many, but not all, NATs.)

In passive discovery, the CASP node uses existing routing information to determine the next CASP node. In the simplest case, a node knows that the next IP forwarding hop is CASP-aware and simply picks that address. This case may be uncommon except in edge networks. Within a network, a node may have access to the intra-domain routing table if provided by a link-state protocol such as OSPF. The node can then track which other nodes in the network are CASP-aware, determine the path for the data packet with the responder’s IP address and connect to this CASP node. We are investigating whether OSPF can be extended to advertise CASP-aware nodes.

Finally, with directory-based discovery, a node consults a directory, such as DNS or LDAP, to find the next hop. This mode is particularly interesting for path-decoupled signaling, where the message does not follow the router path, but rather visits, say, only the same autonomous systems (AS) along the way. BGP paths contain the AS number of the next hop. As one technique, we proposed to use DNS SRV records. Each AS has one such DNS entry, e.g., AS 1248 would have the entry 1248.as.arpa. Each entry then identifies a number of CASP servers that can handle CASP messages, ensuring load-balancing and redundancy.

Active, passive and directory-based discovery can be combined along one CASP chain. For example, in the edge domain, the CASP nodes might use passive discovery based on next-hop routers if all routers support

CASP. Once leaving the domain, the CASP-aware edge router of the next AS may be discovered via active scouting. That edge node may then direct the request to the bandwidth broker for the domain, using directory-based discovery.

Finally, we introduce a mechanism that simplifies bidirectional state establishment. We allow an initiator to include a fully-formed CASP message in a message, addressed to the responder. The responder then takes this message and in turn becomes an initiator. This approach greatly simplifies the creation of bidirectional resource reservations or firewall setup, where the credentials of the initiator may be needed, rather than those of the responder. (Consider as an example a conferencing server (MCU) for multimedia conferences. The operator of the conference server may have contracts with a number of ISPs to provide higher-quality IP transport, but it is unlikely that each individual customer of the conferencing service has negotiated such arrangements.)<sup>1</sup>

#### 4. A QoS Resource Allocation Client for CASP

CASP is a generic signaling protocol. Two client layer protocols have been defined so far: CASP-QoS, which is described in this paper and CASP-Midcom [CASPMid], which is a “Firewall/NAT Traversal Client for CASP”. First versions have been presented to the IETF NSIS WG, prototypes are currently being implemented.

CASP-QoS is a client protocol for resource allocation and reservation [CASPQoS]. It is designed to work with arbitrary QoS provisioning techniques, to support per flow signaling for unicast as well as source-specific multicast flows [SSM], for in-path and out-of path signaling, and for sender- and receiver initiated operation.

CASP-QoS defines five message types – query, response, commit, reserve and release [CASPQoS]. They are related to the client layer only and thus transparent for the M-layer of CASP. The originator of a “*Query*” message can find out whether resources are available before they are requested and committed. The “*Response*” message is an answer to a “*Reserve*” message, where resources are assigned, or to a “*Commit*” message, where resources are allocated, and reports the result of these operations. The “*Release*” message releases all resources for a particular session.

Each CASP-QoS message may contain several objects [CASPQoS], below we describe some of them in detail. Their TLV (type-length-value) structure gives the possibility to change and add new objects without impacting the rest of the protocol.

Obvious objects that have been defined are resource specific objects [CASPQoS], among them the bandwidth

---

<sup>1</sup> This mechanism is similar to the SIP REFER request.

object. Thereby the originator can define a desired value and a minimum value, and this way specify its ability to adapt to the network resources. The intermediate CASP nodes can choose a value within this range and update the nodes along the path in the reverse direction. Also defined are a PHB object to use with DiffServ and a RSVP Rspec object to use with IntServ. Other objects, like delay, are under evaluation and could be added in the further design and implementation phase.

The Time object describes the time a resource reservation is to be effective as a start and end time. I.e. an advance reservation is defined to support resource allocation at any time in the future. This feature is attractive especially for conferencing and gives the possibility to query for available resources in the network.

