Signaling Approaches
Volume II, Part 2, Chapter 97

Edmundo Monteiro, Fernando Boavida, Marília Curado, Luís Cordeiro
Department of Informatics Engineering
University of Coimbra, Polo 2
3030-290 Coimbra
Portugal
Telephone +351 239 790000
Fax +351 239 701266
{edmundo, boavida, marilia, cordeiro}@dei.uc.pt

Key Words: Signaling, Quality of Service (QoS), Resource Reservation, Integrated Services, Differentiated Services

Abstract

Signaling is essential for current and next generation inter-networks. Content distribution, multimedia applications, intensive distributed computing, virtual reality, and games are examples of applications that require session and/or call management, service negotiation, guaranteed or controlled quality of service levels, and end-to-end secure channels. The provision of all of these is not possible without resorting to signaling mechanisms and protocols, even though most of them are invisible to the users.

This chapter provides an overview of the main signaling protocols and approaches in use in the current Internet. These include network technology-independent protocols such as RSVP, MPLS and NSIS, and application level control protocols such as SIP and H.323. For each of them, the origins, objectives, standardization, main characteristics and basic operation are identified and explained. The chapter ends with a case study where some of the presented protocols and approaches are put into practice.

Introduction

As IP networking evolves in order to support all kinds of applications, signaling becomes of essential part of both networks and applications, providing the means for resource management and control, security support, services announcement and initiation, and capabilities negotiation, among other functions. This chapter addresses the main signaling approaches in use in the current, highly-heterogeneous Internet.

OBJECTIVES AND ORGANIZATION

The Objectives of Signaling

Signaling has always been a controversial issue when the Internet is concerned. Ideally, signaling on the Internet should be reduced to a minimum and be performed by end-systems, in order to keep the network as simple as possible. Reality has shown that this is not possible and that some forms of signaling must be used, for several reasons.

Although the current Internet is still data-driven – as opposed to the signaling-driven nature of, for instance, the telephone network – signaling is present in virtually all its components, for network operation support, quality of service support, management and application/user support. Routing protocols, such as OSPF and BGP, can be considered operation-oriented signaling protocols, as they are indispensable to the operation of the current Internet. RSVP, MPLS and NSIS are examples of signaling protocols for the support of quality of service. SNMP and COPS are examples of management protocols. On the other hand, SIP and H.323 are examples of application/user support signaling protocols. All of them have the common objective of providing some form of control over and support of user traffic and services, thus contributing to the good operation of the network.

Usually, signaling messages follow the same path as data messages, in which case it is said that on-path signaling is being performed. However, sometimes entities that are not on the data path need to be signaled. Signaling protocols that allow the signaling of entities that are not on the data path are called off-path signaling protocols.

Organisation of the Chapter

The main on-path and off-path signaling protocols are presented in this chapter. The first section is dedicated to the Resource Reservation Protocol (RSVP), an important signaling protocol for quality of service provision in Integrated Services environments. This protocol has inspired the emerging IETF's Next Steps In Signaling (NSIS) framework, which is addressed in the second section of the chapter. Section 3 is devoted to the two main signaling protocols for voice, video and multimedia support, namely SIP and H.323. Other relevant signaling protocols and approaches are presented in Section 4, of which the Common Open Policy Service (COPS) and Multi-Protocol Label Switching (MPLS) are the most important. The chapter ends with a brief presentation of a case study – the EuQoS project – where various signaling protocols and approaches are combined in order to provide end-to-end quality of service over heterogeneous networks.

RESOURCE RESERVATION PROTOCOL

Background

Resource reservation is a preventive way to provide quality of service guarantees. By reserving resources along a given path, the flows are protected from congestion situations and get the contractually agreed service level no matter the network usage level. This is, of course, good from the user perspective but may be not as good from the network operator perspective, as resources are reserved and cannot be used by other flows even if the respective flows are not using them.

The Resource Reservation Protocol (RSVP) was originally defined in RFC 2205 (Braden et al., 1997), later updated by RFC 2750 (Herzog, 2000), and provides ways to perform resource reservation in Internet-like environments. The basic idea behind RSVP is that receivers identify their quality of service needs and send signaling messages upstream (i.e., towards the sources) that will be used by routers along the path to reserve the necessary resources. RSVP was designed in order to overcome the limitations of the Internet Stream Protocol Version 2, ST2 (Delgrossi and Berger, 1995), its predecessor protocol.

RSVP is the signaling protocol used by the Integrated Services (IntServ) QoS architecture, as specified in RFC 2210 (Wroclawski, 1997), for the support of multiple traffic types, including data traffic, audio traffic and video traffic. In IntServ, resources are reserved at network elements on an individual flow basis, through the use of RSVP. This approach supports what is called fine-grained QoS provision, as routers must be aware of the QoS needs of each of the existing flows. With this approach, the IntServ architecture provides three types of service: best effort (the Internet's default service), controlled load (similar to best effort in an un-congested network), and guaranteed service (assured bandwidth, no losses and bounded delay service).

In addition to signaling, which is based on RSVP, IntServ resorts to a set of other supporting functions to provide quality of service guarantees, namely policing, shaping, admission control, packet classification, and queue management. Figure 1 presents the general IntServ architecture.

Figure 1 about here

Although resource reservation has clear advantages in some cases, it is unfeasible in many other due to the high signaling overhead. RSVP messages pertain to individual flows and must be refreshed periodically. This leads to scalability problems and precludes the use of IntServ in core networks, where the number of flows may be extremely high.

Main Characteristics of RSVP

RSVP was designed to support both unicast and multicast operation, meaning that it supports reservations for unicast and multicast flows. Resources are reserved based on the individual requirements of multicast members. In addition, RSVP dynamically adapts itself to changing group membership and to changing routes.

RSVP has a simplex mode of operation. Reservations are made for unidirectional data flows. Bidirectional flows require two separate reservations. This is tied with the fact that resource reservations are receiver-initiated which, in turn, results from the fact that different members (receivers) of a multicast group may have different QoS requirements and, thus, may require dissimilar resources.

Another important characteristic of RSVP is that it is a soft-state protocol. IP is a stateless, connectionless-oriented protocol and, thus, one of the major concerns of the RSVP developers was to avoid connection-oriented operation. The way to do this was to cause reservations (i.e., state kept in the routers) to expire unless receivers regularly refresh them. This has some drawbacks and some advantages. On one side, it leads to higher communication overhead, as reservation messages keep being generated and send over the network. On the other hand, if a receiver leaves a multicast group or if a route changes, the reservations for the unused route expire, without the need for any further action.

A key aspect of RSVP is the aggregation of reservation requests. As reservation requests flow from receivers to sources, routers along the way can aggregated resource requests (e.g., bandwidth requests) whenever possible, thus optimizing the use of Internet resources. There are several forms of aggregating reservation requests for the same multicast group, depending on what is called the reservation styles used by the receivers. This will be explained further down in this chapter. Figure 2 illustrates the aggregation of reservation requests for a given multicast group.

Figure 2 about here

In this figure, requests from Hosts 1 and 2 are merged at Router 3. This merged reservation request is, in turn, merged with that from Host 3 at Router 1.

RSVP Data Flows

RSVP data flows are called sessions. A session is characterised by a session identification, a flow specification and a filter specification. The session identification is composed of the destination IP address (unicast or multicast), the IP user-protocol identifier (e.g., TCP, UDP) and the destination port. The flow specification (FlowSpec) determines the desired quality of service and is composed of the following sub-parameters: Service class, which identifies the requested type of service; Reservation specification, Rspec, which defines the desired QoS; and Traffic specification, Tspec, which describes the data flow. The filter specification (FilterSpec) determines the packets the reservation applies to. The filtering sub-parameters are the source address and/or the TCP/UDP source port.

Figure 3 illustrates the handling of packets in a session. First, the FilterSpec is checked. The packets that do not pass the filter get a best effort service. On the other hand, the packets that pass the filter specification get the service determined by the flow specification.

Figure 3 about here

RSVP protocol operation

The RSVP protocol relies on two basic message types: RESV and PATH.

RESV messages are originated by unicast or multicast group receivers in order to request resource reservation for packet flows from one or more sources. These messages propagate upstream through the distribution tree, and the respective resource requests can be merged when appropriate at each node (router) along the way. RESV messages create soft states within the routers of the distribution tree and they must be periodically re-issued in order to maintain the resource reservation.

On the other hand, PATH messages are generated by sources in order to announce themselves as multicast group sources. These messages are transmitted downstream through the distribution tree to all multicast destinations. Each router along the way creates a path state that indicates the reverse hop to be used for the corresponding source. Similarly, each destination host creates a path state that indicates the reverse hop to be used for the corresponding source. This path state allows hosts and routers to route RESV messages.

The RSVP protocol basic operation sequence can be described as follows:

1. A receiver joins a multicast group, by using the general group management mechanism, i.e., by sending an IGMP Join message to a multicast router in its network;
2. A source issues a PATH message to the multicast group;
3. The receiver receives the PATH message identifying the source;
4. The receiver uses the reverse path information to send RESV messages, specifying the desired QoS;
5. The RESV message propagates upstream, is merged with other RESV messages from other receivers, and is delivered to the source;
6. The source starts sending data packets;
7. The receiver starts receiving data packets.

This basic operation is illustrated in Figure 4.

Figure 4 about here

Other RSVP messages are teardown messages and error and confirmation messages. Teardown messages remove the path and reservation state without waiting for the cleanup timeout period. These messages can be generated by a receiver, or an intermediate router as the result of a state timeout. Reservation-request teardown messages delete the reservation state, travel upstream towards all senders from the point of teardown initiation, and are routed like reservation-request messages. In order to add reliability to the requests, RSVP comprises three error and confirmation messages: path-error messages, reservation-request error messages, and reservation-request acknowledgment messages.

Reservation styles

Reservation styles determine the way in which resource requirements are aggregated at each router. There are three reservation styles, determined by a combination of two options: the reservation attribute and the sender selection. The former can specify a distinct reservation for each sender or, alternatively, a shared reservation applicable to groups of senders. The latter can specify an explicit list of sources or all sources (wild-card). The combinations that lead to the three reservation styles (wild-card filter, shared-explicit filter, and fixed filter) are identified in Table 1.

Table 1 about here

In the wild-card filter reservation style, reservations are made irrespectively of the senders and are, thus, applicable to all senders of the group. This is the simplest reservation style and the one with more potential in terms of request aggregation, as any resource request is valid to all senders of the group. A wild-card filter reservation style scenario is illustrated in Figure 5. In this figure, bandwidth reservation is requested by the sources in multiples of B bandwidth units. For each link, the router simply propagates upstream the highest bandwidth request received from the downstream links.

Figure 5 about here

In the shared-explicit filter reservation style, receivers provide an explicit list of sources but the reservations may be shared by several sources. This scenario is illustrated in Figure 6.

Figure 6 about here

In the case of the fixed filter reservation style, receivers perform a distinct reservation for each sender and, thus, request can only be aggregated at routers if they pertain to the same source. This scenario is illustrated in Figure 7.

Figure 7 about here

NEXT STEPS IN SIGNALING

Background

Next Steps In Signaling (NSIS) is a signaling framework being developed by the Internet Engineering Task Force (IETF) in the context of the NSIS Working Group (Hancock et al., 2005), for the purpose of installing and maintaining flow state in the network. NSIS is based on various signaling protocols, the main one being RSVP. The intention is to reuse RSVP mechanisms whenever possible, since these mechanisms have already been widely tested, leaving out all unnecessary complexity (e.g., multicast support). It is, thus, a simpler and more scalable approach to resource reservation, when compared to RSVP.

By using a two-layer signaling architecture, signaling transport is separated from signaling applications, which opens the way to develop several such applications, of which quality of service signaling is the first use case.

