ABSTRACT

KHAN, KHURRAM MATIN. COPS Usage for Managing Media Authorization (Under the direction of Dr Douglas Reeves)

The unreliable nature of packet networks leads to unpredictable service for the end user. It is hence desirable to negotiate end-to-end QoS before establishing multimedia calls. Different mechanisms for the reservation of resources have been defined to allow guaranteed service levels on packet networks, e.g. Resource Reservation Protocol (RSVP) and Differentiated Services (Diffserv). Authorization of network resource usage is, however, necessary before the resources can be reserved for a host and a multimedia session is established. To prevent fraud and facilitate a coherent view of the resource usage throughout the network, coordination between call signaling and resource reservation is necessary.

The media authorization framework being developed by IETF addresses these issues. When a host initiates a multimedia call, it is issued a media authorization token. This token is then transparently forwarded to the network edge router with the resource reservation request sent by the host. We have identified a few requirements that are not addressed by the current framework. For example, the network currently has no way of revoking an already issued authorization. Proper checks need to be in place to ensure that the token is not misused. Also, there is a need for closer cooperation between the network elements to allow a consistent view of the resources reserved by the user. This is of special significance in situations where different administrative domains involved have service agreements based on the actual resource usage (e.g. for billing purposes). A solution is proposed to overcome these problems and afford new possibilities in the framework, like sub delegation of resources. The objective of the proposed extensions is to provide greater flexibility in the framework and facilitate better coordination between the session

setup process and the reservation of resources. Common Open Policy Service (COPS) protocol is extended to exchange the additional messages and a new COPS client is defined.

COPS Usage for Managing Media Authorization

By

KHURRAM M KHAN

A thesis submitted to the Graduate Faculty of
North Carolina State University
In partial fulfillment of the
requirements for the Degree of
Master of Science

COMPUTER NETWORKING

Raleigh

August 2002

APPROVED BY:

Dr Douglas Reeves
Chair of Advisory Committee

Dr Peng Ning
Committee Member

Dr Mihail Devetsikiotis
Committee Member

BIOGRAPHY

Khurram Khan was born in Gujranwala, Pakistan in 1977. He received Bachelor of Engineering degree in Telecommunications from National University of Sciences and Technology, Pakistan in 1998. He has worked for over two years in Pakistan with various data networking companies. In the fall of 2000, he joined the Computer Networking Graduate program at North Carolina State University.

ACKNOWLEDGEMENTS

I would like to avail of this opportunity to express my heart-felt gratitude towards my advisor, Dr Douglas Reeves. His vision and outstanding knowledge in the field has been a continuous source of motivation for me. I have greatly benefited from his guidance and support. I would also like to extend my appreciation to the committee members, Dr Peng Ning and Dr Mihail Devetsikiotis.

I would like to thank my parents and family for their encouragement, support and prayers that I know I will always enjoy.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1 Introduction

With the increasing use of IP telephony in today's networks, new issues and concerns have risen for the users and the network operators. Both the users and the service providers alike have recognized the tremendous potential of IP telephony, but in order to match the existing services on the public switching telephone network and provide other value added services, the packet network infrastructure has to be robust and reliable. The unreliable delivery mechanism of the packet networks results in unpredictable service for the end user. It is hence desirable to negotiate end-to-end Quality of Service (QoS) before establishing multimedia calls. In this chapter, we look at some of the issues related to QoS networks and the motivation behind our research.

1.1 Motivation

While analysts foresee a continuing growth in the Voice over IP (VoIP) market, quality of service remains a major concern for the users. The International Telecommunications Union considers a one-way delay of 150 milliseconds as acceptable, whereas delays above 250 milliseconds are easily noticeable and often make the conversation difficult. Delays at different levels contribute to the end-to-end voice call delay. Switching delays in the network and the packet loss inherent in the packet networks makes it difficult to meet the tight delay budget in a voice call. Therefore QoS mechanisms need to be adopted in order to minimize latency for the voice packets and give a higher priority to the delay-sensitive applications. Providing Quality of Service for voice applications is a vital aspect of offering toll-quality voice in the IP networks.

Different mechanisms for the reservation of resources have been defined to allow guaranteed service levels on the packet networks, e.g. Resource Reservation Protocol (RSVP) [1] and Differentiated Services (Diffserv) [2]. Authorization of network resource usage is, however, necessary before the resources can be reserved for the host and a multimedia session is established. To prevent fraud and facilitate a coherent view of the resource usage throughout the network, coordination between call signaling and resource reservation is necessary.

The IETF working group, Resource Allocation Protocol (RAP) [3] has defined protocols and procedures to implement policies for the advanced network services such as QoS and traffic engineering. A simple protocol has been defined for supporting policy control over signaling protocols. Common Open Policy Service (COPS) [4] is the protocol that can be used to enforce and provision policies in the signaling networks. A general framework for policy based admission control has been defined and its application in RSVP and other networks is suggested. The working group has also defined a framework for setting up multimedia sessions with media authorization [5]. The framework suggests mechanisms for coordination between session setup and authorization of network resources, to prevent fraud and to ensure accurate billing.

The need for this "coordination" has been explained in the draft [5] as follows:

> "Mechanisms have been defined through which end hosts can use a session management protocol (e.g. SIP) to indicate that QoS requirements must be met in order to successfully set up a session. However, a separate protocol (e.g. RSVP) is used to request the resources required to meet the end-to-end QoS of the media stream. To prevent fraud and to ensure accurate billing, some linkage is required to verify that the resources being used to provide the requested QoS are in-line with the media streams requested (and authorized) for the session".

As we will discuss later, the focus of our research was to identify additional requirements in the suggested framework and propose necessary extensions.

1.2 Organization of Remaining Chapters

The arrangement of remaining chapters is as follows. In the following chapter we provide the reader with an overview of technologies related to our discussion. Chapter 3 explains the media authorization framework in more detail. Chapter 4 deals with some of the problems, requirements and assumptions about the existing media authorization framework. Chapter 5 will introduce the extensions we propose in order to overcome the discussed problems and afford new possibilities in the framework [5]. Chapter 6 will provide a more detailed discussion of the extensions we have proposed in the framework. New protocol objects and messages needed are explained. Finally chapter 7 will conclude with a summary of the problem at hand, the solution proposed and possible future work in this area.

Chapter 2 Technical Overview

Before going into the specific details of the media authorization framework [5], it would be helpful to introduce the various technologies and protocols relevant to our discussion. Although the media framework is independent of the underlying technologies, we will use specific protocols for the purpose of explanation. Session Initiation Protocol (SIP) [9] is used as an example of call signaling protocol. Resource reservation is accomplished in our examples by using the Resource Reservation Protocol (RSVP) [1] and the policy decisions are enforced in the network using the Common Open Policy Service (COPS) [4] protocol. Apart from discussing these protocols, we will also go over some telephony concepts in this chapter and introduce terminology used throughout the rest of the document.

2.1 Session Initiation Protocol (SIP)

The Session Initiation Protocol (SIP) is a simple text-based protocol used for initiating and managing multimedia sessions between users. It has been designed to make use of existing protocols wherever possible and hence simplify integration with other applications. SIP has been designed by The Internet Engineering Task Force (IETF) and is maintained by the SIP working group under Transport Area of IETF. The simple design and extensible nature of the protocol makes it suitable for most of the multimedia applications used over IP networks. Typically, SIP is used in conjunction with other protocols to build a complete multimedia architecture. Examples of such protocols are the Real-time Transport Protocol (RTP) [7] for transporting real-time data and providing QoS feedback and the Session Description Protocol (SDP) [8] for describing multimedia sessions.

SIP has been designed as a request response protocol. A user wishing to establish a session with another user will send a request stating this intention. This initial request message is the SIP INVITE method. The users in the network can be reached using special Uniform Resource Identifiers called SIP URIs. The format of a SIP URI is similar to an email address, typically containing a username and a hostname. A SIP proxy handles requests on behalf of a requestor and may be bypassed in situations where its services are not required. The figure below shows a typical example of a SIP voice call.

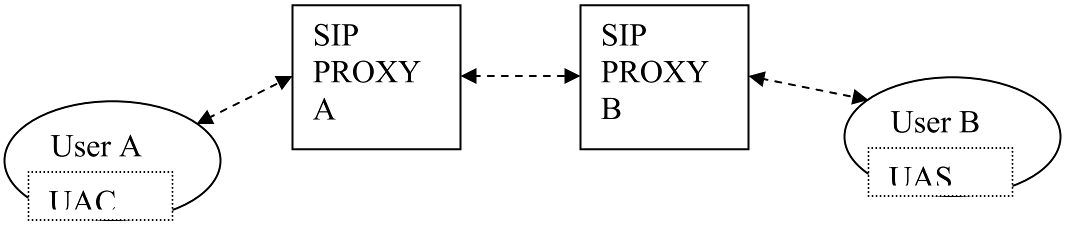

Figure 1: An example of entities involved in call establishment using SIP. UAC sends a SIP INVITE request to initiate the call

In this simple example user A uses his SIP phone to send a SIP INVITE request to user B. User A's SIP phone hence acts as a User Agent Client (UAC) and user B's SIP phone acts as a User Agent Server (UAS). This request also describes the session that user A wants to establish, using Session Description Protocol (SDP) [8] parameters. After SIP proxy A receives this message, it sends a 100 (Trying) response to user A, indicating that the proxy server is now working on its behalf. The INVITE request is then forwarded towards the destination until it reaches user B. The SIP application at user Bs side sends a 180 (Ringing) response to user A informing him that user B is now being alerted. If user B receives the call, a 200 (OK) response is sent to user A indicating that the call has been answered. This 200 response contains the SDP parameters describing the type of media session user B agrees to establish with User A. At this stage both the parties know the types of media sessions commonly supported and can establish a session with the characteristics acceptable to both users. After receiving the 200 (OK) response, user A's SIP phone sends an acknowledgement message back to user B confirming the establishment of the session. At this stage both users send media packets using the format they negotiated using the SDP parameters. The end-to-end media packets generally take a different path from the SIP signaling messages. After user B hangs up the phone, a BYE message is generated which is routed to user A's SIP phone. The BYE message indicates end of this session and is followed by a 200 (OK) response sent by user A to user B.