The "Partial reservation" object defines how many failed reservations are allowed and records how many routers could not make reservations. In case of overprovisioning or if other QoS mechanisms are used on certain portions of the data path, this object is used to allocate resources even if not all nodes are CASP aware.

The originator of a CASP-QoS message can trigger a specific behavior of the destination using the "Next"-object. If the "Reserve" message contains for example *Next = Commit*, the receiver should allocate the reserved resources. Using the "Priority" object, depending on the local policies, a high priority request can be handled faster than normal or low priority requests, which may be queued. In the worst case a low priority reservation will be deleted in favor of a preempted reservation. The "Status" object contains information about the reserved resources and the number of failed and successful authentication and reservation requests. This object is always part of a "Response" message and is added by each CASP-QoS node. A "Version" object is used for a quick look-up of the status of a QoS message.

The CASP-QoS client has a modular structure, which allows for example to easily extend it for signaling for aggregated traffic by adding new objects, even though the QoS client protocol is currently defined only for per flow signaling.

Any node in the network can initiate and terminate a QoS signaling session. This allows limiting the QoS signaling to that part of the network where it is needed, instead of always signaling end-to-end (e2e). The signaling information thus is sent only to the relevant nodes, reducing processing overhead and avoiding excessive data transfer. This ability to restrict the signaling range is useful in a number of scenarios: reservations can be pre-established in just one domain, or between two domains, signaling proxies can be used, which allow for example in mobile communications a more efficient use of the air interface.

Also local information, like DSCP (DiffServ codepoint), accounting or operator specific information

can be sent without setting up an e2e signaling session. Additionally, for mobile hosts, partial route changes due to handover can be handled more efficiently.

The CASP-QoS signaling messages can be sent along the data path. In that case, which is called in-path signaling, the CASP-QoS aware nodes in the data chain process the signaling information and perform resource reservation. Those nodes, which do not run a CASP-QoS client, will forward the signaling information to the next CASP-QoS node. Besides in-path signaling, the resource reservation messages can be sent off-path, i.e. on a different path than the data. This allows e.g. the exchange of information between QoS brokers in different domains.

Similar to RSVP [RFC2205] CASP-QoS supports receiver-oriented reservations. In addition, sender-oriented reservations are possible, which have a reduced number of signaling messages. The direction-neutral feature is interesting especially for adaptive applications where both sides can initiate the resource reservation and can request information about resources at the other end or may initiate an independent signaling session at the other side. This works well for symmetric and asymmetric routes. The resource allocation in both directions can be done for symmetric routes with a single reservation message and is applicable in traffic engineering. In case of asymmetric routes the originator initiates a signaling session at the destination in the backward direction.

The use of a reservation range instead of fixed values reduces the signaling traffic and allows nodes along the data path to adapt itself to changes without renegotiating resources.

This and the partial reservation feature support flexible reservation along the data path, increase the probability of successful reservation and reduce the information overhead.

The CASP-QoS client protocol relies on security and reliability mechanisms of the CASP protocol. This implies an easier implementation and a faster processing of the CASP-QoS. However, resource objects and data can be additionally protected by CMS without protocol extensions.

## 5. Applicability of CASP to Inter-Domain QoS Signaling

The CASP QoS client has not yet been finalized and interdomain signaling is not yet explicitly included. Therefore in this Section we discuss requirements for the CASP-QoS client to cover inter-domain signaling.

Due to its generic approach, CASP QoS as it is defined now can already be used for exchange of signaling information between arbitrary nodes that belong to different domains. The signaling can be in-path or, e.g. between bandwidth brokers, off-path.

CASP QoS currently supports per-flow signaling between two domains. It could be easily extended for signaling for aggregated traffic and for allowing merging several reservations into a single aggregated one, whereby the e2e reservations are transparent to the aggregated region. This idea of state aggregation is adopted from [RFC3175]

QoS interdomain signaling however goes beyond the ability to per-flow or per-aggregate signal on- or off-path between domains. QoS interdomain signaling is about negotiating – which may be more than one-way signaling – and dynamically adapting SLSs between providers. Resources can be used this way more efficiently and can be assigned according to the changing environment. Therefore it is necessary to define messages that allow such negotiation and adaptation. Additionally objects need to be defined that transport SLS content. Since so far no generally accepted SLS template exists, the format of these objects must be very general and easily extendible.