The initial requirements of NSIS include support for the independence of signaling and network control paradigms, ability to place NSIS initiators, forwarders, and responders anywhere in the network through on-path and off-path signaling, transparent signaling through the network, grouping of signaling for several micro-flows, flow aggregation, scalability, flexibility, and security.

NSIS Characteristics

Although NSIS can work on a per-flow basis, it allows flow aggregation based on the use of the DSCP field or on tunnels. Additionally, it works on a hop-by-hop basis, between NSIS-aware nodes (NSIS Entities, NE, also referred to as NSIS hops). Nodes not supporting NSIS are transparent, which means that there is no need for deployment of NSIS in every network entity. This is illustrated in Figure 8, where the end-systems and two of the routers support NSIS Entities that exchange signaling messages related to the data flow.

Figure 8 about here

NSIS allows signaling to hosts, network elements and proxies. Proxies allow the existence of NSIS-unaware hosts, by carrying out signaling on their behalf, as illustrated in Figure 9.

Figure 9 about here

NSIS supports both on-path and off-path signaling. In the case of path-coupled signaling, signaling messages are routed through NSIS entities on the data path only, although between adjacent NEs, the route taken by signaling and data might diverge. In the case of path-decoupled signaling, messages are routed to NEs which are not assumed to be on the data path, but which are aware of it. In this case, the signaling endpoints may have no relation at all with the ultimate data sender or receiver.

NSIS Architecture

As stated before, NSIS is being developed as a two-layer modular approach, comprising an NSIS transport layer and an NSIS signaling layer.

The NSIS transport layer protocol, known as General Internet Signaling Transport (GIST), is responsible for the transport of signaling messages between network entities. The signaling layer contains specific functionality of signaling applications and may comprise several NSIS signaling layer protocols, generically known as NSLPs. With this approach, illustrated in Figure 10, the transport of the signaling messages and the signaling application are separate, which allows the use of the same signaling transport protocol for the support of all signaling applications. Examples of signaling protocols are the QoS-NSLP (Manner et al., 2006) and the Network Address Translation (NAT) & Firewall (FW) NSLP (Stiemerling et al., 2006).

Figure 10 about here

GIST

The GIST layer is responsible for the transport of signalling messages. When a signalling message is ready to be sent, it is given to the GIST layer along with information about the flow it is for. It is then up to the GIST layer to get the message to the next NSIS Entity (NE) along the path, downstream in the flow direction from the source to the destination, or upstream, in the opposite direction of the flow, from the destination to the source. In the receiving NE, GIST either forwards the message directly to the next hop or, if there is an appropriate signaling application, passes it upwards for further processing. The signaling application can then generate another message to be sent via GIST. Figure 11 illustrates this behavior, showing two different signlling applications and how NEs handle the signaling messages accordingly.

Figure 11 about here

GIST allows two modes of operation, Datagram mode (D-mode) and Connection mode (C-mode). D-mode resorts to UDP to encapsulate messages and is used for small and infrequent messages. All Query messages must be sent in D-mode. The C-mode uses TCP or any other stream or message-oriented transport protocol (currently only Stream Control Transmission Protocol, SCTP, is being considered in addition to TCP), which allows GIST to provide reliability and security (for example using Transport Layer Security, TLS, over TCP) in the message transport. In addition, GIST defines a 3-way handshake connection setup between adjacent peers, composed of a Query, a Response and an optional Confirm message.

GIST was designed as a soft-state protocol to manage all the messages and associations. Each time a state is entered or updated, a timer is setup or restarted. GIST has two main state tables: Message Routing State (MRS) and Message Association State (MAS). The MRS is responsible for managing individual flows and the MAS is responsible for managing associations between individual peers. When a timer expires (if no message is received for the corresponding flow or association) the state is automatically removed from the state tables. If a state is required again, a new handshake is needed and a new association must be created.

QoS-NSLP

To be able to establish and maintain resource reservations, QoS NSLP (Manner et al., 2006) defines four messages types: RESERVE, QUERY, RESPONSE and NOTIFY. Each message contains three parts: Control information, QoS specification (QSPEC), and Policy objects.

The RESERVE message is used to create, refresh, modify and remove states. RESERVE messages are the only messages that manipulate states and if received more than once the result is the same. The QUERY message is used to probe the network without installing or changing any state. The result of a QUERY message can be used to configure subsequent messages. The RESPONSE message is used to provide feedback from previous messages. The NOTIFY message is similar to the RESPONSE message, but is an asynchronous message that can be generated without any previous message. Usually the NOTIFY messages report error conditions.

Figure 12 illustrates the use of QoS NSLP in a basic reservation scenario. This example represents a receiver-initiated reservation, where the data sender probes the network by means of a QUERY message that gathers information on the network capability. When this message arrives at the receiver, it triggers the start of the reservation. The receiver, which in this case plays the role of QoS NSLP Initiatior, QNI, starts the reservation by sending a RESERVE message to the QoS NSLP Responder, QNR. This message will install the request in the network. In all NSIS entities along the data path for which the QoS NSLP is available, the message is processed (in this case in the two QNEs) and the reservations are made if possible. After the request is installed in the QNR, the QoS-NSLP generates a RESPONSE message. This message is sent to the QNI with the result of the RESERVE message. When this message arrives at the QNI, the procedure is complete.

Figure 12 about here

NAT & Firewall NSLP

The NAT and Firewall (NATFW) NSIS Signaling Layer Protocol (Stiemerling et al., 2006) is being defined in the NSIS IETF Working Group to provide dynamic configuration of NAT and firewall devices along the data path of a specific flow. These types of devices may create obstacles to applications such as IP telephony and peer-to-peer application unless appropriate firewall rules or NAT bindings are in place.

Figure 13 illustrates a simple example of NATFW NSLP use between a sender and a receiver, with two middleboxes (NAT or firewall devices). The source host (sender) generates a NATFW NSLP signaling message and sends it to the destination host (receiver). This message follows the data path and every NSIS entity along the data path with NATFW NSLP functionality processes the message. Based on the message processing, the NATFW NSLP triggers configuration changes of the middleboxes accordingly and forwards the message to the receiver. After all middleboxes are configured for the specific flow the data flow can start.

Figure 13 about here

NATFW NSLP signaling messages contain general information (like IP address, ports, protocol) and policy rules. Policy rules are abstractions of actual network equipment policy rules. The request initiator generates the abstract policy rules and in each NATFW NSLP along the path these rules must be mapped to the particular NAT or firewall rules. This mapping is vendor- and model-dependent.

NATFW NSLP defines five message types: CREATE, RESERVE-EXTERNAL-ADDRESS (REA), TRACE, NOTIFY and RESPONSE. The CREATE message creates, changes, refreshes and deletes NATFW NSLP sessions on the data path between the sender and the destination. The REA message is forwarded from the receiver to the edge NAT to allow inbound CREATE messages to be forwarded to the receiver. This message reserves an external address (and a port number if needed) in the edge router and requests the configuration of all intermediate middleboxes (between the receiver and the edge

NAT). The TRACE message gathers information from all NATFW NSLP on the data path. The NOTIFY message is an asynchronous message used by NSLP to notify upstream peers of specific events. The RESPONSE message is a response to CREATE, REA and TRACE messages.

SIP AND H.323

With the advent of multimedia and QoS-demanding applications, the need for application signaling in the Internet arose. Applications such as IP telephony, video conferencing, multimedia conferencing and GRID computing are examples of such applications.

There are two basic types of approaches to application signaling: data-driven and signaling-driven. The former is the approach typically used on the Internet. The latter has its origin in the telecommunications community, for which separate signaling and data networks were common. With the convergence of the Internet and the telecommunication communities, both types of approaches can be found in current IP networks.

This section presents the two main application signaling approaches used nowadays: SIP, an Internet-originated approach to application signaling, and H.323, a telecommunications-originated approach. Other application signaling protocols, such as MGCP or MEGACO, will not be addressed, as they are historic only or their installed base is rapidly decreasing, in favor of SIP and/or H.323.

Session Initiation Protocol

This sub-section is dedicated to the Session Initiation Protocol, SIP. The sub-section addresses the origin, objectives, elements, addressing, message types and dialogue organization of SIP.

Origin and Objectives. SIP is a text-based application layer control protocol developed by the IETF. It is based on HTTP and, thus, it inherited the message header format from RFC 822. SIP is defined in several RFCs, of which RFC 3261 (Rosenberg et al., 2002) contains the core specification.

SIP is used for multimedia session control, namely session creation, session modification and session termination. SIP sessions consist of sets of senders, receivers and associated state. Telephone calls, multimedia conferences and distributed computer games are examples of sessions.

SIP provides the protocol mechanisms for the establishment of point-to-point and point-to-multipoint calls, caller and callee authentication, call forwarding and transfer, call forking, terminal capability negotiation, and location-independent addressing.

By itself, SIP is not enough to make the communication possible. It requires some companion protocols, namely the Session Description Protocol, SDP (Handley and Jacobson, 1998) and the Real-Time Protocol / Real-Time Control Protocol, RTP/RTCP (Schulzrinne et al., 2003). SDP is necessary for the description of multimedia sessions (e.g., media types, time characteristics), while RTP/RTCP carries the real-time multimedia data over the network.

SIP Elements. SIP relies on the use of several different types of elements, namely user agents, proxies, registrars and redirect servers. Some of these may be logical entities residing in the same system.

User agents (UA) are session endpoints. They negotiate session characteristics and they are logically divided into User Agent Clients (UAC), which send requests and receive responses, and User Agent Servers (UAS), which receive requests and send responses.

Proxy servers route session invitations according to the current location of the called party. Additionally, they perform user authentication and accounting. Session invitations may traverse several proxies until they reach the remote UA. Proxy servers may be stateless or stateful. The former are simple message forwarders, do not take care of transactions (messages are usually organised into transactions), are faster than stateful proxies, and do not perform forking (for multiparty communication) nor recursive traversal. The latter keep track of ongoing transactions, may perform forking, keep track of retransmissions and can perform call redirection.

Figure 14 illustrates a session establishment between two users from different organizations. Each organization has its own proxy server, and these are used to route the session invitation.

Figure 14 about here

Registrars are usually co-located with proxy servers. They receive registrations from users and keep track of their current location (namely username, IP address and port number). User location information is stored in a location database. Because user registrations have a limited life span, user agents must refresh their registration periodically. If a registration is not refreshed it expires and the user becomes unavailable.

Redirect servers are also usually co-located with proxy servers. They receive location requests and return a list of the current locations of users.

Addresses. SIP entities are identified through Uniform Resource Identifiers (URI), which are similar to e-mail addresses. A SIP URI consists of a username part and a domain part, using the following format: sip:username@domain. This is very convenient, as it is thus possible to use the same URI for e-mail and SIP communication, and to resort to the omni-present DNS service to find the network location of SIP users and corresponding servers.

Although SIP addressing works perfectly on the Internet, it is not adapted to other environments, such as the telephone network. With the advent of IP telephony, for which SIP is being widely used as signaling protocol, it is extremely important to find a way to map traditional telephone numbers (unique numerical addresses administered by the International Telecommunications Union according to Recommendation E.164) into DNS query strings and, consequently, into SIP addresses. This mapping is provided by ENUM (Electronic Numbering), defined in RFC 3761 (Faltstrom and Mealling, 2004).

Messages. There are two types of SIP messages: requests and responses. Requests are used to initiate actions. The main types of requests are: INVITE, ACK, OPTIONS, BYE, CANCEL and REGISTER. Responses, also named replies, are used to confirm that a request has been received and processed. They include a status code that conveys the result of the corresponding request processing. All requests must be replied, with the exception of the ACK request.

The INVITE message is the first message sent by the calling party in the call establishment cycle. It contains endpoint IDs and call ID, among other parameters. Usually, it also contains an SDP description. The ACK message indicates that the calling party has received confirmation (a 200 OK response) to an INVITE request. The OPTIONS message is sent to query the capabilities of a call agent, namely to determine which media types a remote user supports. The BYE message is used to release a call. The sending endpoint terminates the media flow and considers the call terminated. The CANCEL message cancels a request in progress. It has no effect if no requests are in progress. The REGISTER message is used by a user agent to register with a local server on start up. This message is sent to the well-known “all SIP servers” multicast address 224.0.1.75.