In more complex situations, SIP servers can forward INVITE requests to different locations if a user has moved or has requested to receive SIP calls at more than one location. SIP has been extended to provide other services like Instant Messaging or PSTN Intelligent Network (IN) services on IP networks. SIP also supports negotiation of Quality of Service (QoS) or security parameters between the hosts involved in a multimedia session. The fulfillment of these requirements can be made a pre-requisite for call establishment so that proper resources or security can be ensured before the involved parties are alerted.

A concept of pre-conditions [9] has been introduced in SIP for situations where the involved parties only want to establish sessions provided certain conditions have been met. After the initial INVITE request, the UAS does not send a 180 (Ringing) response but sends a 183 (Session in progress) response instead. The phone at the Callee side does not ring at this stage. The initial INVITE request contains the conditions that need to be satisfied before a session can be established. The 183 response sent by the UAS also contains the SDP description of the media session that the callee wants to establish and the required conditions. The UAC then responds with a PRACK (Progress Acknowledgement) response. The three way handshake is completed by a 200 (OK) message sent by the UAS. The end hosts at this stage can try to satisfy the negotiated set of conditions, for example, reservation of resources. After the end hosts complete the required conditions, a SIP UPDATE request is sent by the caller, which is acknowledged by a 200 (OK) response. Both ends indicate the status of required conditions in these messages and proceed with the session establishment process, if required conditions are satisfied. If status of any of the required conditions changes during the session, either party can generate a SIP UPDATE [10] request and terminate the session if involved parties are unable to meet the required conditions.

2.2 Resource Reservation Protocol (RSVP)

Resource Reservation Protocol (RSVP) [1] provides mechanisms for applications to reserve network resources like bandwidth or processor usage. An RSVP aware application on an end host transmits a resource reservation request through the network with each router, on the path towards the destination, committing to reserve the requested resources for this end host. In case any of the intermediate routers does not commit to this reservation, the end host cannot be guaranteed the desired quality of service. The routers, after committing to the reservation, maintain a soft state for the reservations made for each application. This soft state needs to be refreshed regularly by exchanging messages between the routers. The routers can use these refresh messages to monitor the actual resources being used by the application or provide alternate routes between the end hosts. Quality of service is implemented by using packet classifiers, that determine the QoS class and packet schedulers that ensure promised QoS on the outgoing interfaces.

RSVP reserves resources in one direction and hence to provide a guaranteed service in both directions, separate requests need to be generated. Also, since the resources are reserved in one direction, RSVP treats a sender as logically distinct from a receiver, although the same application might act as a sender and a receiver at the same time. The sender initiates the process by sending an RSVP PATH message towards the receiver. The receiver then requests a reservation by sending an RSVP RESV message towards the sender.

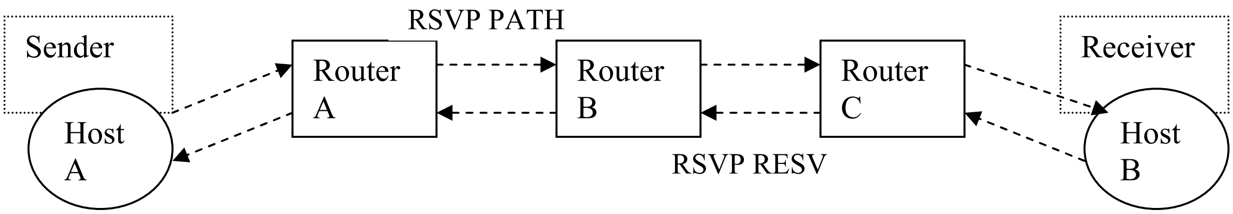

Figure 2: Sender initiates the reservation process by sending an RSVP PATH message towards the receiver. Receiver reserves the resources by sending an RSVP RESV message.

In the figure above, host A originates an RSVP PATH message towards host B. This message is sent to the first router in the network, router A. The RSVP PATH message records the next hop information so that the RSVP RESV messages can take the exact reverse path through the network. The PATH message also contains the estimated size of flow in the form of TSpec so that the receiver can choose an appropriate size for reservation request. Router A checks its local decision modules to see if enough resources are available to fulfill the requested reservation. Also a policy control request is made locally to verify that host A has administrative permission to reserve these resources. The RSVP PATH message is then forwarded to router B and subsequently all intermediate routers towards host B. All intermediate routers similarly make the admission control and policy control decisions locally. Upon reception of the PATH message, host B generates an RSVP RESV message to request reservation from the network. The RSVP RESV message specifies the requested service in the form of RSpec and the size of the expected data flow (TSpec). The RESV message also specifies which packets can use the reservation by providing a FilterSpec. Each router receiving the RESV message makes local admission control and policy control decisions. In case either of the tests fails, an error message is returned to host B informing it of the problem. If the reservation is successful in all routers, a RESVCONF message is sent to host B, indicating that the requested reservation was most likely successful.

An interesting attribute of RSVP is the ability to carry user-provided policy control information. This policy control information is opaque to RSVP and serves as an input to the policy control decisions on intermediate routers. The policy information extracted from the RSVP messages and provided to the policy control module is referred to as "policy data", which RSVP carries in POLICY_DATA objects. The policy data may contain authentication information about the user or user classes.

2.3 The Common Open Policy Service (COPS) protocol

The Common Open Policy Service (COPS) [4] protocol is a simple query and response protocol that can be used to implement policy control in signaling protocols like RSVP. As discussed in the previous section, RSVP requires policy control decisions to allow or reject reservation requests received from the applications. COPS defines mechanisms to exchange policy information between a policy server and its client, for example an RSVP router. The policy server is responsible for making the policy control decisions and is called a Policy Decision Point (PDP). The client requesting decisions from the PDP is called a Policy Enforcement Point (PEP).

An optional Local Policy Decision Point (LPDP) can make local decisions in situations where the remote PDP is not available for the decisions. COPS is a stateful protocol in that the requests and the corresponding decisions are remembered by the PDP until they are explicitly deleted by the PEP. Also, a PEP maintains state information for the decisions issued by the PDP and will continue to use the last issued decisions if unable to receive new decisions. The following figure illustrates a basic model for the various components involved in the COPS framework.

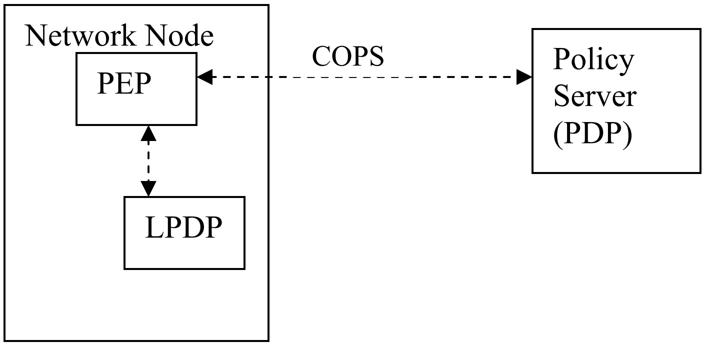

Figure 3: COPS Model: The PEP requests decisions from the PDP. Optional LPDP is used in cases where PDP is not available.

A PEP is responsible for initiating and maintaining a TCP connection with the PDP. The PEP uses this connection to send the requests to the PDP and receive the decisions. There are two models defined for the enforcement of the policy decisions, outsourcing and configuration. In the outsourcing model, the decisions are pulled from the PDP. For example, when a Policy Enforcement Point (PEP) receives an RSVP message that requires a policy decision, the relevant RSVP objects from the message are put into a COPS Request message (REQ) and sent to the PDP. In response to this request the PDP makes a decision and returns a COPS Decision message (DEC) to the PEP. In the outsourcing model, there is a direct relationship between the requests sent by the PEP and the decisions issued by the PDP. A PDP can, however, send unsolicited messages to modify the previously approved request states. After the PEP has completed the decision given by the PDP, it sends a COPS Report (RPT) message to PDP with the new status. In the configuration model, the policy decisions are pushed by the PDP to its clients. A PDP can send provisioning information to the PEP responding to external events, PEP events or a combination of these two. Whereas outsourcing model provides instantaneous policy decisions, the configuration model is more flexible in its application.

Since COPS uses self-identifying objects, it is easily extensible and can support diverse client types. Different clients might define different client specific data and might require different kinds of policy decisions. To differentiate between different clients, a client type field is included in each COPS message.

2.4 Internet Telephony Basics

Traditional telephony involves users connecting through the Public Switched Telephone Network (PSTN) using dedicated switched-circuit channels. Internet telephony, however, makes use of the IP infrastructure to transmit voice across the network. Computers can establish and release voice calls with other computers using telephony protocols. Internet Telephony is growing because of the lower cost and efficient use of the available bandwidth with this technology. Unlike PSTN, the voice data on the IP networks can be compressed and hence the same amount of bandwidth can be utilized for a much greater number of calls. Also, companies save costs by maintaining only one network with the ability to send voice and data on it.