## 6. Conclusion

In this paper we described CASP, a generic signaling protocol, with its client for QoS signaling, CASP-QoS. Other possible clients include NAT and firewall control, topology discovery, measurement data collection, active network instantiation and any other application where state needs to be established along a data path.

CASP has been designed to be lightweight, flexible and easily extendible. For example:

- CASP uses an existing transport protocol rather than re-implementing transport protocol features.
- The separation into a generic messaging layer and an application-specific client layer such as CASP-QoS allows adding new signaling applications easily. Thereby the client layer relies on functionality, i.e. security, reliability, neighbor discovery etc. provided by the messaging layer
- CASP messages consist of a sequence of message objects. New objects can be added at the messaging and the client layer as needed to support new functionality.

The CASP QoS Client currently is specified for per-flow, intra-domain signaling only. We briefly discussed what needs to be included to extend it for inter-domain signaling. CASP and CASP QoS have both been presented to the IETF NSIS WG, which has been chartered to explore next steps in Internet signaling.

## Acknowledgements

We would like to thank Hannes Tschofenig and Jochen Eisl for being co-developers of CASP, as well as Robert Hancock and Mehmet Ersue for providing vital input to discussions.

## References

- [ATM] ATM Forum, “ATM User-Network Interface (UNI) Signalling Specification Version 4.1”, af-sig-0061.002, Apr. 2002.
- [Berger] L. Berger, Ed., et al, “Generalized MPLS Signaling - RSVP-TE Extensions”, Internet Draft, <draft-ietf-mpls-generalized-rsvp-te-07.txt>, Sept. 2002.
- [BGRP] P. Pan, E. Hahne, and H. Schulzrinne, "BGRP: A Tree-Based Aggregation Protocol for Inter-domain Reservations", Journal of Communications and Networks, Vol. 2, No. 2, June 2000, pp. 157-167.
- [Boomerang] G. Feher, K. Nemeth, M. Maliosz, I. Cselenyi, J., Bergkvist, D. Ahlrad, T. Engborg, "Boomerang: A Simple Protocol for Resource Reservation in IP Networks", IEEE RTAS, 1999.
- [Braden] B. Braden and B. Lindell. “A Two-Level Architecture for Internet Signaling”, Internet Draft, <draft-braden-2level-signaling-01.txt>, Nov. 2002.
- [CASP] H. Schulzrinne, H. Tschofenig, X. Fu, J. Eisl. “CASP – Cross-Application Signaling Protocol”, Internet Draft, <draft-schulzrinne-nsis-casp-00.txt>, Oct. 2002
- [CASPQoS] H. Schulzrinne, H. Tschofenig, X. Fu, J. Eisl, “A Quality-of-Service Resource Allocation Client for CASP”, Internet Draft, <draft-schulzrinne-nsis-casp-qos-00.txt>, Oct. 2002
- [CASPMid] H. Tschofenig, H. Schulzrinne. “A Firewall/NAT Traversal Client for CASP”, Internet Draft, <draft-tschofenig-nsis-casp-midcom-00.txt>, Oct.2002
- [Fu02] X. Fu, C. Kappler, H. Tschofenig “Analysis of RSVP regarding Multicast”, Internet Draft, <draft-ietf-fu-rsvp-multicast-analysis-01.txt> , Oct. 2002.
- [GMPLS] L. Berger, Ed., “Generalized MPLS - Signaling Functional Description”, Internet Draft, <draft-ietf-mpls-generalized-signaling-09.txt>, Aug. 2002.
- [ITU] ITU-T, “Broadband Integrated Services Digital Network (B-ISDN)-Digital Subscriber Signaling System No. 2 (DSSx2)-User-Network Interface (UNI) Layer 3 Specification for Basic Call/Connection Control”, ITU-T Recommendation Q.2931, Sept. 1995.
- [Mannie] E. Mannie, D. Papadimitriou, Eds., “Generalized Multi-Protocol Label Switching Extensions for SONET and SDH Control”, Internet Draft,<draft-ietf-ccamp-gmpls-sonet-sdh-07.txt>, Oct. 2002.