Responses convey a status code, suitable to be processed by machines, and a reason phrase, that is, a human-readable message. Both of them describe the result of the processing of the associated request. The defined status code ranges are identified in Table 2.

Table 2 about here

Transactions, dialogues and calls. SIP messages are normally arranged into transactions, dialogues and calls. A transaction is a sequence of one request message and all subsequent responses to that request. For a request there may be zero or more provisional responses and one or more final responses. A dialogue is a sequence of transactions that are, somehow, related and that persist for some time. Dialogues facilitate the sequencing of messages can be viewed as peer-to-peer SIP relationships between two user agents. On the other hand, a call consists of one or more dialogues and may involve multiple user agents. Each responding user agent establishes a separate dialogue with the calling party.

Figure 15 illustrates a dialogue between two user agents and the constituent transactions.

Figure 15 about here

Figure 16 shows a SIP messages exchange between a calling party and a called party, through a proxy server, for the purpose of session establishment. The figure illustrates the use of two types of requests (INVITE and ACK) and of provisional responses (100 Trying and 180 Ringing) as well as the positive final response 200 OK. After session establishment, the data streams are carried by RTP.

Figure 16 about here

H.323

This sub-section introduces the ITU-T Recommendation H.323 (ITU-T H.323, 2003), by identifying its origin and objectives, components, protocol architecture, signaling models and communication phases.

Origin and objectives. ITU-T Recommendation H.323 is part of a series of recommendations on video telephony and multimedia conferencing services. The first versions of these recommendations were developed in the mid 1990s, and addressed the support of the referred services over a variety of network technologies, namely ISDN (H.320), ATM (H.310, H.321), PSTN (H.324), Isochronous Ethernet (H.323) and LANs (H.323).

With the widespread use of LANs, IP protocols and the Internet in general, the developers quickly arrived at the conclusion that the original H.323 recommendation scope could be extended to Internet-wide deployment, and not just LANs.

Nowadays, H.323 plays an important role as application signaling technology, with a large installed base. Nevertheless, due to its complexity and overhead, it is being precluded in favor of SIP. This is also related with the fact that H.323 uses a signaling paradigm very similar to the one used in the telephone network (which is consistent with the fact that it originated in the telecommunications community), as opposed to SIP, which originated in the Internet community.

Components. Figure 17 illustrates the basic components of H.323. These are terminals, gateways, gatekeepers and multi-point control units.

Figure 17 about here

Terminals are H.323-capable endpoints, implemented in software or hardware, and with one or more identifiers (URI and/or telephone number). Gateways interconnect H.323 entities to other network and/or protocol environments. Gatekeepers are central management entities of zones. They perform access control, address resolution, call management and provide basic and supplementary services. They can also act as proxies for security, QoS or accounting purposes. Thus, in terms of functionality, H.323 gatekeepers are equivalent to SIP proxies. Multipoint Control Units (MCU) perform multipoint control, i.e., call signaling and conference control of two or more H.323 entities belonging to the same call, and multipoint processing, such as combination of media streams from individual endpoints belonging to a multipoint call.

Protocol architecture. H.323 specifies a complex protocol architecture, represented in Figure 18. This includes signaling protocols, data transport protocols and audio/video transport protocols.

Figure 18 about here

The H.225.0 RAS (Registration, Admission, Status) protocol is used for gatekeeper discovery by endpoints, endpoint registration, endpoint location, endpoint admission requests (authentication/authorization), bandwidth requests, information on the state of endpoints, and resource availability checks.

The H.225.0 Call Signaling protocol was derived from ISDN call signaling and, in fact, is a simplified version of the Q.931 protocol. It is used for conveying information on call setup intention, call setup progress, call setup success, call setup failure, call release and call status.

H.245 is used for conference control, namely for the establishment and control of two-party calls and multi-party conferences, negotiation of modes for media exchange according to terminal capabilities, and configuration of media streams. It is a complex, heavy-weight protocol, inherited from other network technologies, namely ATM and the PSTN.

Signaling models. As mentioned before, H.323 is a complex protocol. This complexity derives from its extensive functionality and modes of operation. The richness of H.323 is also apparent in the signaling models that it uses. These are:

1. Direct signaling – In this case, only H.225.0 RAS messages are routed through the gatekeeper. Other signaling messages are directly exchanged between the two endpoints.
2. Gatekeeper-routed call signaling – In this case, H.225.0 RAS and H.225.0 call signaling messages are routed through the gatekeeper. H.245 conference control messages are directly exchanged between the two endpoints.
3. Gatekeeper-routed H.245 control, RAS and call signaling - All signaling messages are routed through the gatekeeper. Only media streams are directly exchanged between the two endpoints.

Figures 19 a) b) and c) illustrate the three H.323 signaling models descrideb above.

Figures 19 a), b) and c) about here

Call setup examples. In the following, some H.323 call setup examples are presented. Call setup is only the first phase of the several phases undergone by H.323 calls. The various phases are the following:

1. Call set up - Point-to-point or multipoint call establishment between parties.
2. Initial communication and capability exchange - Role determination (master, slave), terminal capabilities determination.
3. Establishment of audio-visual communication - Opening of logical channels for the various information streams.
4. Call services - Request/establishment of additional services (e.g., bandwidth changes).
5. Call termination.

Figure 20 illustrates a call setup, according to the direct signaling model.

Figure 20 about here

Figure 21 illustrates a call setup, now according to the gatekeeper-routed call signaling model.

Figure 21 about here

In addition to the basic services, such as call establishment and termination, H.323 includes several other services, such as multipoint tightly-coupled conferences for which access control and synchronization are performed by an MCU, broadcast loosely-coupled conferencing (Mbone style), and supplementary services such as call transfer, call diversion, call hold and message waiting indication, among other.

OTHER SIGNALING PROTOCOLS AND APPROACHES

ST2

The Internet Stream Protocol Version 2, ST2 (Delgrossi and Berger, 1995) was designed for point-to-multipoint communication. Being a sender-initiated signaling protocol, the sender originates a Connect message destined to a group of receivers. Each intermediate node makes the corresponding reservations. Once the message reaches the receiver, an Accept or a Refuse message must be sent back to the sender. The Accept messages can convey a reduction in the resources requested by the Connect message. Because it is a sender-initiated protocol, it does not scale well with an increase in the number of multicast group receivers. This protocol also does not manage states and does not provide security. ST2 is an on-path signaling protocol and a predecessor of RSVP.

YESSIR and Boomerang

YESSIR (Pan and Schulzrinne, 1998) and Boomerang, the latter described in (Manner and Fu, 2005), were designed as an attempt to simplify RSVP. YESSIR is an extension to the Real Time Transport Control Protocol, RTCP (Schulzrinne et al., 2003) that reduces the processing overhead to a minimum, uses soft state to maintain reservation states, supports shared reservations and is able to merge different flow reservations. Boomerang has only one message type and a single signaling loop for reservation setup and teardown, and it has no requirements on the far-end node. The flow initiator is responsible for all of the protocol management. This protocol supports sender- and receiver-initiated requests.

None of these on-path signaling protocols registered a significant deployment base due to their limitations. Being an extension of RTCP, YESSIR requires support from applications and routers need to unpack packets and analyze the RTCP data. Boomerang is so light that misses important functionality such as security and multicast..

SIBBS

The Simple Inter-domain Bandwidth Broker Protocol, SIBBS, was defined by the QBone Signaling Design Team, available at http://qbone.internet2.edu/bb/ (date of access: January 2, 2007), for use in DiffServ bandwidth-broker-based domains. SIBBS is a very simple protocol to be used between Bandwidth Brokers (BB) that resorts to TCP for reliability. It comprises two main protocol data units (PDU): Resource Allocation Request (RAR) and Resource Allocation Answer (RAA).

The RAR message includes a globally well-known service identifier, information related to the QoS request (classes of service and bandwidth), a destination IP address, a source IP address, an authentication field and other parameters of the service. The sender can be a client host, a BB or a proxy. The RAA message contains the answer to an RAR PDU.

When a bandwidth broker receives an RAR message, it checks its authenticity, determines the egress router (interface) from its (inter-domain) routing tables, checks that the requested resources fall within the Service Level Specification (SLS), ensures that there are sufficient resources within the domain to support the flow, and determines whether or not the flow can be accepted according to the policies of the domain. If all these constraints are met, the request is propagated recursively through the inter-domain path to the last BB. This last BB returns an RAA message to its immediately upstream BB and the process continues until the originating BB is reached. Resources are confirmed by means of periodically sent refresh messages.

In order to configure border routers, bandwidth brokers must have access to them. SIBBS does not specify a particular protocol for this purpose, but it can use protocols such as DIAMETER (Calhoun et al., 2003), Simple Network Management Protocol (Case et al., 1990) or Common Open Policy Service (Durham et al., 2000).

COPS

The Common Open Policy Service, COPS is a client/server protocol designed for policy based network management. The basic model of COPS is presented in Figure 22.

Figure 22 about here

The Policy Decision Point (PDP) is a central management entity, in charge of taking policy decisions. The Policy Enforcement Point (PEP) is the point where the policies are applied, such as a router. The optional Local Policy Decision Point (LPDP) can be used by the device to make local policy decisions in the absence of a PDP.

COPS is a request/response protocol that allows a PEP (router) to interrogate its PDP about the action to perform once an event has occurred (for instance, if a signaling message arrived).

COPS-SLS (Nguyen et al., 2002) is an extension to the COPS protocol for SLS management in a multi-domain environment. COPS-SLS basically has the same behavior as SIBBS: a request is propagated from one BB to the other in each domain on the data path. Each BB has a dual role: PDP role for its domain, and PEP role for the next domain BB.

Compared to SIBBS, COPS-SLS adds some features to the protocol, such as renegotiation of classes of service in case of failure of admission control. The communication between BB and border routers is assured by the COPS-PR protocol (Chan et al., 2001). The discovery of bandwidth brokers and border routers is outside the scope of COPS-SLS.

MPLS

Multi-Protocol Label Switching, MPLS, originated in the mid 1990s, in order to explore the performance of ATM switches in IP environments, and is now standardized by the IETF in RFC 3031 (Rosen et al., 2001). Some of its predecessor protocols are IP switching (developed by Ipsilon), Tag switching (developed by Cisco Systems) and Aggregate route-based IP switching (developed by IBM).

Although, strictly speaking, MPLS is not a signaling protocol, it is an alternative approach to QoS-oriented signaling and, thus, its inclusion in this section. The main objective of MPLS is the support of quality of service through the use of pre-defined paths and adequate resource allocation.

Main characteristics. MPLS provides connection-oriented-equivalent QoS support by efficiently routing packets through pre-established paths, to which resources can be allocated based on service-level agreements and service-level specifications (SLA/SLS). By labeling packets and associating them with given paths, MPLS leads to a reduction in the amount of per-packet processing at each router. In addition, it eases traffic engineering, optimizes network utilization and provides flow-based dynamic routing.

Basic operation. MPLS-enabled routers (label switched routers, LSR) switch and route packets based on a label appended to each packet. This is clearly simpler and quicker than IP routing. Labels define a unicast or multicast flow of packets, called Forwarding Equivalent Classes (FEC). Each FEC is allocated to a specific path, composed of a sequence of LSRs. There is a traffic characterization for each FEC, defining the QoS requirements for the flow. A particular per-hop behavior can be defined at each LSR for a given FEC.

Prior to the beginning of a flow, a Label Switched Path (LSP) must be defined, either manually or by using an interior routing protocol (e.g., OSPF). Labels must be assigned for the various links that make up the path. Labels have local significance: incoming packets with a given label are, in the general case, forwarded to the next link with a different label.