Since traditional telephony networks have already been installed at a large scale, there is a need for communication between PSTN and the Packet Networks. Internet Telephony then involves three kinds of calls, computer to computer, computer to PSTN and PSTN to computer. The link between the circuit-switched network and the Internet is provided by gateway devices. “Gateways” provide translation between the Internet protocols and the PSTN infrastructure. Certain gateways also provide translation between different types of media, for example translating Real-time Transport Protocol (RTP) packets from the Internet to Time Division Multiplexed (TDM) packets on the circuit switched network. Gateways also provide translation between IP addresses and telephone numbers, so that a user on PSTN can reach any Internet user by knowing his phone number and vice versa.

Telephony requires signaling in order to setup, manage and release voice calls. This function is performed in traditional networks using the Common Channel Signaling System #7, commonly known as SS7. On the Internet, protocols like SIP and H.323 provide the architecture for multimedia signaling. Using these protocols, telephony services like toll free calling, roaming services and Local Number Portability (LNP) are also supported on the packet networks. Both PSTN and the Internet Telephony networks of today provide out of bound signaling for voice calls. Since the signaling network is different from the circuit-switched network, signaling and voice data generally take different paths through the network. The network carrying voice traffic is generally called the *bearer domain* and the network carrying signaling traffic is called *signaling domain*.

For widespread deployment of Internet Telephony to become a reality, providing QoS is also important. Unlike circuit-switched networks, the voice calls on the Internet do not have a dedicated channel between the two parties. To provide guaranteed level of service to the user, resources must be reserved on the network using protocols like Resource Reservation Protocol (RSVP). For users to migrate to Internet Telephony infrastructure, it not only has to be robust and reliable but should also provide comparable or better value added services as compared to PSTN.

The discussion in this chapter was meant to serve as background information for the rest of the document. The terminology introduced in this chapter will frequently be used in the chapters that follow. In the next chapter, we will provide an overview of the media authorization framework. We will see how the framework provides coordination between the call signaling and resource reservation.

Chapter 3 The Media Authorization Framework

As mentioned earlier, the draft “Framework for session set-up with media authorization” [5] defines a framework to allow coordination between session setup and media authorization. During the call establishment process, user is issued an authorization “token” by a policy server (or a session management server). This token is then transparently forwarded to the network edge router with the resource reservation request (e.g. RSVP PATH message) sent by the host. The policy enforcing elements in the network (e.g. COPS PEP) can check this token and verify that the requested resources are in line with the media authorization for that user. The format of this authorization token has been defined in the draft “Session Authorization for RSVP” [11]. In this chapter we take a detailed look at the media authorization framework, the network elements involved and the messages exchanged between them.

3.1 Network Diagram

As an example application of this framework, we will consider a network where call signaling is handled by SIP and the resources are reserved using Resource Reservation Protocol (RSVP). The call signaling messages traverse the Service Control Domain (SCD) and media packets are sent over the Resource Control Domain (RCD). The network model would be as shown in figure 4. We will refer to this network model in our further discussion on the media authorization framework throughout this document.

To understand the sequence of messages that are exchanged, take the case of a SIP User Agent Client (UAC) establishing a call with a SIP User Agent Server (UAS). The response to the initial INVITE request from the UAC will be a SIP 183 message sent by the UAS. The Originating SIP proxy (OP) on receiving this Session Progress message can request a policy decision from it’s associated Policy Decision Point (SCD PDP). If the user is authorized to make the request, the PDP will issue a media authorization token, which is encoded in the SIP 183 message and sent to the UAC. Before sending any media, the end host will send an RSVP PATH message to reserve the required resources in the network. This PATH message will also carry the media authorization token previously issued to the user. On reception of this PATH message, the Edge Router (ER) will send a COPS request to it's associated PDP (RCD PDP). The COPS Request (REQ) message also carries the media authorization token as policy data. The PDP can now verify the authorization for the user and reply with a COPS decision (DEC) message. Similarly, all the routers on the path towards the RSVP receiver will request a COPS decision from their associated PDPs.

A detailed explanation of the messages involved in this process will be given shortly.

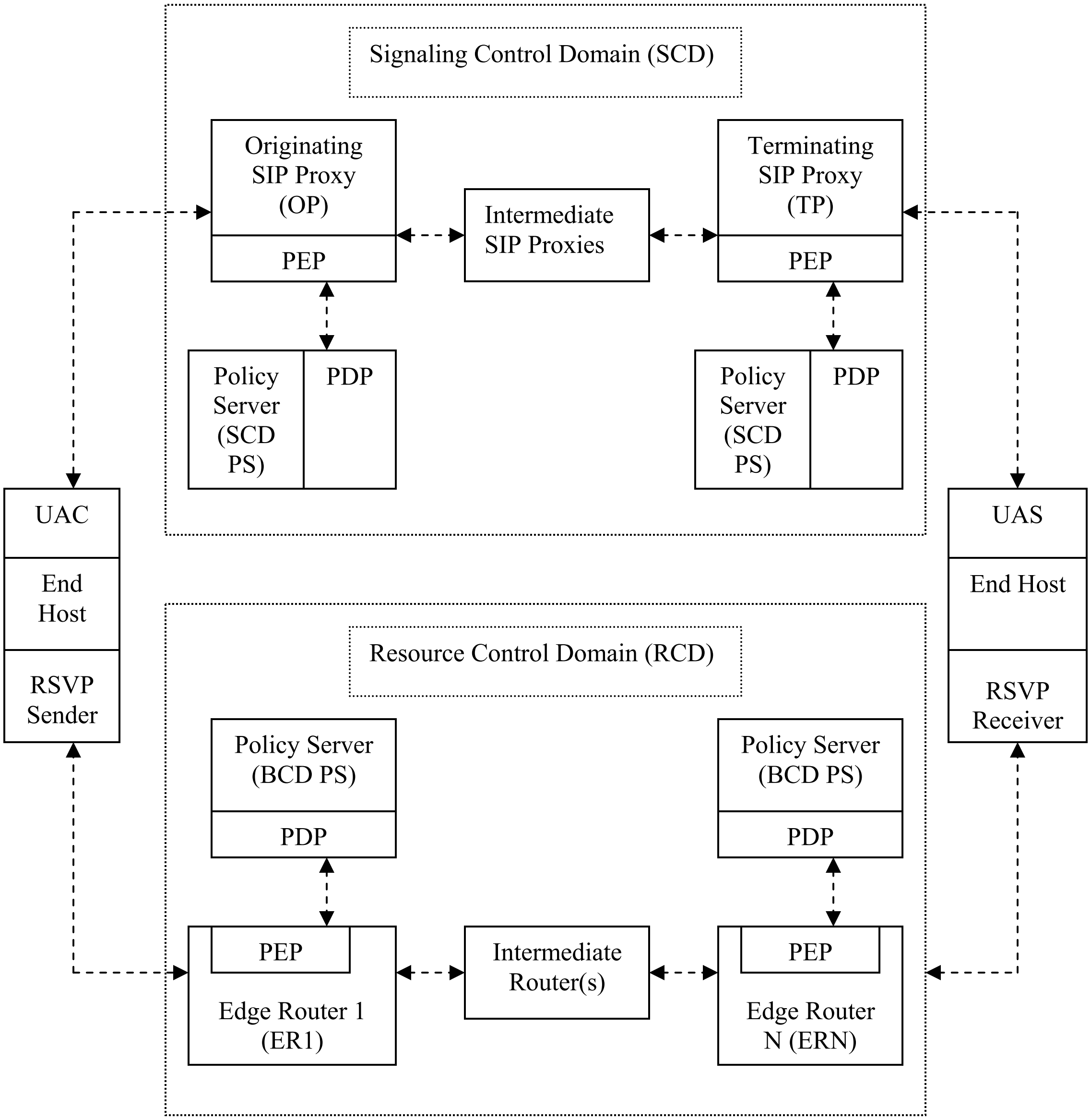

Figure 4: Network Diagram: An example of the Media Authorization Framework network model presented in "Framework for session set-up with media authorization" [5]. SIP is used here for call signaling and RSVP for resource reservation.

3.2 Network Models

The network diagram presented in the previous section is one example of how the network elements in the framework interact with each other. Depending on the network topology and the trust relationships that exist between the network elements, the framework can be modeled differently. For example, if the network topology is simple enough, one Policy Server (PS) can handle the functions of both SCD PDP and RCD PDP. In this case, the policy decisions for the Service Control Domain and the Resource Control Domain will be handled by this Policy Server. Whether the policy decisions in the two domains are made by the same policy server or not, both the Edge Router and the SIP Proxy need to have a trust relationship with the associated PS. A trust relationship between two entities implies that both are aware of each other's identity and are configured with security keys, if needed, to establish secure connections between them. Such a relationship can be "pre-established" in which case the entities know of each other's identity before setup of a session. If, however, a pre-established relationship between the entities does not exist, it might be possible to establish this relationship dynamically. In order to establish a trust relationship dynamically, the entities need to be in the same domain or within domains that have a business relationship and agreements on resource usage and information sharing. Functions like reporting the start and stop of service and checking for agreements between the two domains is out of the scope of our discussion. A discussion of such functions is provided in [15].

Three models have been defined in the media authorization framework depending on the network topology and the trust relationships between the network elements like the Edge Router, the Policy Server(s) and the SIP Proxy Server.

- The Coupled Model

- The Associated Model
- The Non-Associated Model

In the following sections we explain the assumptions and requirements in each of these models. We will take a more detailed look at the non-associated model because it satisfies the assumptions of our research.