[McDonald03] A. McDonald, et al., "Design Considerations for an NSIS Transport Layer Protocol", Internet Draft, <draft-mcdonald-nsis-ntlp-considerations-00.txt>, Jan 2003.

[NSIS] IETF Next Steps in Signaling (NSIS) working group, <http://www.ietf.org/html.charters/nsis-charter.html>

[NSISFrame] R. Hancock et al. "Next Steps in Signaling: Framework", Internet Draft, <draft-ietf-nsis-fw-01.txt>, Nov. 2002.

[NSISReq] M. Brunner(ed.), "Requirements for Signaling Protocols", Internet Draft, <draft-ietf-nsis-req-06.txt>, December 2002.

[Pan03] P. Pan et al, "An Evaluation on RSVP Transport Mechanism", Internet Draft, <draft-pan-nsis-rsvp-transport-00.txt>, Jan 2003

[RFC1819] Delgrossi, L. and Berger, L., "Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+", RFC1819, Aug 1995.

[RFC2205] Braden., R. Ed., et. al., "Resource ReSerVation Protocol (RSVP) -- Version1 Functional Specification", RFC 2205, September 1997.

[RFC2210] Wroclawski, J., "The Use of RSVP with IETF Integrated Services", RFC 2210, Sept. 1997.

[RFC2745] Terzis, A., Braden B., S. Vincent, and L. Zhang, "RSVP Diagnostic Messages", RFC 2745, Jan. 2000.

[RFC2746] Terzis, A., Krawczyk, J., Wroclawski, J. and L. Zhang, "RSVP Operation Over IP Tunnels", RFC 2746, Jan. 2000.

[RFC2814] R. Yavatkar, D. Hoffman, Y. Bernet, F. Baker, and M. Speer. "SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style networks", RFC 2814, May 2000.

[RFC2996] Bernet, Y., "Format of the RSVP DCLASS Object", RFC 2996, Nov. 2000.

[RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P. and F. Tommasi, "RSVP Refresh Reduction Extensions", RFC 2961, Apr. 2001.

[RFC3031] Rosen, E., Viswanathan, A. and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, Jan. 2001.

[RFC3175] Baker, F. et. al., "Aggregation of RSVP for IPv4 and IPv6 Reservations", RFC 3175, September 2001.

[RFC3331] K. Morneault, R. Dantu, G. Sidebottom, T. George, B. Bidulock, and J. Heitz. "Signaling System 7 (SS7) Message Transfer Part (MTP)2 - User Adaption Layer.", RFC 3331, Sept. 2002.

[RSVP-TE] Awduche, D., Berger, L., Gan, D., Li, T., Swallow, G. and V. Srinivasan, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, Dec. 2001.

[RSVPanalysis] J. Manner, X. Fu, "Analysis of Existing Quality of Service Signaling Protocols", Internet Draft, <draft-ietf-nsis-signalling-analysis-00.txt>, Oct. 2002.

[SSM] H. Holbrook and B. Cain, "Source-specific multicast for IP," Internet Draft, <draft-ietf-ssm-arch-01.txt>, Nov. 2002.

[SS7] T. Russell, "Signaling System #7, 2nd ed"., McGraw-Hill, New York, 1998.

[TIST] Shore, M., "The TIST (Topology-Insensitive Service Traversal) Protocol", <draft-shore-tist-prot-00.txt>, May 2002.

[YESSIR] Ping Pan, Henning Schulzrinne, "YESSIR: A Simple Reservation Mechanism for the Internet". In the Proceedings of NOSSDAV, Cambridge, UK, July 1998.