Figure 23 illustrates the basic operation of MPLS. In this figure, two FECs are shown.

Figure 23 about here

Labels. MPLS is said to be a Layer 2.5 technology because MPLS protocol control information (i.e., labels) are put between layer 2 and layer 3 headers. MPLS labels are stackable. A packet may carry an unlimited number of labels, organized as a last-in-first-out stack. MPLS processing at LSRs is always based on the top label. Label stacking allows aggregation of LSPs into a single LSPs for portions of the route.

Each LSR in an LSP must assign a label to the LSP so that packets belonging to the corresponding FEC can be recognized, inform upstream LSRs of the label assigned by this LSR to this FEC, and determine the next hop for the FEC and learn the label that the downstream LSR has assigned to this FEC.

Labels can be manually configured at each LSR or distributed using a label distribution protocol. Label distribution protocols establish a correspondence between FECs and LSPs and inform LSRs of label/FEC bindings. Additionally, they negotiate LSR forms of interaction and advertise/negotiate MPLS capabilities.

Label distribution protocols are either extensions to existing protocols, e.g. extensions to BGP (Rekhter and Rosen, 2001) or to RSVP, or specially developed protocols, such as the Label Distribution Protocol, LDP (Andersson et al., 2001) or the Constraint Routed LDP, CR-LDP (Jamoussi et al., 2002).

CASE STUDY – EUQOS

The EuQoS Project

The EuQoS Project (http://www.euqos.org) is a European IST FP6 Integrated Project, involving 24 partners (5 Network Providers, 5 Corporations, 5 SMEs and 9 Research Institutes). The motivation for the project is the increasing shift to the Internet

Protocol in the vast majority of networks and the desire of the telecommunications service providers to offer new added-value services to their customers. In this context, there is the need to coordinate the delivery of end-to-end quality of service, so that providers may offer services that can respond to the needs of their customers' applications.

The general objective of the EuQoS project was to solve design issues associated with the delivery of end to end QoS service across heterogeneous networks. The main result of the project was the so-called 'EuQoS system', which supports the delivery of end to end quality of service, and encompasses network technologies, protocols, applications and management.

End-to-end QoS provision is achieved not only by end-to-end mechanisms and functions at the application layer but also at network-technology-independent and at network-technology-dependent layers, requiring the cooperation of a range of system features and functions that reside on end systems and on servers/network elements.

The Role of Signaling in EuQoS

End-to-end quality of service is central to the objectives of the EuQoS project. The EuQoS system is totally oriented to this goal, and in the scope of the project several mechanisms, components and approaches were developed in order to achieve it.

The basic mechanisms for end-to-end QoS provision are signaling and resource reservation. Signaling is carried out at various levels: end-to-end, at application level, through the use of SIP; hop-by-hop, at inter-domain level, through the use of NSIS, complemented by routing and/or traffic engineering; locally, at intra-domain level, for resource management and provision, through the use of COPS and technology-dependent mechanisms.

The EuQoS architecture is made up of a series of components that cooperate in order to achieve the end-to-end QoS provision goal. These components, called features and functions, make use of the mechanisms referred above in order to negotiate, manage and provide the required quality of service on an end-to-end basis. For example, the Signaling and Service Negotiation (SSN) function coordinates the use of the various signaling mechanisms (end-to-end, hop-by-hop, local) in order to support the establishment of QoS-enabled end-to-end sessions between user applications.

In the scope of EuQoS, two basic approaches for end-to-end quality of service provision are being proposed, developed and studied. These are called hard-model and loose-model approaches. The hard-model approach relies on inter-domain MPLS-based traffic engineering in order to accommodate traffic demands. This model is more appropriate for situations where traffic needs are reasonably predictable and quite stable over time. The loose-model relies on the use of BGP routing, allowing for more flexible traffic support and being subject to possible resource availability constraints.

It can thus be said that EuQoS addresses the end-to-end QoS problem in several dimensions, using a comprehensive methodology that explores complementarities and cooperation: signaling mechanisms at various levels, system components cooperation, and inter-domain routing and traffic engineering.

Signaling and Service Negotiation

The main purpose of SSN is to provide the means for end-to-end session establishment according to user requirements. This requires a variety of signaling protocols in the various levels that make up the EuQoS control plane.

In EuQoS, QoS provision is achieved by complex resource management and resource allocation, involving inter-domain and intra-domain signaling interactions. There are four different “signalling levels” in the EuQoS architecture, as illustrated in Figure 24.

Figure 24 about here

The first signaling level pertains to applications. In order to establish, maintain and release sessions with the required QoS levels, it is necessary that applications express their needs and interact with the communication system, with the objectives of guaranteeing that these needs will be fulfilled, of adapting to network conditions whenever the available resources do not allow the initially requested level of QoS to be met, and of releasing the resources when sessions end. Application signaling uses the SIP protocol. SDP is also be used to support session characterization.

To enable interactions between Level 1 signalling functions and Resource Managers (RMs), vertical interactions are needed. EuQoS-aware application proxies interact with RMs for technology-independent connection admission control and trigger Level 2 signalling for resource reservation. These interactions can be bidirectional and RMs can also notify proxies of changes in network conditions.

The main objective of the Level 2 signalling functions is the support of resource reservation and management along the data path across the various network domains. To enable domain administrative independence, Level 2 signalling functions operate on a hop-by-hop basis, between Resource Managers of adjacent administrative domains.

Level 2 signalling in EuQoS is based on NSIS. The vertical interactions between RMs and resource allocators (RAs) are supported on the COPS protocol.

The third level of signaling in the EuQoS architecture is the network technology-dependent (NTD) hop-by-hop inter-domain signaling. NTD hop-by-hop inter-domain signaling is commonly carried out by BGP (Border Gateway Protocol), which provides mechanisms for inter-domain traffic routing and enables the use of routing policies to control the exchange of routing information between different administrative domains (Autonomous Systems, ASs).

The definition of signaling mechanisms to be used at intra-domain level (the fourth signaling level) in the network technology-dependent layers is also in the scope of the EuQoS project. The knowledge about specific network-dependent signaling mechanisms needs to be included in the domain's Resource Allocator in order to enable intra-domain resource management and QoS control.

Examples of intra-domain NTD signaling that can be used by RAs to configure network resources include RSVP and specific access network mechanisms like ADSL bandwidth management, 3GPP, and Ethernet and Wi-Fi priorities. NSIS path-coupled can also be used for this level of signaling.

CONCLUSION

In order to avoid complexity inside the network, the Internet has been developed according to a data-driven approach, i.e., avoiding signaling as much as possible. Nevertheless, even the core Internet protocols, such as IP and TCP, do not work without some form of signaling functionality. ICMP is an IP-companion protocol that can be considered a signaling protocol. Many of the control mechanisms of TCP are signaling mechanisms.

Nowadays the Internet is used for the support of countless applications, with varying needs and characteristics. Internet elements must be managed, data flows must be secured and quality of service must be provided to applications according to service-level agreements. All this must be provided in a highly heterogeneous environment, as dynamically as possible. In this environment, the original paradigm of keeping the network simple and free from signaling is not possible anymore.

Signaling is indispensable for quality of service provision because QoS requires control over the existing resources. It is also indispensable for security, having an impact on NAT boxes and firewalls. With the advent of Voice over IP and of multimedia conferencing applications, in multi-point environments, signaling has become a key element of the current Internet.

In this chapter we have looked into the main signaling approaches and solutions for signaling. RSVP was developed as the supporting protocol for the Integrated Services architecture for QoS provision. The Differentiated Service QoS architecture also requires some form of signaling, and this has also been dealt with in this chapter. Due to the growing need for signaling, the IETF is developing a comprehensive signaling framework, called NSIS, that has the potential to support the signaling needs of an unrestricted number of application areas, of which QoS and NAT/Firewall are the first cases. On the other hand, for voice, video and multimedia applications, H.323 and, especially, SIP/SDP are the main solutions to session-oriented signaling. The future of signaling on the Internet will certainly include all of these.

REFERENCES

Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S. (1997). RFC 2205 – Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification, Internet Engineering Task Force.

Herzog, S. (2000). RFC 2750 – RSVP Extensions for Policy Control, Internet Engineering Task Force.

Delgrossi, L., Berger, L., Eds. (1995). RFC 1819 – Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+, Internet Engineering Task Force.

Wroclawski, J. (1997). RFC 2210 – The Use of RSVP with IETF Integrated Services, Internet Engineering Task Force.

Hancock, R., Karagiannis, G., Loughney, J., Van den Bosch, S. (2005). RFC 4080 – Next Steps in Signaling (NSIS): Framework, Internet Engineering Task Force.

Manner, J. (ed.), Karagiannis, G., McDonald, A. (2006). NSLP for Quality-of-Service Signaling, <draft-ietf-nsis-qos-nslp-12.txt>, Internet Engineering Task Force.

Stiemerling, M., Tschofenig, H., Aoun, C., Davies, E. (2006). NAT/Firewall NSIS Signaling Layer Protocol (NSLP), <draft-ietf-nsis-nslp-natfw-13.txt>, Internet Engineering Task Force.

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E. (2002). RFC 3261 - SIP: Session Initiation Protocol, Internet Engineering Task Force.

Handley, M., Jacobson, V. (1998). RFC 2327 - SDP: Session Description Protocol, Internet Engineering Task Force.

Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V. (2003). RFC 3550 – RTP: A Transport Protocol for Real-Time Applications, Internet Engineering Task Force.

Faltstrom, P., Mealling, M. (2004). RFC 3761 – The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM), Internet Engineering Task Force.

Pan, P., Schulzrinne, H. (1999). YESSIR: A Simple Reservation Mechanism for the Internet", Computer Communication Review, Vol. 29, No. 2, ACM SIGCOMM.

Manner, J., Fu, X. (2005). RFC 4094 – Analysis of Existing Quality-of-Service Signaling Protocols, Internet Engineering Task Force.

Calhoun, P., Loughney, J., Guttman, E., Zorn, G., Arkko, J. (2003). RFC 3588 – Diameter Base Protocol, Internet Engineering Task Force.

Case, J. D., Fedor, M., Schoffstall, M. L., Davin, J. (1990). RFC 1157 – Simple Network Management Protocol (SNMP), Internet Engineering Task Force.

Durham, D., Ed., Boyle, J., Cohen, R., Herzog, S., Rajan, R., Sastry, A. (2000). RFC 2748 – The COPS (Common Open Policy Service) Protocol, Internet Engineering Task Force.

Nguyen, T., Boukhatem, N., Doudane, Y., Pujolle, G. (2002). COPS-SLS: a Service Level Negotiation Protocol for the Internet, IEEE Communication Magazine, Vol. 40, No. 5.

Chan, K., Seligson, J., Durham, D., Gai, S., McCloghrie, K., Herzog, S., Reichmeyer, F., Yavatkar, R., Smith, A. (2001). RFC 3084 – COPS Usage for Policy Provisioning (COPS-PR), Internet Engineering Task Force.

Rosen, E., Viswanathan, A., Callon, R. (2001). RFC 3031 – Multiprotocol Label Switching Architecture, Internet Engineering Task Force.

Rekhter, Y., Rosen, E. (2001). RFC 3107 – Carrying Label Information in BGP-4, Internet Engineering Task Force.

Andersson, L., Doolan, P., Feldman, N., Fredette, A., Thomas, B. (2001). RFC 3036 – LDP Specification, Internet Engineering Task Force.

Jamoussi, B., Ed., Andersson, L., Callon, R., Dantu, R., Wu, L., Doolan, P., Worster, T., Feldman, N., Fredette, A., Girish, M., Gray, E., Heinanen, J., Kilty, T., Malis, A. (2002). RFC 3212 – Constraint-Based LSP Setup using LDP, Internet Engineering Task Force.

ITU-T H.323 (2003). Series H: Audiovisual and Multimedia Systems, International Telecommunications Union.