3.2.1 The Coupled Model

In the Coupled Model, a tight relationship between network elements is assumed. This model is suitable for situations where the network is not a complex one. The number of multimedia sessions established is small enough to allow a single Policy Server (PS) to make all policy decisions. Hence, in this model a common Policy Server makes policy decisions for both SCD and RCD. The identity of network elements involved is known a priori. For example, the End Host might be aware of the SIP Proxy that will handle the call signaling. Also, there is a pre-established relationship between the SIP Proxy and the PS and between the Edge Router and the PS. Both SIP Proxy and the Edge router know the identity of the Policy Server and have been configured with the security keys if needed. The following figure shows the Coupled Model.

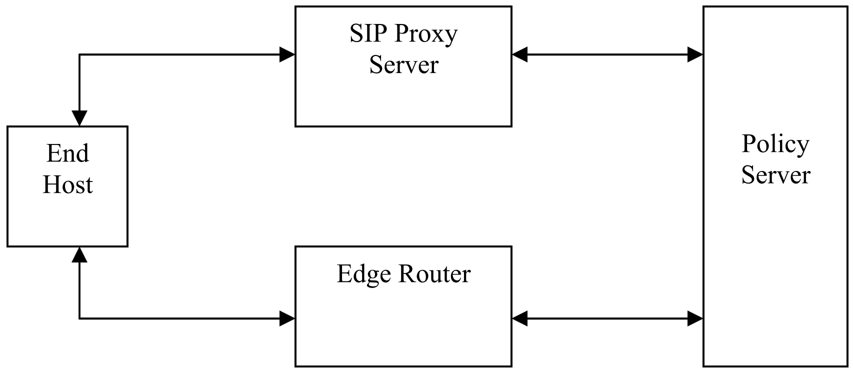

Figure 5: The Coupled Model: A single Policy Server issues decisions for both RCD and SCD.

The media authorization token issued by the policy server in this model contains a pointer to the session state on the PS. After receiving the authorization token in RSVP PATH message, a policy control request from the edge router verifies that this resource reservation request is in line with the authorized media for a particular host.

3.2.2 The Associated Model

This model applies to networks that are too complex for the coupled model. It is assumed that the network is too complex for each network element to be aware of other elements. In such a network, a single SIP Proxy cannot handle all the session setup requests. Similarly the host might send the resource reservation request to different edge routers, for example in the case of a mobile user changing his location continuously. Furthermore, there are more than one Policy Servers involved in the policy control decisions. The network elements still have pre-defined relationships which makes communication between them possible. For example, the SIP Proxy and the PS should have a pre-defined relationship between them and Edge Router and the PS should have a relationship between them.

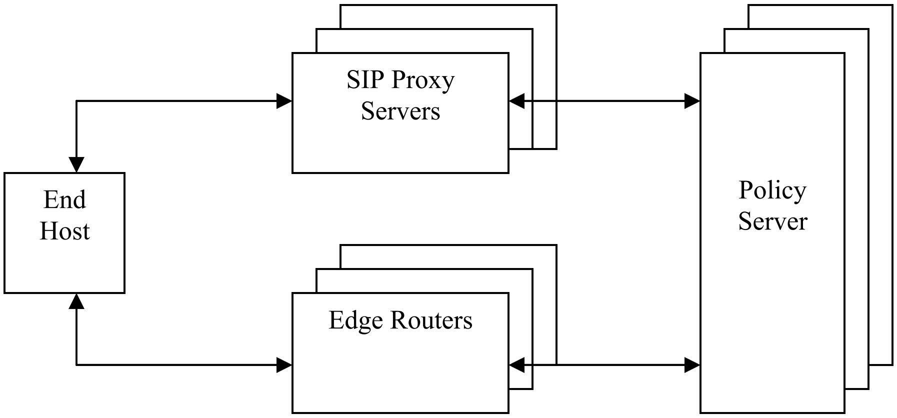

Figure 6: The Associated Model: Multiple instances of network elements are present in the topology.

The Policy Server in the Associated model also includes its identity in the authorization token. During the resource reservation process, edge router extracts the identity of the Policy Server that authorized the media and contacts it for a policy control decision about the reservation.

3.2.3 The Non-Associated Model

In this model, the Resource Control Domain (RCD) and the Service Control Domain (SCD) use different policy servers for policy control decisions. Both Policy Servers make independent decisions for their domains. The network elements do not have knowledge of the network topology and there are no pre-established trust relationships between the network elements in RCD and the SCD. There is, however a pre-defined relationship between the SIP Proxy and the SCD PS and the Edge Router and the RCD PS. The network model for the non-associated model is as follows:

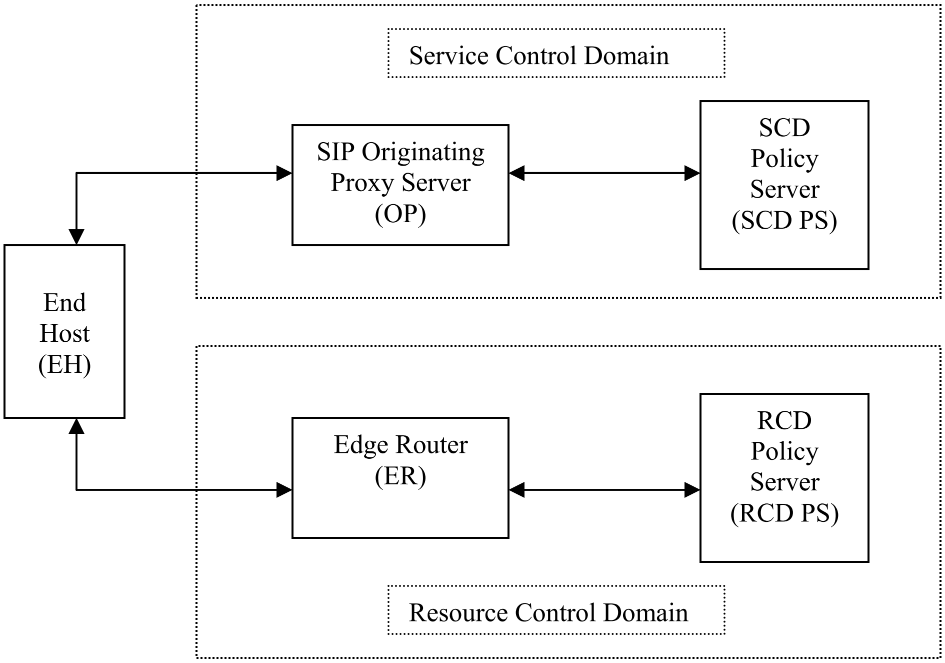
Service Control Domain
SIP Originating Proxy Server (OP)
SCD Policy Server (SCD PS)
End Host (EH)
Edge Router (ER)
RCD Policy Server (RCD PS)
Resource Control Domain

Figure 7: The Non-Associated Model: The separate Policy Servers in RCD and SCD make independent decisions for their domains.

The non-associated model further necessitates some information in the authorization token. The identity of the caller and the called host has to be added in the token. The characteristics of the authorized resources and the identity of the authorizing entity also need to be included in the token. Furthermore, authentication data is added with the token to protect it from tampering. Format of this authorization token is provided in Appendix A of this document.

3.3 Message Flow

The sequence of messages exchanged during session setup with media authorization is as follows:

1. When a user goes off-hook and dials a telephone number, the UAC collects the dialed digits and sends the initial INVITE request to the originating SIP proxy.
2. The originating SIP proxy (OP) authenticates the user/UAC and forwards the INVITE message to the next SIP proxy towards the UAS.
3. Assuming the call is not forwarded, the terminating end-point sends a 183 response to the initial INVITE via the OP. Included in this response is an indication of the negotiated bandwidth requirement for the connection (in the form of SDP parameters).
4. When the OP receives the 183, it has sufficient information regarding the end-points, bandwidth and characteristics of the media exchange. It initiates a Policy-Setup message to associated SCD PS.
5. The SCD PS generates an authorization token that is returned to the OP. The contents of this token have been defined in [11] and provided in Appendix A of this document.
6. The OP includes the authorization token in the P-Media-Authorization header extension of the 183 message. Upon receipt of the 183 message, the UAC stores the media authorization token from the P-Media-Authorization header.