FURTHER READING

Brandl, M., et al (2004). IP Telephony Cookbook, TERENA Report.

Chowdhury, D. (2000). High-Speed LAN Technology Handbook, Springer-Verlag.

Douskalis, B. (2000). IP Telephony, Prentice Hall.

Hassan, M., Jain, R. (2004). High Performance TCP/IP Networking – Concepts, Issues and Solutions, Pearson Prentice Hall.

Johnston, A., Donovan, S., Sparks, R., Cunningham, C., Summers, K. (2003). RFC 3665 - Session Initiation Protocol (SIP) Basic Call Flow Examples, Internet Engineering Task Force.

Stallings, W. (2002). High-Speed Networks and Internets – Performance and Quality of Service, 2nd Edition, Prentice Hall.

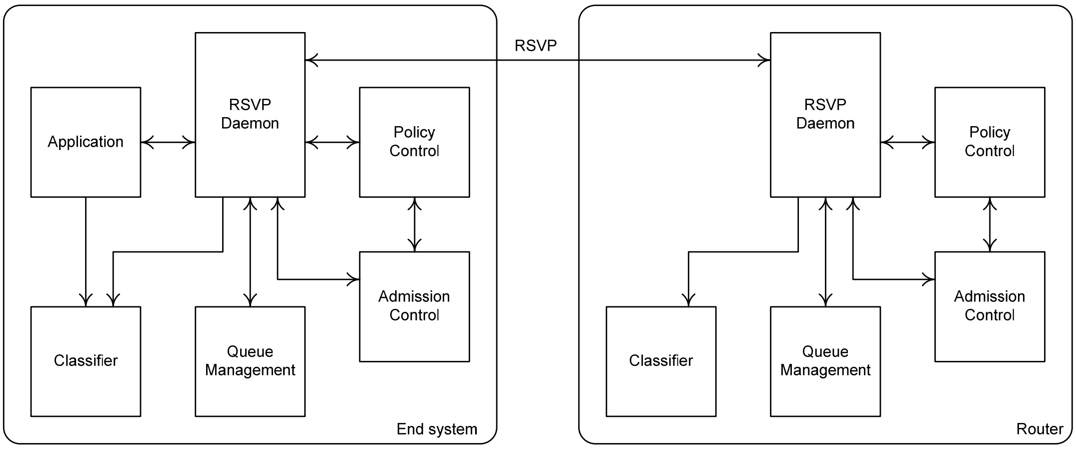


Figure 1 – IntServ architecture

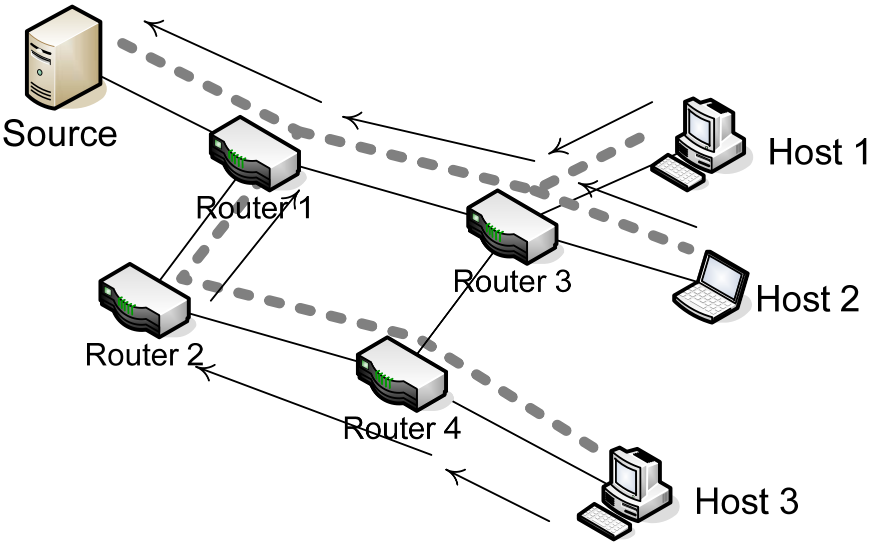


Figure 2 – Aggregation of reservation requests

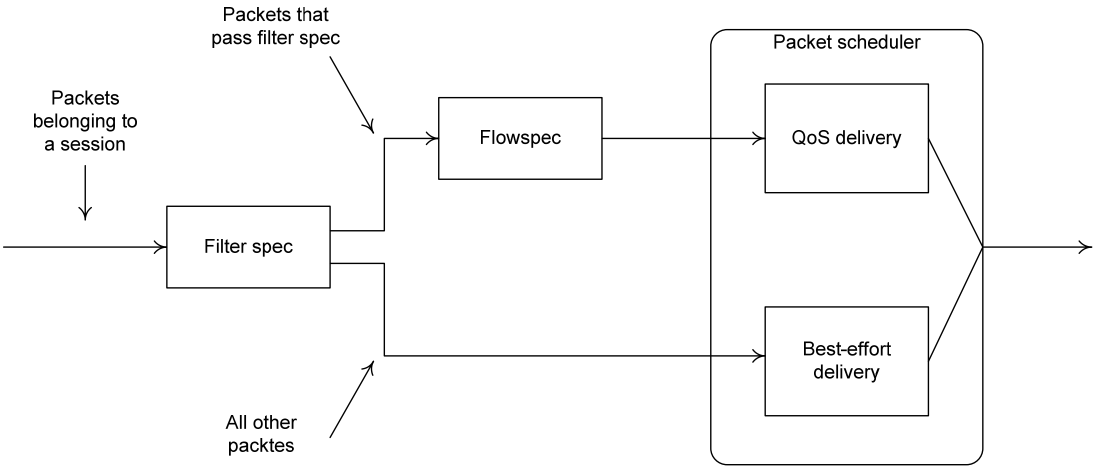


Figure 3 – Handling of packets in an RSVP session

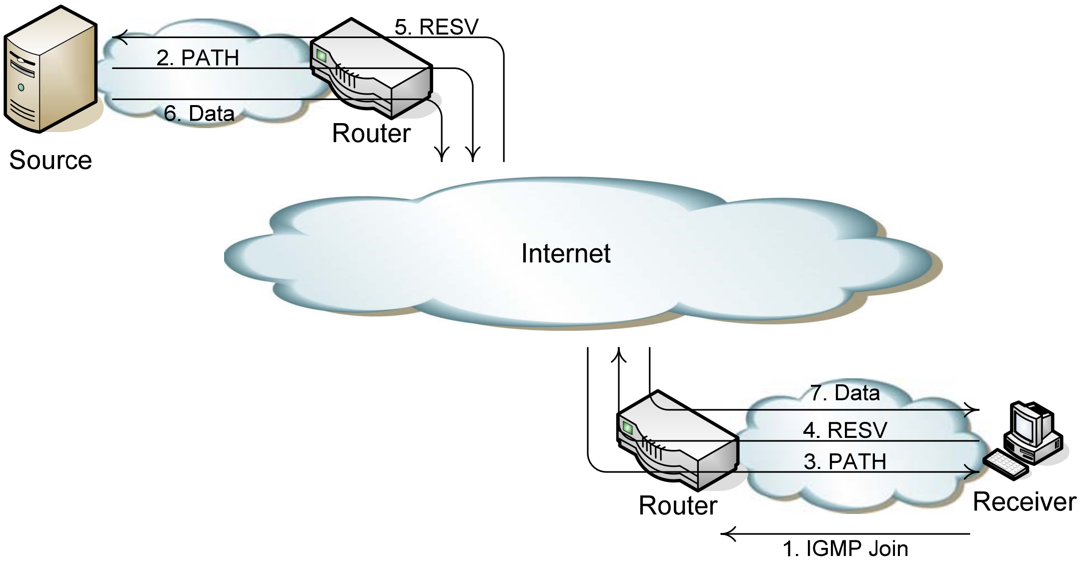


Figure 4 – RSVP protocol operation sequence

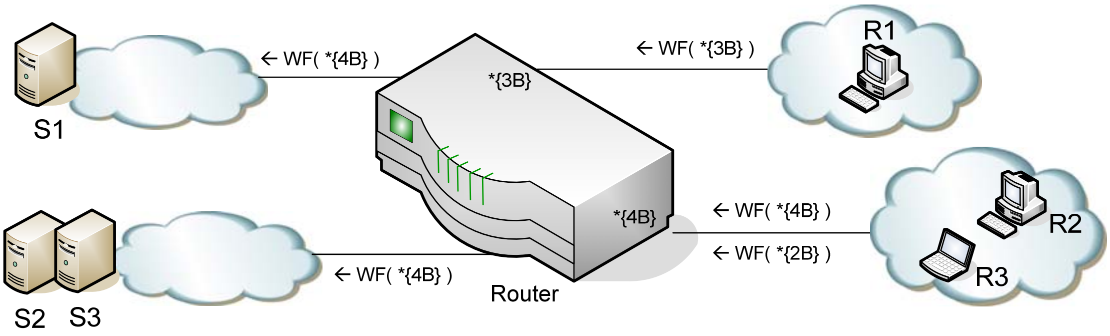


Figure 5 – Wild-card filter reservation style

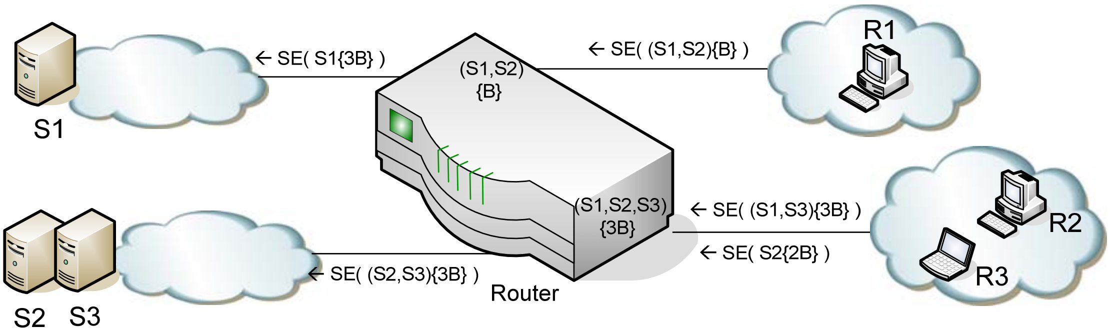