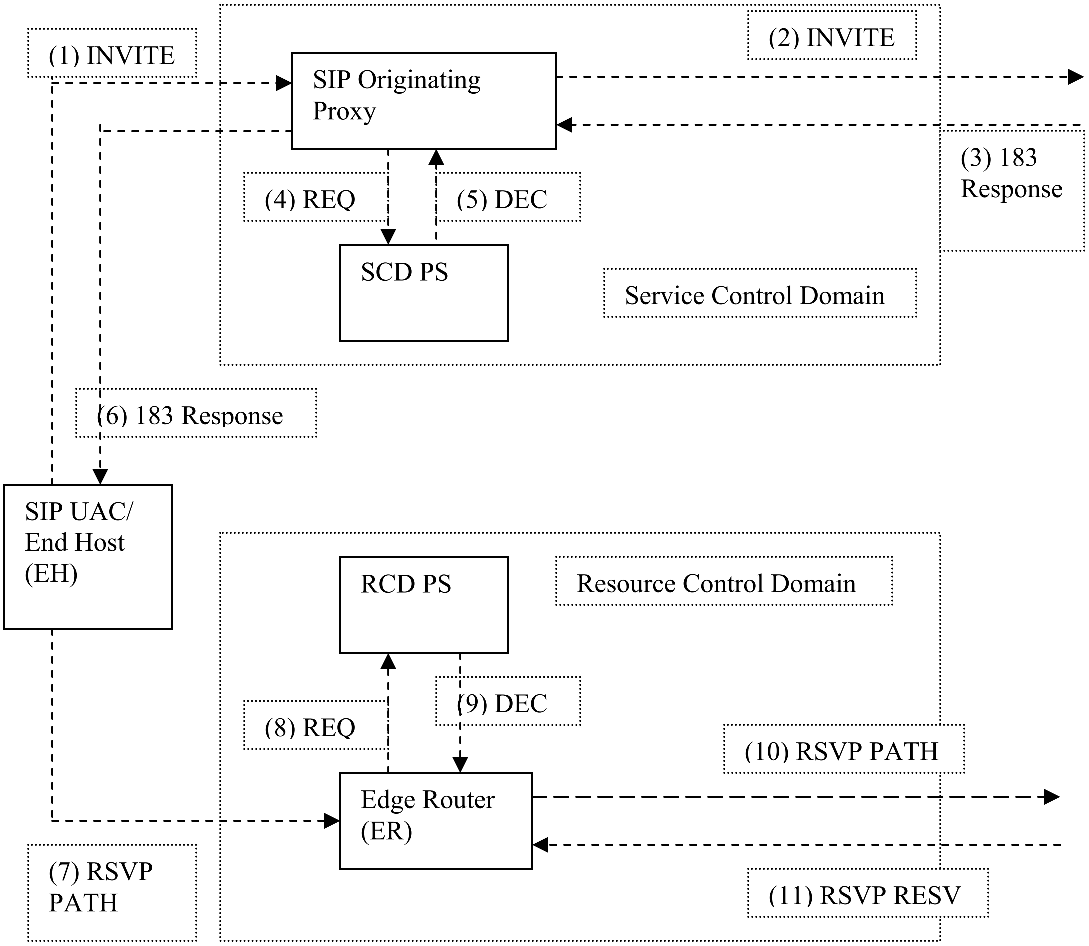

Figure 8: Message Flow: The sequence of messages from Media Authorization Framework [5]. Initial phase of call establishment using SIP and resource reservation with RSVP.

7. Before sending any media, the End Host (EH) requests QoS by sending an RSVP-PATH message, which includes the previously stored P-Media-Authorization-Token as a Policy-Element.
8. The edge router at the originating side, upon reception of the RSVP-PATH message checks the authorization through its associated policy server using COPS message exchange
9. If the authorization is successful, the RCD PS returns an "install" Decision.

10. Edge Router at the originating end checks the admissibility for the request and if admission succeeds, it forwards the RSVP-PATH message. Similarly other routers on the path check for admissibility and proper authorization.
11. Upon successful reservation in all the routers upstream, the edge router on the originating side will receive an RSVP RESV message indicating that resources have been reserved upstream.

In this chapter, we looked at the Media Authorization Framework and the various Network Models presented in it. Application of a specific model depends on the network topology and the relationships existing between the network elements like the End Host, SIP Proxy Server and the Edge Router. The Non-Associated model is most relevant to our research, since it satisfies our assumptions about the relationships existing between the Service Control Domain and the Resource Control Domain. In the following chapter we will identify some additional requirements in this framework and the motivation behind extending the existing framework.

Chapter 4 Problems, Requirements and Assumptions

We now provide a more detailed discussion of the non-associated model and the assumptions and requirements that motivated our research. The observations about the media authorization framework [5] provided in this chapter establish the basis for our proposed extensions to the framework.

4.1 The Non-Associated Model revisited

Since our solution is concerned with the non-associated model, it would be useful to mention a few salient features of this model.

- A common feature of all the network models discussed above is that the user cannot be trusted. The contents of the authorization token are digitally signed to protect it from tampering. Any change in the token will be easily recognized when the message signature is verified.
- Once an authorization token is issued to the user and the resources are reserved, the user cannot be expected to revoke this reservation voluntarily. Although in most cases the user will relinquish the reservation after the completion of the call, a malicious user can block resources even after the session has ended. In the event that the reservation has to be torn down, the network has to take some action to free the resources.
- An important point in the non-associated model is that the SCD PS does not know the identity of the RCD PS a priori. Similarly the RCD PS is unaware of the identity of the SCD PS before the session establishment. Hence, no pre-established relationship exists between the two Policy Servers.

- The RCD PS only knows the identity of the SCD PS *after* receiving the policy element encoded in the resource reservation request.
- There exists a business relationship between the Resource Control Domain and the Service Control Domain. It is assumed that a trust relationship can be dynamically established between the network elements in the two domains using mechanisms like Public Key Infrastructure (PKI) [13]. For example, if the SCD PS includes it's digital certificate in the authorization token, the RCD PS can establish the identity of the SCD PS by contacting a Certificate Authority (CA) and validating the certificate provided in the token.

Other relevant observations about the non-associated model are as follows:

4.2 Revoke Authorization

In this framework, network has no way of revoking an already issued authorization token. Once the media authorization token is issued, the Service Control Domain loses control over the media authorization given to the user. If an error occurs in the Resource Control Domain, PDPs can send an unsolicited decision to routers and tear down the reservation. However, if some anomaly is discovered in the Service Control Domain, there is no way of communicating this to the Resource Control Domain. Hence, the mechanism explained above needs to be extended in order to account for such cases. For example, when operator in the SCD needs to force termination of the resource reservation and have the user renew authorization.

4.3 Prevent misuse of token

As mentioned above, the user cannot be trusted completely with how the token will be used at the host. The authorization token can have a timestamp in order to prevent usage after certain duration. A malicious user can still use the authorization token for more than one reservations as long as it is used before the time of expiration. The authorization lifetime would minimize the chances of using one authorization for multiple reservations in the same resource domain. However, a single authorization token can possibly be used with two or more different networks and reserve resources at the same time.

4.4 Sub-delegate resources

In some environments, the network operator might benefit from the ability to allow an end host or a router to sub-delegate resources. One possible extension to the framework could be to allow an end host or a router to sub delegate the authorized network resources. In the existing framework, the Service Control Domain is not aware of the actual resources reserved by the host. Within this framework, sub-delegation of the resources might not be feasible. This concept could be useful, for example, in cases where service providers issue a chunk of bandwidth to another entity and allow it to manage this amount of bandwidth independently.

4.5 Synchronize resource usage information

In the absence of communication between the SCD PS and the RCD PS, the two domains might have an inconsistent view of the resources reserved by the host. One possible scenario in this framework can be when the user was only able to reserve fewer resources than it was allowed. Since the Service Control Domain does not get a feedback of the actual resources reserved in the Resource Control Domain, the two might have different view of the resource usage. This could be a significant problem in situations where the two domains have service agreements that depend on actual resource usage (e.g. for billing purposes).

We have discussed, in this chapter, certain requirements that currently can not be satisfied by the media authorization framework [5]. The observations about the framework and additional requirements mandate extensions to the media authorization framework. In the following chapter, we will show that our proposed extension to the framework does not increase the call setup delay. The implementation only affects the Common Open Policy Service (COPS) [4] protocol, which is an easily extensible protocol with self-identifying objects. The solution therefore adds minimally to the complexity of the framework. We will also discuss how extending the framework helps in achieving our requirements.

Chapter 5 Proposed Extension

The focus of our research was to introduce, in the framework, the ability to revoke an already issued authorization. In considering a solution for this problem, we came across other requirements that were mentioned in the previous chapter. In this chapter we present the proposed extension to the framework. Introducing the extension to the framework not only allows the SCD PS to revoke an already issued authorization but also affords new possibilities for the network service operator managing the RCD.

5.1 Considerations

In considering a solution for the above-mentioned problems, we realize that any further complexity in the framework should be avoided. Any changes should preferably not add any additional delay in the call establishment process. We made an effort to minimize the overhead incurred due to the suggested changes in the framework. Also, the effect on other protocols should be minimal. As we shall see soon our proposed extension to the framework only affects the COPS [4] protocol.

To satisfy the requirements discussed earlier and achieve more functionality, the following addition to the framework is proposed.

5.2 Additional message exchange

Our proposed extension to the framework involves creating another connection between the SCD PS and the RCD PS. After the resource reservation process is complete, the RCD PS should send a message to the SCD PS reporting successful reservation of the authorized resources. This message should contain information about the session for which confirmation is being sent and information about the source and destination entities. The message also contains the ID of the sending entity, the RCD PS in this case. Similarly, information about the actual resources reserved can be sent to the SCD PS. Because of this message the SCD PS is now aware of the identity of the concerned RCD PS and the actual resources reserved. The SCD PS can now inform the RCD PS of any abnormal conditions and request termination of the reservation that it previously authorized. We will look at some of the advantages we gain by adding these steps to the process and then explain the additional message exchange in detail.

5.3 Advantages

The following advantages of the proposed solution are worth mentioning

a) Note that the call establishment process does not suffer any delays since the additional message is sent after the reservation process is complete. The SCD PS checks the session state to see if the resources reserved are in line with the authorized media. In most cases the SCD PS will not need to do any further processing. If however a discrepancy is found, it can send a message back to the RCD PS instructing it to terminate the reservation.

b) This method is scalable since it only involves communication between the policy servers. The overall performance is not affected by increasing the number of policy servers in the network, since only one policy server (i.e. RCD PS associated with the edge router) needs to communicate with the SCD PS.

c) In this mechanism the SCD PS is aware of the resources that the user has actually reserved. Hence sub-delegation is possible. (The exact mechanism to achieve this has to be defined).

d) The Service Control Domain and the Resource Control Domain will have a consistent view of the actual resource usage. The reports received from the RCD can be used for accounting purposes.

e) The user cannot use the same authorization token for more than one reservation. In case of such attempts, the SCD PS will get feedback from all the Resource Control Domains involved. The communication from RCD includes the Session ID for a reservation. This Session ID was issued by the SCD PS and is included in the Authorization token communicated to the RCD PS. Hence, the SCD PS can take action to terminate all the reservations except for the valid one.

f) After the RCD PS establishes a TCP connection with the SCD PS for one policy request, it only needs to send one message for additional policy requests. Hence the overhead is minimal.

5.4 Types of messages exchanged

Hence, to provide better coordination between call signaling and resource reservation, we have identified the need to add three types of messages in the framework.

1. Message confirming the reservation of resources in the Resource Control Domain (RCD). This message is sent from the RCD PS to the SCD PS. The message should include the identity of the sending entity, the Session ID for which the message is being sent and information about the source and the destination hosts establishing this session.
2. Reports of actual resource usage sent by the RCD PS to the SCD PS.
3. Message to revoke a previously authorized reservation. The SCD PS sends this message to the RCD PS. This message should also have the identity of the sending entity, Session ID and Source and destination of the session for which the message is being sent.

5.5 Protocol selection for message exchange

The above-mentioned messages can be exchanged by using the Common Open Policy Service (COPS) protocol [4]. Some of the advantages that COPS provides are as follows:

- Both the RCD PS and the SCD PS can maintain state information for the established sessions. We have to define a new COPS client to exchange these messages. For our purpose the RCD PS will act as a PEP and the SCD PS will act as a PDP.
- The information about the session can be synchronized at both ends. COPS provides mechanisms to synchronize this information between the PEP and the PDP.
- Since COPS uses TCP as transport protocol, a reliable connection can be established between the two domains. For transport level security, “COPS over TLS” [14] can be used.
- The Service Control Domain (SCD) policy server can ensure the tear down of reservation, i.e. a feedback from the RCD PS is also possible after teardown. This mechanism is already defined in COPS since the PEP can send a COPS Report (RPT) message to the PDP after the decision command has been completed.

Hence we define a new COPS client to support the messages discussed in the previous sections. The RCD PS will act as a PEP in this scenario and SCD PS, as PDP.

The additional messages for reservation would then be as follows:

12. After the reservation of resources, the PEP in the edge router will send a COPS “Report State” (RPT) message to the associated policy server confirming the reservation. This Report message is already a part of the COPS framework and indicates successful

completion of the reservation. The Edge Router will generate this Report Message for the RCD PS after receiving and processing the RSVP RESV message.

13. After receiving this RPT message, the RCD PS will send a COPS REQ message to the SCD PS indicating successful reservation of the resources in send direction. This Report (RPT) message will contain the identifier for our new COPS client and hence the SCD PS will record information contained in this message. The contents of this message have been explained in the next chapter.

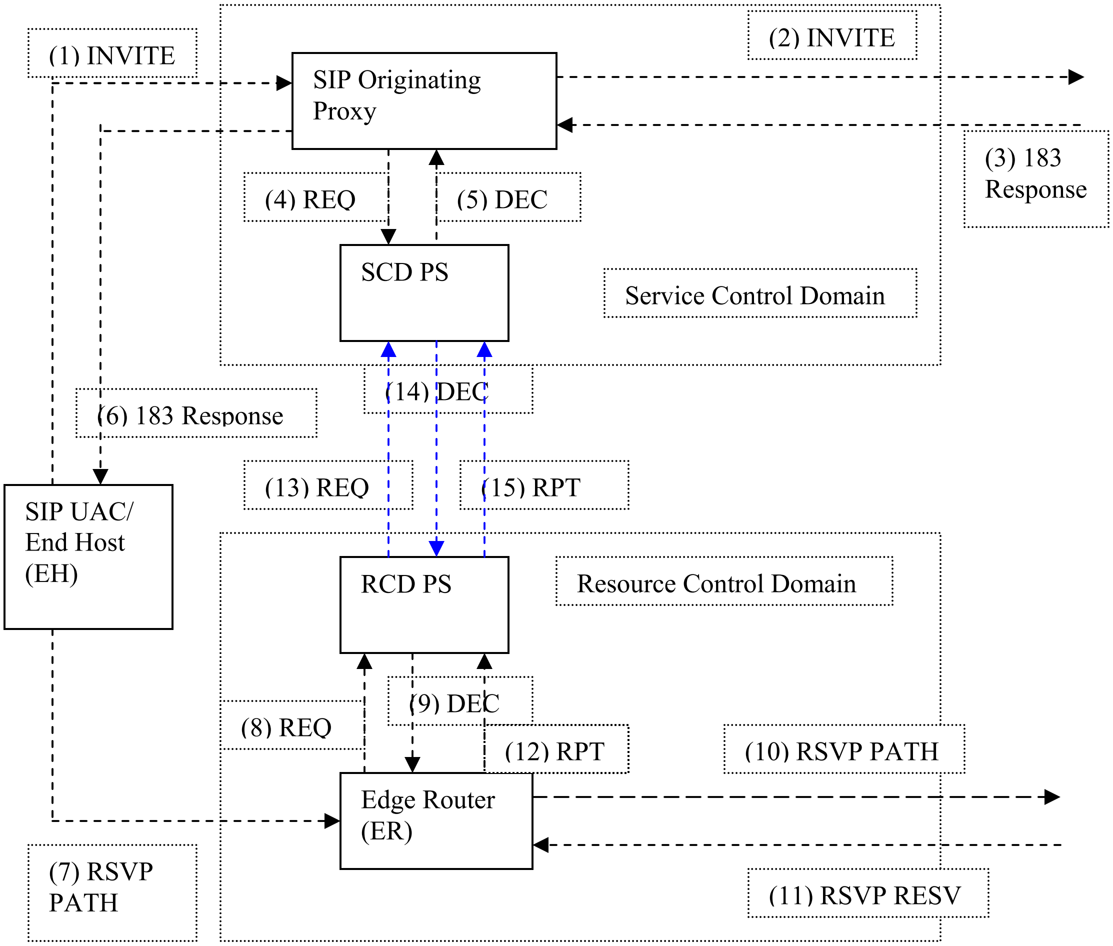

Figure 9: Additional messages exchanged between the RCD and the SCD to increase flexibility in the framework.

14. The SCD PS can send a DEC message back to the RCD PS to confirm the acceptance of this session establishment. If no discrepancies are found, an install decision is sent which does not require any further processing by the RCD PS. The format of this Decision message is also specific to our COPS client and is explained in the next chapter.
15. A RPT message sent from the RCD PS to the SCD PS confirms the reservation of resources.

If at a later stage the SCD PS needs to issue a reservation tear down message it can send an unsolicited COPS Decision (DEC) message to the RCD PS as defined in the COPS specification. The sequence of messages exchanged will be as shown in figure 10.

1. The SCD PS sends an unsolicited COPS Decision (DEC) message to the RCD PS. The command sent in this message will be “remove”. This signals to the PEP in the RCD PS to terminate the connection.
2. Upon receiving this message, the RCD PS should send a COPS DEC message to the edge router, with the “remove” command. The format of this Decision (DEC) message has already been defined in the COPS specification.
3. The Edge Router now sends an RSVP-PATHTEAR message downstream to tear down the reservation. Similarly an RSVP-RESVTEAR message is sent upstream. The PATHTEAR and RESVTEAR messages have been defined in the RSVP specification and need no modification here. The purpose of these messages is to tear down the reservation and remove the reservation state from all the routers on the path between the sender and the receiver.
4. The PEP in edge router sends a COPS RPT (Report Status) message to the RCD PS indicating successful teardown of the reservation.

5. Similarly the RCD PS sends a COPS RPT message to the SCD PS. The format of this Report (RPT) message is also specific to our COPS client.

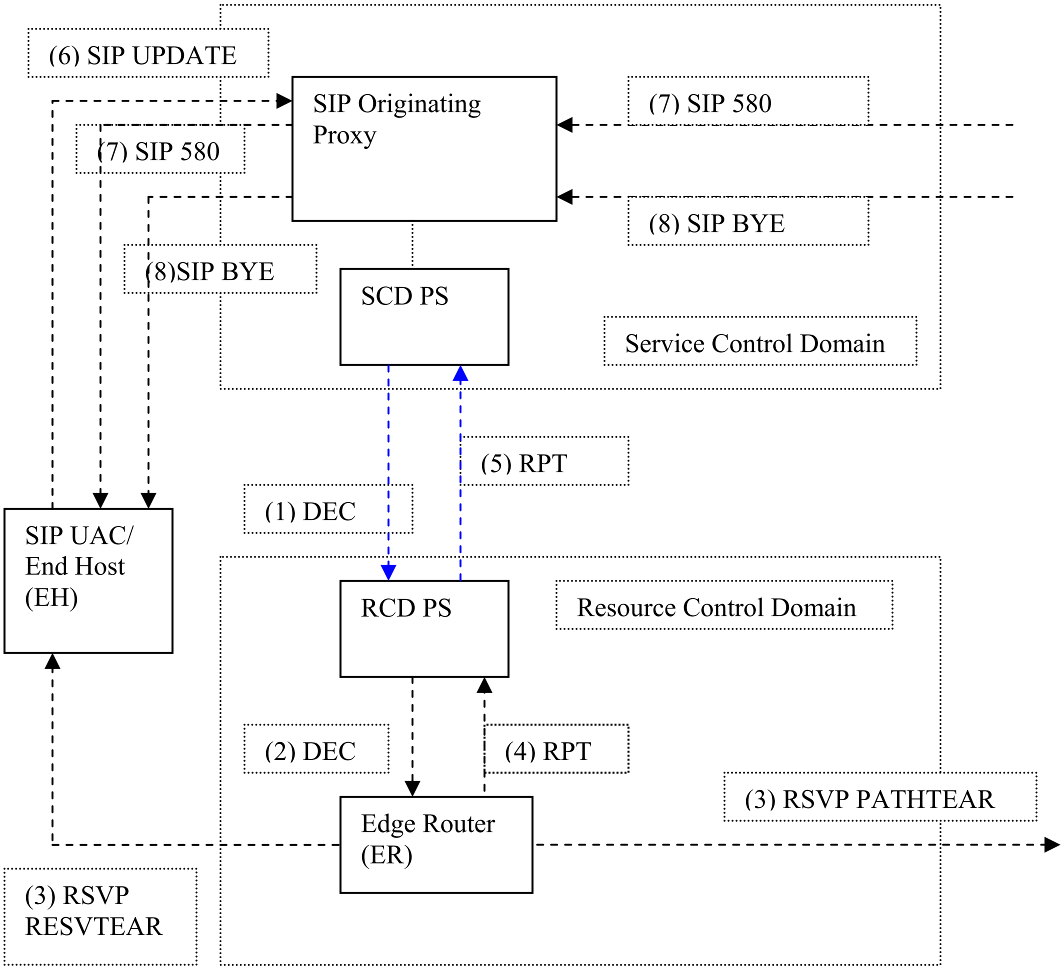

Figure 10: Sequence of messages to tear down an authorized reservation for a particular session. The tear down is initiated by the SCD Policy Server. The detailed format of the COPS messages is explained later.