Figure 6 – Shared-explicit filter reservation style

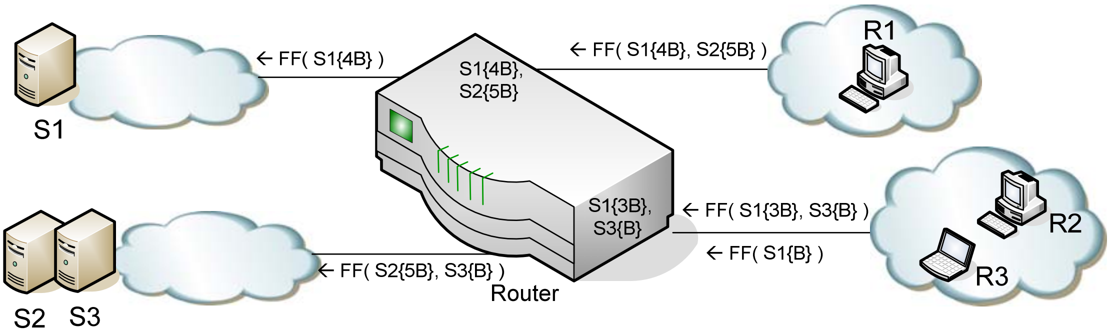


Figure 7 – Fixed filter reservation style

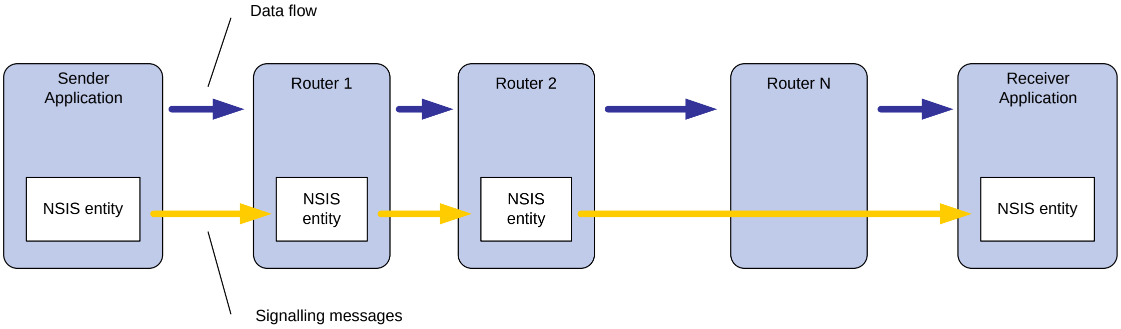


Figure 8 – Simple NSIS scenario

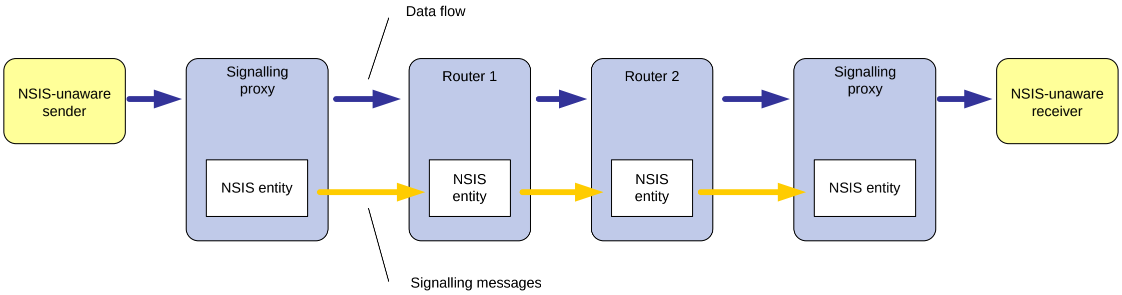


Figure 9 – Signaling proxies

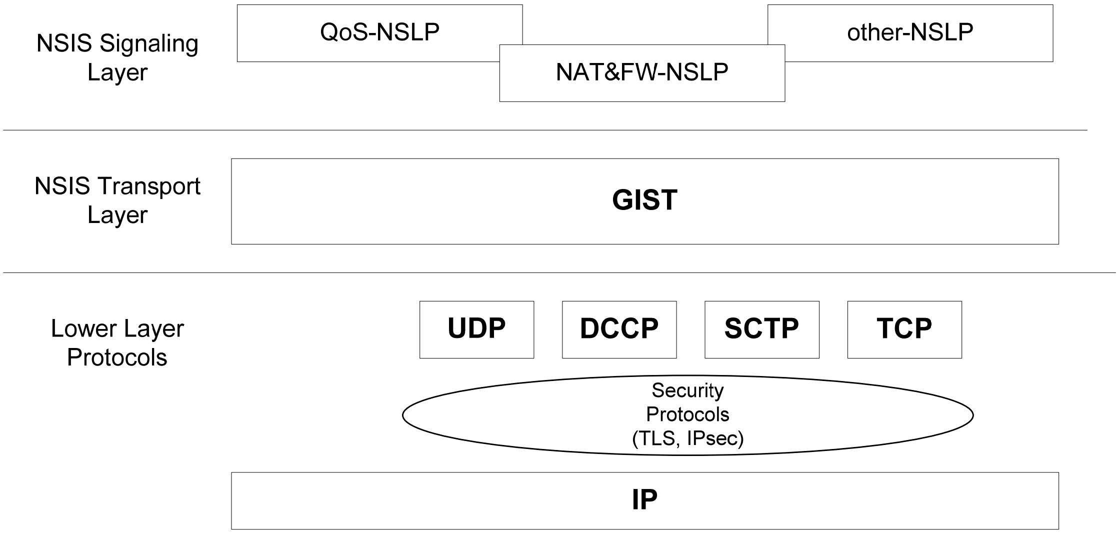


Figure 10 – NSIS two-layer architecture

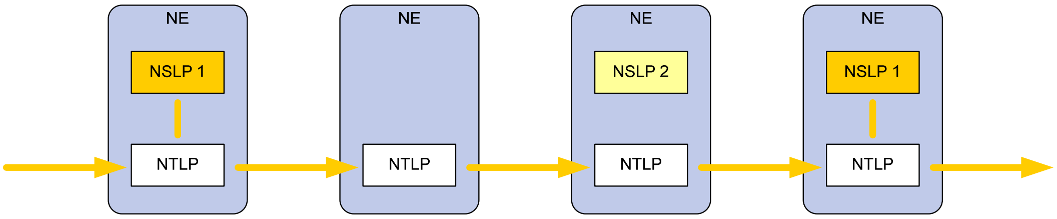


Figure 11 – Example of GIST transport between NSIS entities

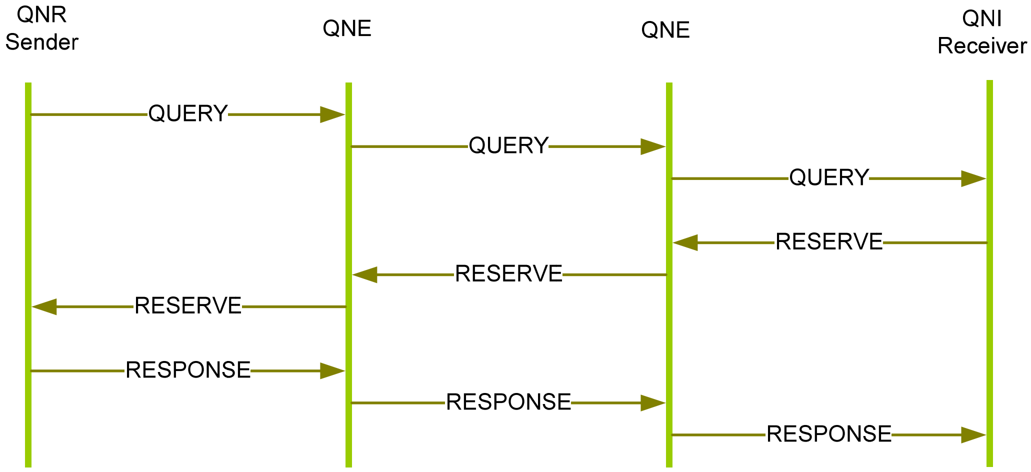


Figure 12 – QoS NSLP basic reservation scenario

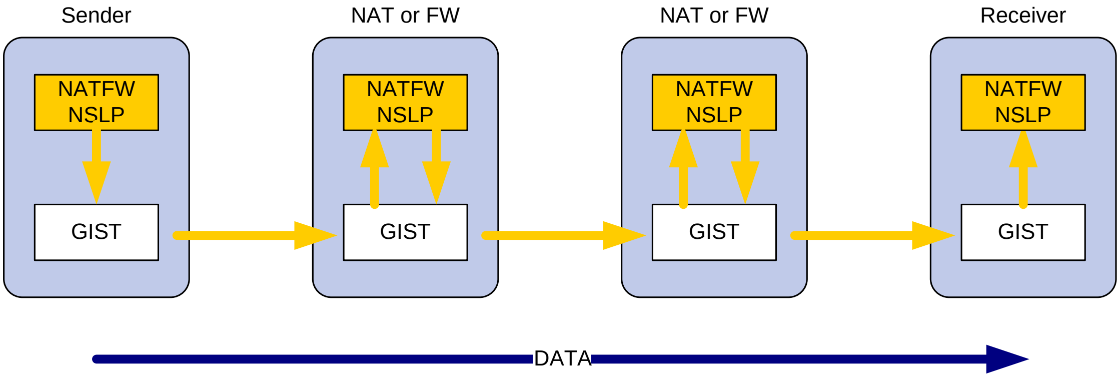


Figure 13 – NATFW NSLP scenario

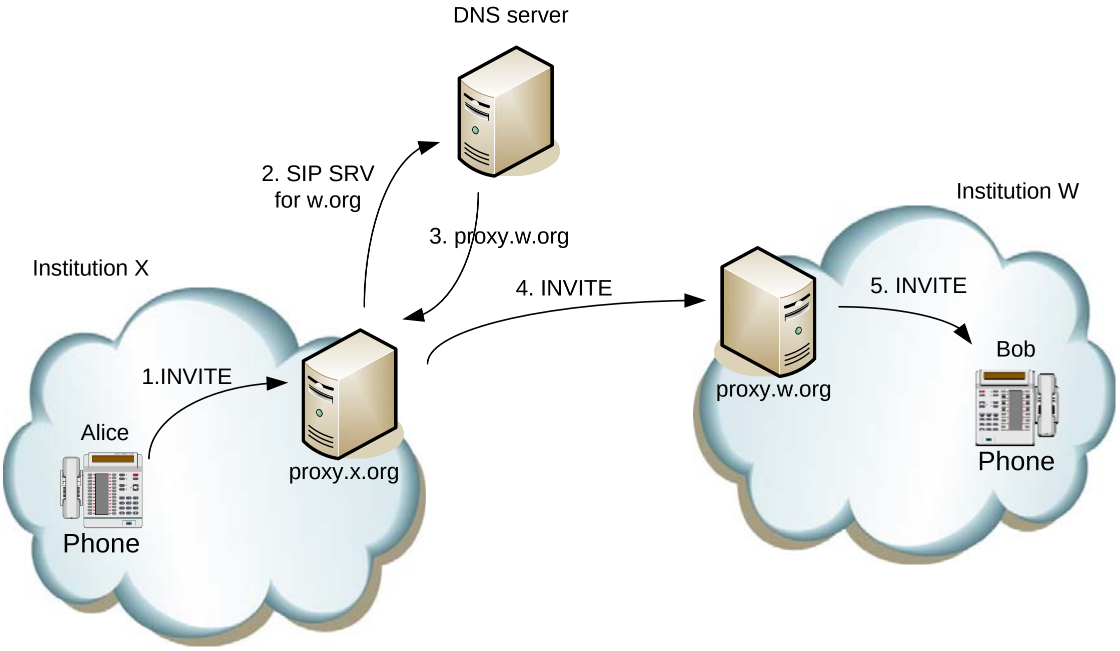


Figure 14 – Example of SIP session establishment

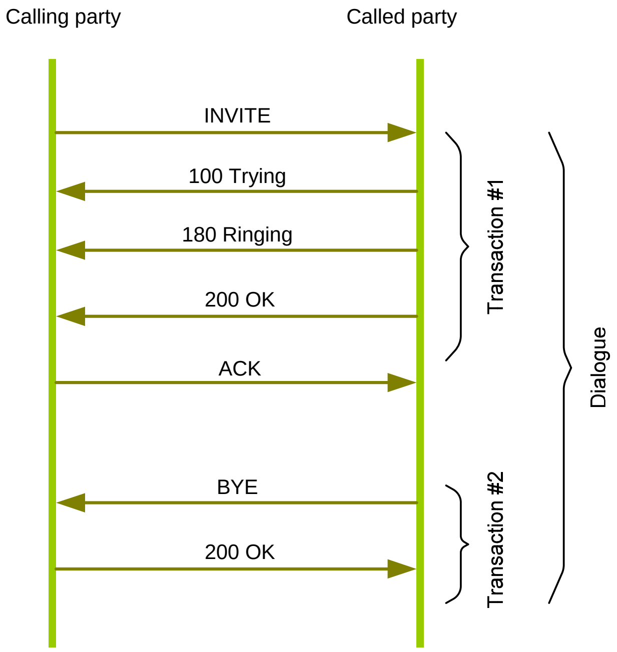


Figure 15 – SIP dialogue

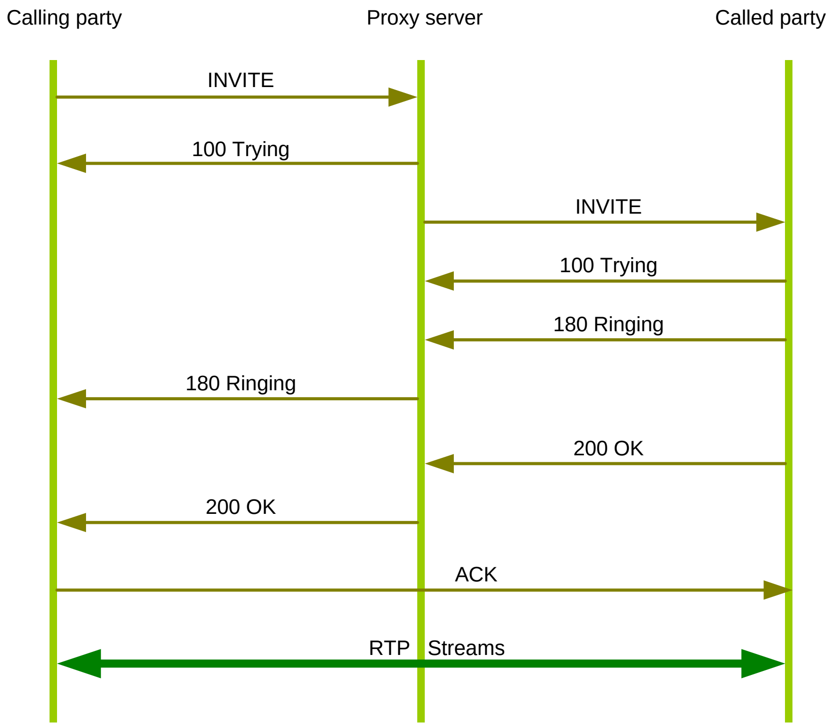


Figure 16 – SIP invitation example

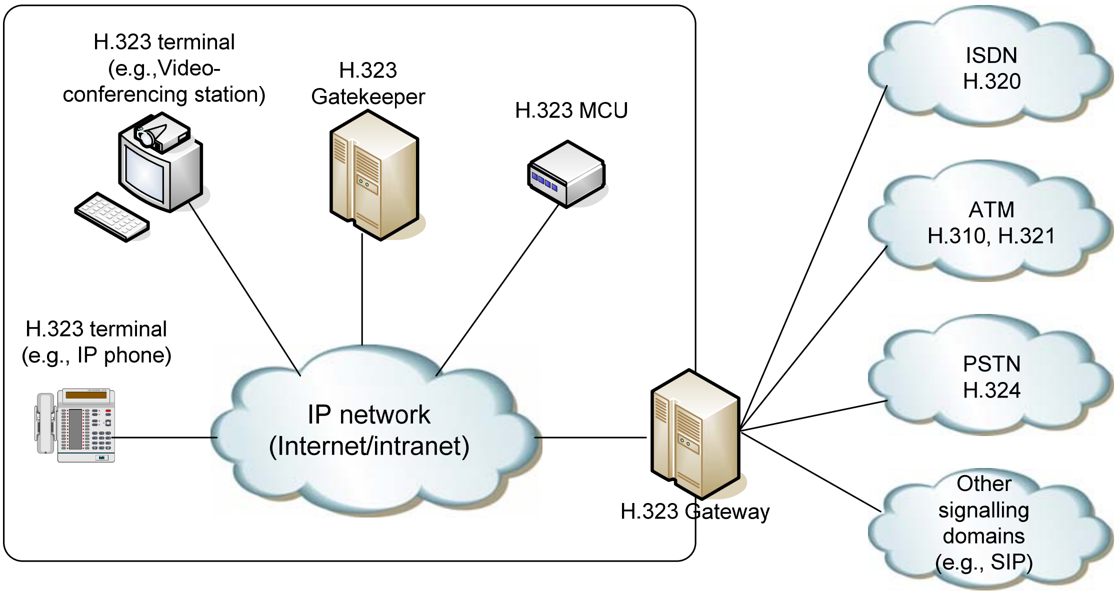


Figure 17 – H.323 components

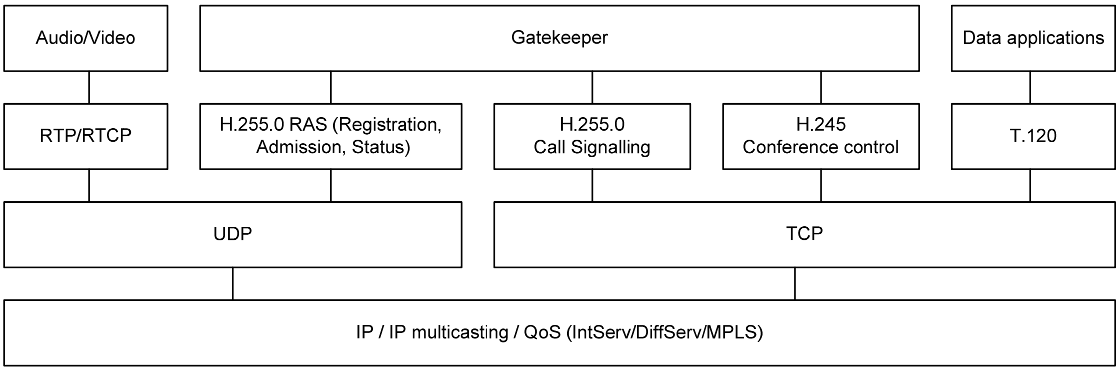


Figure 18 – H.323 protocol architecture

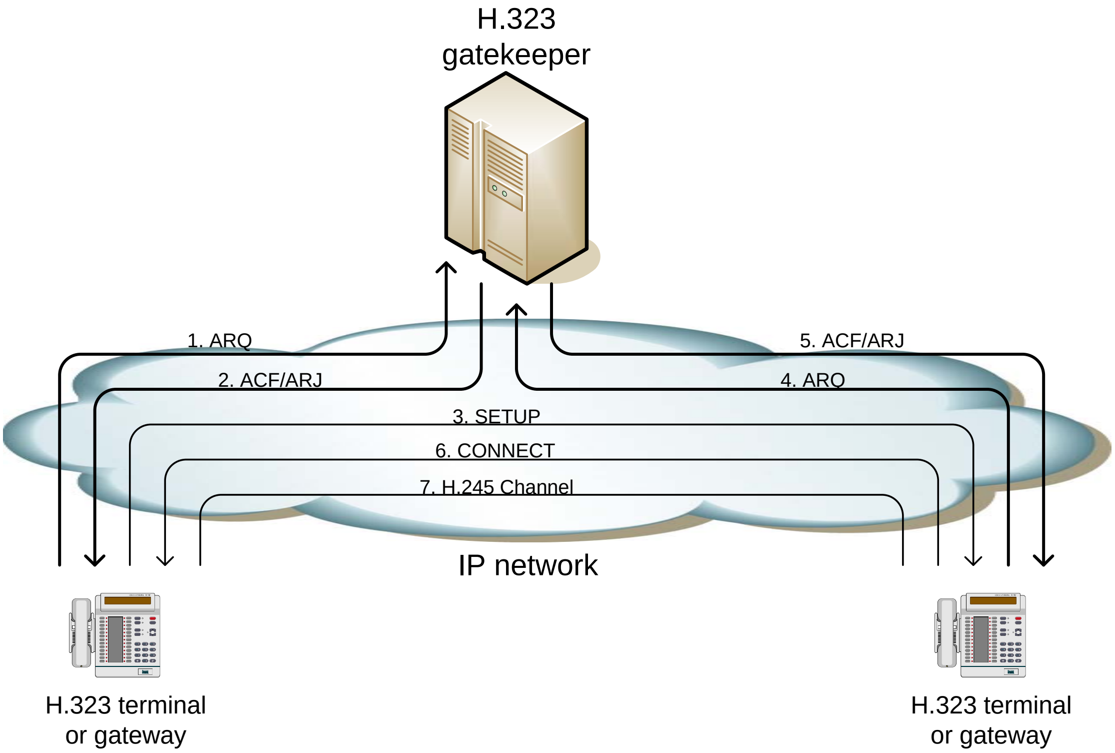