6. After receiving the RESVTEAR message, the UAC is aware of the reservation teardown. Since the resource reservation level has now changed, the UAC will send an UPDATE

request to the UAS as defined in [9], Integration of Resource Management and SIP. The changed status of reservation is communicated to the other end by using the SDP parameters defined in [9].

7. If the required conditions are not met, the UAS will respond with a SIP 580 (Precondition Failure) code in the response to refuse the updated offer. The 580 response is sent to the End Host through the SIP Proxy.
8. UAS then sends a SIP BYE message. The SIP Proxy again forwards this BYE message to the End Host to terminate the session.

Thus by providing the additional link between the SCD PS and the RCD PS we allow the SCD PS to issue revocation decision for any session authorized earlier. Since the Policy Servers in both domains are now aware of each other's identity, reports of resource usage can be exchanged for accurate billing. Also, communication between the two domains minimizes the chances of token being used for more than one reservations. As we have seen, this extension does not incur any additional delays in the call establishment process and the impact on other protocols is minimal. In the following chapter we provide a detailed discussion of the COPS message content for the exchange of required messages between the RCD PS and SCD PS. New protocol objects have been defined and a new client type is introduced.

Chapter 6 COPS Usage For Proposed Extensions

For reasons discussed in the earlier discussion, COPS was selected as the protocol for message exchange between the RCS PS and the SCD PS. In this chapter we take a closer look at the message content and protocol objects that need to be defined in order to support the additional messages. A new COPS client is defined with specific meaning assigned to the COPS Request (REQ), Decision (DEC) and Report (RPT) messages.

6.1 COPS Message content

In the following discussion, PEP implies the Policy server in the Resource Control Domain (RCD) and PDP means the policy server in Service Control Domain (SCD).

6.1.1 Request (REQ) PEP -> PDP

This message is sent by the PEP to the PDP in order to inform the PDP of successful reservation. The message is sent after the PEP receives notification of successful reservation in the resource domain. The message should contain information about the session for which confirmation is being sent and information about the source and destination entities. The message also contains the ID of the PEP so that the PDP can

1. Authenticate the PEP using authentication mechanisms like digital certificates.
2. Contact the PEP at a later stage if reservation needs to be revoked.

The Request message format is as follows:

```
<Request> :: =  <Common Header>
                <Client Handle>
```

```
                    <Context = Admission Control Request>
                    <ClientSI: Confirmation Data>
                    [<Integrity>]
```

6.1.2 Decision (DEC) PDP -> PEP

This message is sent by the PDP back to the PEP. In our solution, the COPS Request only serves the purpose of informing the PDP of successful reservation. Hence, the PEP should not wait for a decision from the PDP and continue with normal operation. The PDP can, however, send an unsolicited remove Decision (DEC) message if the session, for which resources have been reserved, no longer qualifies for this reservation. The PEP in this case should implement the decision and send a COPS Report (RPT) message back to the PDP.

The PDP must send information about the session, e.g. session ID, in the decision message. The message must also have the identity of the PDP giving the decision and the source and the destination involved in this session.

The Decision message has the following format:

```
<Decision> :: =  <Common Header>
                 <Client Handle>
                 <Decision> | <Error>
                 [<Integrity>]
```

The Decision object has the following format:

```
<Decision> :: = <Context>
                <Decision: Flags>
                <Decision: Client SI data>
```

6.1.3 Report (RPT) PEP -> PDP

Using COPS Report message, the PEP can provide feedback to the PDP on events like successful termination of reservation. The report message must be sent after the PDP issues a decision. Also, the PDP can request for Report messages during the session when there is no need of new decisions. This provides for a mechanism to synchronize information between the two domains (resource and service) for billing or administrative purposes.

The Report State Message has the following format:

```
<Report State Message> :: = <Common Header>
                            <Client Handle>
                            <Report Type>
                            <ClientSI: Report Data>
                            [<Integrity>]
```

6.2 COPS Protocol Objects

The common header contained in the COPS messages has the format shown in figure 11. The actual value used for Client-type field needs to be reserved using Internet Assigned Numbers Authority (IANA) procedures. This unique COPS client identifier helps the server in providing appropriate responses for different types of clients.

0		1	2	3
Version	Flags	Op Code	Client-type	
Message Length				

Figure 11: Common header contained in each COPS message.

In order to carry the required information in the COPS messages, the following objects are defined for this COPS client. The format of all these objects is the following

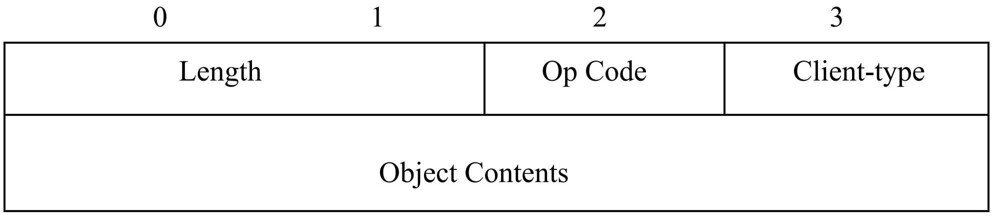

Figure 12: Format of COPS protocol objects.

Protocol Object contents are defined as follows.

6.2.1 Sender entity ID

Sender entity ID is used in the Request message and contains information about the PEP.

S-Num = 1;
S-Type = 1: IPv4 Address of the PEP
S-Type = 2: IPV6 Address of the PEP
S-Type = 3: Fully Qualified Domain Name

6.2.2 Session ID

This object is used to convey information about the session for which the request is being sent or a decision is being issued. The Session ID combined with the source and destination addresses uniquely identifies a particular reservation in the network. Session ID is a unique identifier for this session and is issued by the SCD PS while issuing a token to the end host.

S-Num = 2;

S-Type = 1: Plain ASCII Identifier
S-Type = 2: Simple Unicode string identifier
S-Type = 3: Raw Octet string identifier
S-Type = 4: NTP Timestamp as defined in RFC 1305

6.2.3 Source Address

This is the address of the source requesting resources from the network and establishing a multimedia call.

S-Num = 3;

S-Type = 1: IPv4 Address
S-Type = 2: IPv6 Address
S-Type = 3: UDP Port specification
S-Type = 1: TCP Port specification

6.2.4 Destination Address

This is the address of the destination for which resources are being requested from the network.

S-Num = 4;

S-Type = 1: IPv4 Address
S-Type = 2: IPv6 Address
S-Type = 3: UDP Port specification
S-Type = 4: TCP Port specification

6.2.5 Resources

The amount of resources reserved for this session

S-Num = 5;

S-Type = 1: Bandwidth
S-Type = 2: Flowspec specification as defined in RFC 2205

6.2.6 Reject Reason

This object can be used by the PDP to provide a reason for rejection of connection.

S-Num = 6;

S-Type = 1: User unable to afford resources
S-Type = 2: User Authentication failure
S-Type = 3: Network problem in service control domain
S-Type = 4: Unknown error

6.2.7 Sender Entity Credentials

This object contains credentials of the sender e.g. digital certificate to authenticate itself to the other end.

S-Num = 7;

S-Type = 1: DSA signature using SHA1 [X.509]
S-Type = 2: RSA signature using SHA1 [X.509]
S-Type = 3: RSA signature using MD5 [X.509]
S-Type = 4: HMAC with SHA1 [RFC 2104]
S-Type = 5: HMAC with MD5 [RFC 2104]

6.3 Client Specific Data

6.3.1 Context Object

Request type flag is 1 (Arrival of message/Admission Control). Message type is either “inform” (only to inform PDP, no immediate decision is expected) or “query” (means a decision is expected immediately).

R-Type = 1;

M-type = 1: inform
M-type = 2: query

6.3.2 Client-specific Information object for REQ message

```
<ClientSI: Confirmation Data> ::= <Sender Entity ID>
                                  <Session ID>
                                  <Source ID>
                                  <Destination ID>
                                  <Resources>
                                  <Sender Entity credentials>
```

6.3.3 Decision Client Specific Info Data Object

```
<Decision: Client SI data> ::= <Reject reason>
                               <Session ID>
                               <Source ID>
                               <Destination ID>
```

6.3.4 Client Specific Information Object for Report State Message

```
<ClientSI: Report Data> ::= <Session ID>
                            <Source ID>
                            <Destination ID>
                            <Resources>
```

6.4 Security Considerations

The extension proposed by us does not introduce any new security issues in the framework. It is recommended that the COPS client uses the Message integrity object for the authentication and validation of every COPS message. In addition COPS over TLS ([14]) can be used for secure communication between the PEP and the PDP. The suggested method in fact removes some security considerations in the original framework. Since the resource usage information is now synchronized between the two domains, reuse of an authorization token is not possible. The network can preempt the invalid reservations thus providing higher availability for authorized users.