Figure 19a) Direct signaling

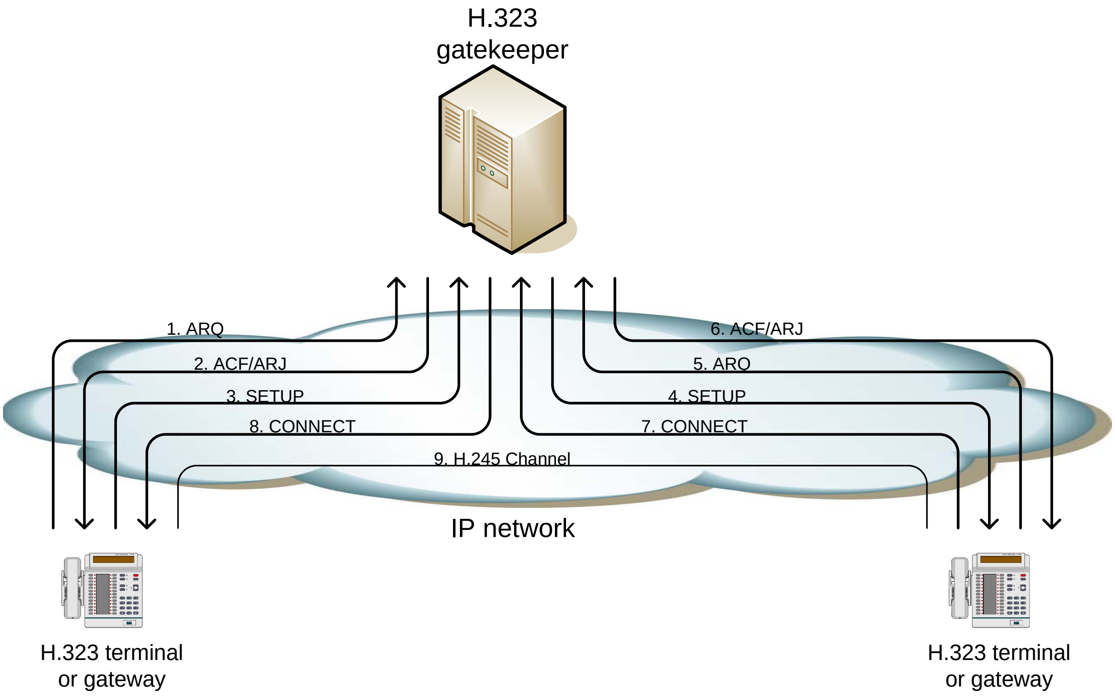


Figure 19b) Gatekeeper-routed call signaling

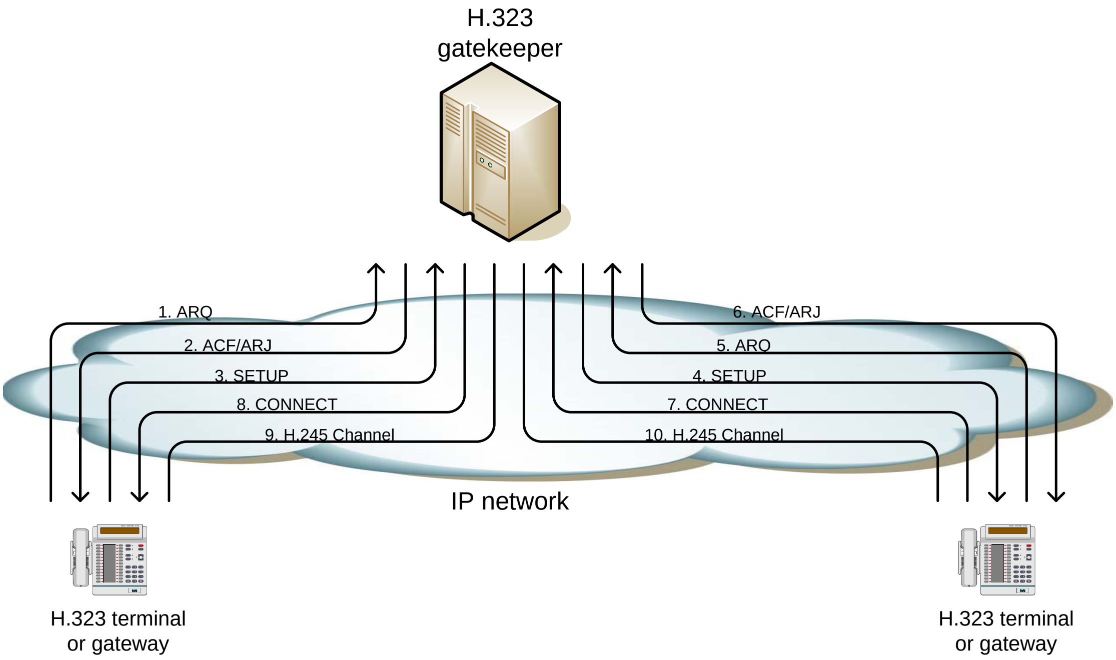


Figure 19c) Gatekeeper-routed H.245 control, RAS and call signaling

Figure 19 – H.323 signaling models

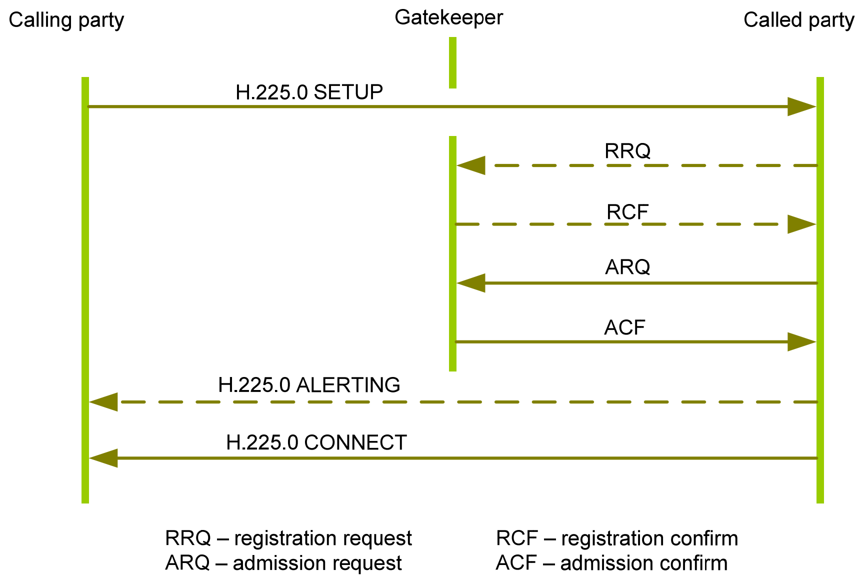


Figure 20 – Call setup using the direct signaling model

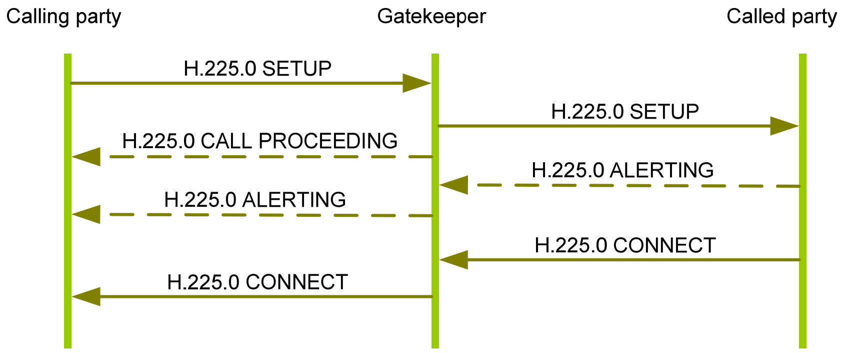


Figure 21 – Call setup using the gatekeeper-routed call signaling model

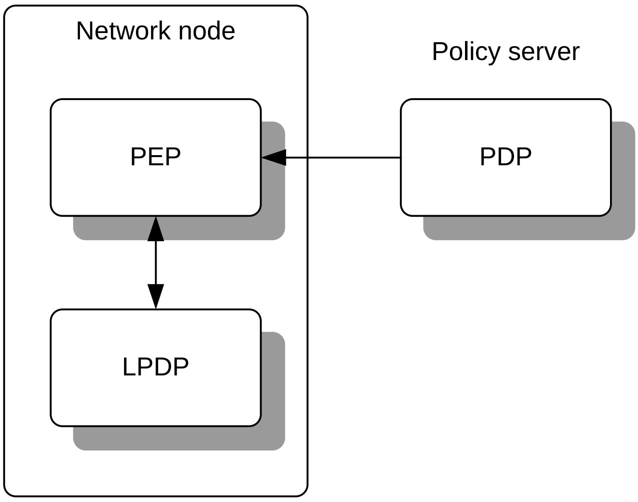
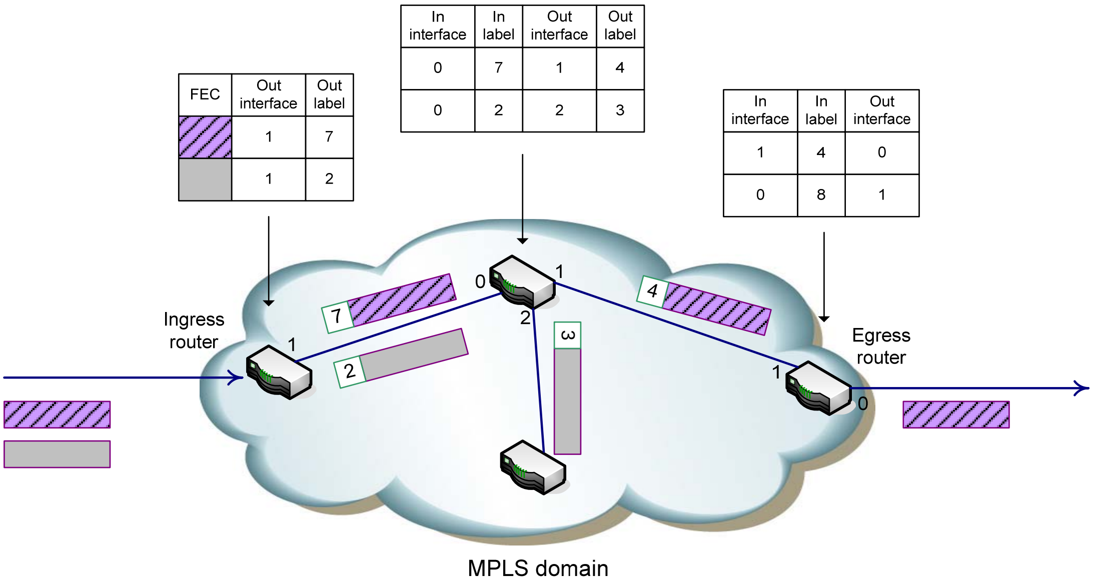


Figure 22 – COPS basic model



FEC	Out interface	Out label
▨	1	7
░	1	2

In interface	In label	Out interface	Out label
0	7	1	4
0	2	2	3

In interface	In label	Out interface
1	4	0
0	8	1

Figure 23 – MPLS basic operation

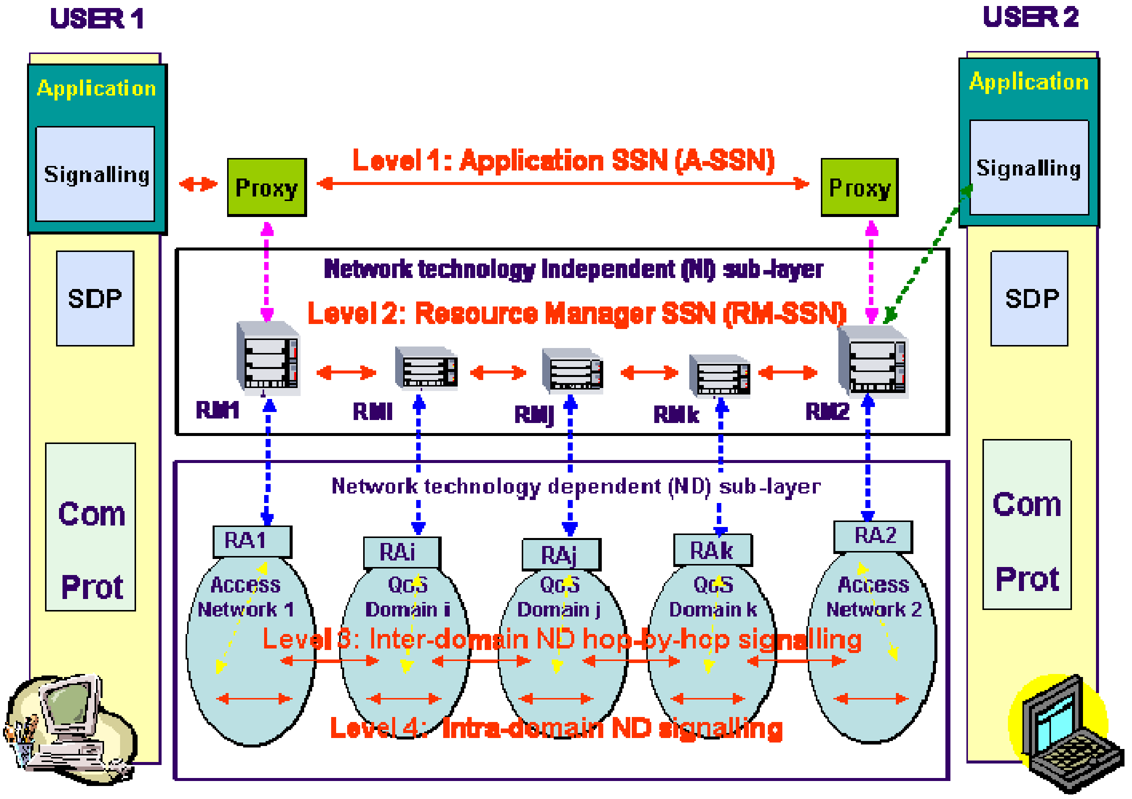

USER 1
USER 2
Application
Signalling
SDP
Com
Prot
Application
Signalling
SDP
Com
Prot
Proxy
Level 1: Application SSN (A-SSN)
Proxy
Network technology Independent (NI) sub-layer
Level 2: Resource Manager SSN (RM-SSN)
RM1
RMi
RMj
RMk
RM2
Network technology dependent (ND) sub-layer
RA1
RAi
RAj
RAk
RA2
Access
Network 1
QoS
Domain i
QoS
Domain j
QoS
Domain k
Access
Network 2
Level 3: Inter-domain ND hop-by-hop signalling
Level 4: Intra-domain ND signalling

Figure 24 – High-level view of the EuQoS signaling architecture

Table 1 – RSVP reservation styles

	Distinct Reservation Attribute	Shared Reservation Attribute
Explicit Sender Selection	Fixed-Filter (FF)	Shared-Explicit (SE)
Wild-card Sender Selection	---	Wild-card Filter (WF)

Table 2 – SIP Responses

Status code	Description
1xx	Informational (code range 100-199). Proceeding with the execution of the request
2xx	Success (code 200). The request was sucessfully parsed and executed by the called party
3xx	Redirection (code range 300-399). The call needs more processing by the calling party in order to be completed
4xx	Client request failure (code range 400-499). The request cannot be parsed by the server or cannot be serviced
5xx	Server failures (code range 500-599). The server cannot execute the request
6xx	Global failures (code range 600-699). The user request cannot be serviced by any server