The Message Content, Protocol Objects and the Client Specific Data defined in this chapter extend COPS to support the additional message exchange between the SCD PS and the RCD PS. All the information needed for generating these messages is either locally generated by the PEP or the PDP or is extracted from the authorization token. In the following chapter we will conclude by summarizing the solution presented in this document and the advantages of the proposed extension. In the final comments we present some ideas for future work in this area.

Chapter 7 Conclusions

Thus we have seen that the absence of communication between the RCD and the SCD Policy Servers in the original framework [5] raises many issues of practical nature. We have identified several requirements that are not fulfilled by the current mechanism. For example, after issuing the media authorization token, the Service Control Domain (SCD) loses control over the resource reservation process and can not enforce tear down of an established reservation. Absence of feedback from the RCD PS might result in inconsistent views of resource usage in the two domains. Furthermore, the media authorization token issued by the SCD PS can potentially be used, by a malicious user, in more than one Resource Control Domains. In order to overcome these problems and add greater functionality to the framework, we have proposed an extension to the framework.

This extension to the framework involves adding a communication link between the SCD PS and the RCD PS. After the resources have been reserved for a session, the RCD PS informs the SCD PS of the successful reservation. The identity of the SCD PS is extracted from the authorization token contained in the reservation request. Since the message sent by the RCD PS also contains its own identity, a trust relationship can now be established between the two entities. The network resource usage information can now be shared between the two domains and policy decisions can be implemented by the SCD. For example, on discovery of any inconsistencies or in response to the events initiated by the operator, the SCD PS can remove the reservation state for a particular session by sending a decision message to the RCD PS. The resource usage statistics can be synchronized between the domains and the resources from invalid reservations can be freed. Also, by synchronizing the resource usage information in the two domains, we have eliminated the possibility of duplicate reservations using a single token. The two domains are aware of the actual resources being reserved by the user. Such a method increases security and ensures efficient usage of resources in the network. The suggested extension thus allows greater flexibility for the network operators in providing value added services to the users.

7.1 Future Work

As mentioned earlier, an added functionality now possible with the extended framework is the sub-delegation of resources. A router or an end host can be issued bulk authorization for network resources and would then be allowed to delegate these resources independently. The end host or the router can request bulk authorization through the use of other protocols like Open Settlement Protocol (OSP) defined by the European Telecommunications Standards Institute [20]. To implement sub-delegation, the network elements (e.g. Policy Servers) will need to differentiate between a bulk authorization and a single authorization. This can be achieved through a flag in the Authorization Token indicating the level of authorization. The token must contain the identities of both the delegating entity and the end host that requests reservation. The actual mechanism to accomplish sub-delegation is a topic of future research in this area.

REFERENCES

1. R. Braden et al., "Resource ReSerVation Protocol (RSVP)", http://www.ietf.org/rfc/rfc2205.txt, RFC 2205, September 1997.

2. S.Blake et al., "An Architecture for Differentiated Services", http://www.ietf.org/rfc/rfc2475.txt, RFC 2475, December 1998.

3. "Resource Allocation Protocol (RAP)", IETF Working Group, http://www.ietf.org/html.charters/rap-charter.html

4. D. Durham et al., "The COPS (Common Open Policy Service) Protocol", http://www.ietf.org/rfc/rfc2748.txt, RFC 2748, January 2000.

5. L-N. Hamer, B. Gage, Hugh Shieh, "Framework for session set-up with media authorization", http://www.ietf.org/internet-drafts/draft-ietf-rap-session-auth-04.txt, Work in Progress, June 2002.

6. J. Rosenberg et al., "SIP: Session Initiation Protocol", http://www.ietf.org/rfc/rfc3261.txt, RFC 3261, June 2002.

7. H. Schulzrinne et al., "RTP: A Transport Protocol for Real-Time Applications", http://www.ietf.org/rfc/rfc1889.txt, RFC 1889, January 1996.

8. M. Handley, V. Jacobson, "SDP: Session Description Protocol", http://www.ietf.org/rfc/rfc2327.txt, RFC 2327, April 1998.

9. G.Camarillo, W. Marshall, Jonathan Rosenberg, "Integration of Resource Management and SIP", http://www.ietf.org/internet-drafts/draft-ietf-sip-manyfolks-resource-07.txt, Work in Progress, April 2002.

10. J. Rosenberg, "The Session Initiation Protocol UPDATE Method", http://www.ietf.org/internet-drafts/draft-ietf-sip-update-02.txt, Work in Progress, April 2002.

11. L-N. Hamer et al., "Session Authorization for RSVP", http://www.ietf.org/internet-drafts/draft-ietf-rap-rsvp-authsession-03.txt, Work in Progress, June 2002.

12. S. Yadav et al., "Identity Representation for RSVP", http://www.ietf.org/rfc/rfc2752.txt, RFC 2752, January 2000.

13. "Public-Key Infrastructure", IETF Working Group, http://www.ietf.org/html.charters/pkix-charter.html

14. Jesse Walker, Amol Kulkarni, "COPS Over TLS", http://www.ietf.org/internet-drafts/draft-ietf-rap-cops-tls-04.txt, Work in Progress, June 2002.

15. H. Sinnreich et al., "Interdomain IP Communications with QoS, Authorization, and Usage Reporting", http://www.cs.columbia.edu/sip/drafts/draft-sinnreich-sip-qos-osp-

01.txt, Work in Progress, February 2000.

16. "IP Quality of Service: An Overview", http://www.ittc.ukans.edu/~rsarav/ipqos/ip_qos.htm

17. Cisco tutorial on RSVP, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/rsvp.htm

18. H.323 versus SIP: A Comparison, http://www.packetizer.com/iptel/h323_vs_sip/

19. Policy-Based Network Architecture, http://www.allot.com/pdf/products/policymgmt.pdf

20. What is IP Telephony?. http://www.itu.int/osg/spu/ni/iptel/whatis/

21. European Telecommunications Standards Institute: http://www.etsi.org

Appendices

APPENDIX A: MEDIA AUTHORIZATION TOKEN CONTENTS

Source: L-N. Hamer et al. "Session Authorization for RSVP", http://www.ietf.org/internet-drafts/draft-ietf-rap-rsvp-authsession-03.txt, Work in Progress, June 2002.

1. **AUTH_ENT_ID:** The unique identifier of the entity which authorized the session.
    a. IPV4_ADDRESS: IPv4 address represented in 32 bits
    b. IPV6_ADDRESS: IPv6 address represented in 128 bits
    c. FQDN: Fully Qualified Domain Name as defined in RFC-1034 as an ASCII string.
    d. ASCII_DN: X.500 Distinguished name as defined in RFC-2253 as an ASCII string.
    e. UNICODE_DN: X.500 Distinguished name as defined in RFC-2253 as a UNICODE string.
    f. URI: Universal Resource Identifier, as defined in RFC-2396.
    g. KRB_PRINCIPAL: Fully Qualified Kerberos Principal name represented by the ASCII string of a principal followed by the @ realm name as defined in RFC-1510 (e.g. principalX@realmY).
    h. X509_V3_CERT: A chain of authorizing entity's X.509 V3 digital certificates.
    i. PGP_CERT: The PGP digital certificate of the authorizing entity.

2. **SESSION_ID:** Unique identifier for this session.

3. **SOURCE_ADDR:** Address specification for the session originator.
    a. IPV4_ADDRESS: IPv4 address represented in 32 bits
    b. IPV6_ADDRESS: IPv6 address represented in 128 bits
    c. FQDN: Fully Qualified Domain Name as defined in RFC-1034 as an ASCII string.
    d. ASCII_DN: X.500 Distinguished name as defined in RFC-2253 as an ASCII string.
    e. UNICODE_DN: X.500 Distinguished name as defined in RFC-2253 as a UNICODE string.
    f. UDP_PORT LIST: list of UDP port specifications, represented as 16 bits per list entry.
    g. TCP_PORT LIST: list of TCP port specifications, represented as 16 bits per list entry.

4. **DEST_ADDR:** Address specification for the session end-point.
    a. IPV4_ADDRESS: IPv4 address represented in 32 bits
    b. IPV6_ADDRESS: IPv6 address represented in 128 bits
    c. FQDN: Fully Qualified Domain Name as defined in RFC-1034 as an ASCII string.
    d. ASCII_DN: X.500 Distinguished name as defined in RFC-2253 as an ASCII string.
    e. UNICODE_DN: X.500 Distinguished name as defined in RFC-2253 as a UNICODE string.
    f. UDP_PORT LIST: list of UDP port specifications, represented as 16 bits per list entry.

   g. TCP_PORT LIST: list of TCP port specifications, represented as 16 bits per list entry.

5. **START_TIME:** The starting time for the session.
   a. NTP_TIMESTAMP: NTP Timestamp Format as defined in RFC 1305.

6. **END_TIME:** The end time for the session.
   a. NTP_TIMESTAMP: NTP Timestamp Format as defined in RFC 1305.

7. **RESOURCES:** The resources which the user is authorized to request.
   a. BANDWIDTH: Maximum bandwidth (kbps) authorized.
   b. FLOW_SPEC: Flow spec specification as defined in RFC-2205.
   c. SDP: SDP Media Descriptor as defined in RFC-2327.
   d. DSCP: Differentiated services codepoint as defined in RFC 2474.

8. **AUTHENTICATION_DATA:** Authentication data of the session authorization policy element.