Internet Engineering Task Force MMUSIC WG
Internet Draft
J.Rosenberg,H.Schulzrinne,S.Donovan
draft-ietf-mmusic-sdp-qos-00.txt
Bell Laboratories,Columbia U.,MCI Worldcom
June 19, 1999
Expires: December 1999

Establishing QoS and Security Preconditions for SDP Sessions

STATUS OF THIS MEMO

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as work in progress.

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

1 Introduction

This document discusses how network QoS and security establishment can be made a precondition to sessions described by SDP [1]. These preconditions require that the participant reserve network resources (or establish a secure media channel) before continuing with the session. We do not define new QoS reservation or security mechanisms; these pre-conditions simply require a participant to use existing resource reservation and security mechanisms before beginning the session.

In the case of SIP [2], this effectively means that the "phone won't ring" until the preconditions are met. These preconditions are described by new SDP parameters, defined in this document. The parameters can mandate end-to-end QoS reservations based on RSVP [3] or any other end-to-end reservation mechanism (such as YESSIR [4]), and security based on IPSEC [5]. The preconditions can be defined independently for each media stream.

To achieve the result of "not having the phone ring" until resources have been reserved, some have proposed adding QoS functions to application level signaling devices [6]. We do not take this approach. Rather, we feel that the QoS architecture of the Internet separates QoS signaling from application level signaling. Application layer devices (such as web proxies and SIP servers) are not well suited for participation in network admission control or QoS management, as this is fundamentally a network layer issue, independent of any particular application. In addition, since application devices like SIP servers are almost never on the "bearer path" (i.e., the network path the RTP [7] takes), and since the RTP path and signaling paths can be completely different (even traversing different autonomous systems), these application servers are generally not capable of managing QoS for the media. Keeping QoS out of application signaling also means that there can be a single infrastructure for QoS across all applications. This eliminates duplication of functionality, reducing management and equipment costs. It also means that new applications, with their own unique QoS requirements, can be easily supported.

This loose coupling works very well for a wide range of applications. For example, in an interactive game, one can establish the game using an application signaling protocol, and then later on use RSVP to reserve network resources. The separation is also effective for applications which have no explicit signaling . However, certain applications may require tighter coupling. In the case of Internet telephony, the following is an important requirement:

> When A calls B, B's phone should not ring unless resouces
> have been reserved from A to B, and B to A.

This could be achieved without coupling if A knew B's address, port,

and codecs before the telephony signaling took place. However, since telephony signaling is used largely to obtain this information in the first place, the coupling cannot be avoided.

A similar model exists for security. Rather than inventing new security mechanisms for each new application, common security tools (such as IPSEC) can be used across all applications. As with QoS, a means in application level protocols is needed to indicate that a security association is needed for the application to execute.

To solve both of these problems, we propose an extension to SDP which allows indication of pre-conditions for sessions. These preconditions indicate that participation in the session should not proceed until the preconditions are met. The preconditions we define are (1) success of end-to-end resource reservation, and (2) success of end-to-end security establishment. We chose to implement these extensions in SDP, rather than SIP [2] or SAP [8], since they are fundamentally a media session issue. SIP is session agnostic; information about codecs, ports, and RTP [7] are outside the scope of SIP. Since it is the media sessions that the reservations and security refer to, SDP is the appropriate venue for the extensions. Furthermore, placement of the extensions in SDP rather than SIP or SAP allows specification of preconditions for individual media streams. For example, a multimedia lecture might require reservation for the audio, but not the video (which is less important).

Our extensions are completely backwards compatible. If a recipient does not understand them, normal SIP or SAP processing will occur, at no penalty of call setup latency.

Others have proposed defining new SIP headers (such as pre-RING) to convey to the remote party that QoS establishment is required, followed by a re-INVITE (with ringing) to actually initiate the session (thus there is a double INVITE to actually initiate the session) [9]. Other mechanisms exist as well. A separate draft addresses the differences in these approaches.

2 Overview

The general idea behind the extension is simple. We define two new SDP attributes, qos and security. The qos attribute indicates whether end-to-end resource reservation is optional or mandatory, and in which direction (send, recv, or sendrecv). When the attribute indicates mandatory, this means that the participant who has received the SDP MUST NOT proceed with participation in the session until resource reservation has completed in the direction indicated. In this case, "not proceeding" means that the participant behaves as if they had not received the SDP at all. If the attribute indicates that

QoS for the stream is optional, then the participant SHOULD proceed normally with the session, but SHOULD reserve network resources in the direction indicated, if they are capable. Absence of the qos attribute means the participant MAY reserve resources for this stream, and SHOULD proceed normally with the session. This behavior is the normal behavior for SDP.

Resource reservation takes place using whatever protocols each participant must use based on support by their ISP. If the ISP's of the various participants are using differing resource reservation protocols, translation is necessary, but this is done within the network, without knowledge of the participants.

When the end-to-end reservation fails for a stream whose qos is mandatory, the behavior is dependent on the specific protocol which delivered the SDP. The sections which follow define these semantics for SIP and SAP.

The direction attribute indicates which direction reservations should be reserved in. If send, it means reservations should be made in the direction of media flow from the session originator to participants. In recv, it means reservations should be made in the direction of media flow from participants to the session originator. In the case of sendrecv, it means both.

In the case of security, the same attributes are defined - optional/mandatory, and send/recv/sendrecv. Their meaning is identical to the one above, except that a security association should be established in the given direction. The details of the security association are not signaled by SDP; these depend on the Security Policy Database of the participant.

3 Syntax

The formatting of the qos attribute is described by the following BNF:

```
qos-attribute = ``a=qos:'' strength-tag SP direction-tag
strength-tag = (``mandatory'' |''optional'')
direction-tag = (``send'' | ``recv'' | ``sendrecv'')
```

and the security attribute:

```
security-attribute = ``a=secure:'' SP strength-tag SP direction-tag
```

The following example shows an SDP description carried in a SIP INVITE message from A to B:

```
v=0
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
m=audio 49170 RTP/AVP 0
a=qos:mandatory recv
m=video 51372 RTP/AVP 31
a=secure:mandatory sendrecv
m=application 32416 udp wb
a=orient:portrait
a=qos:optional sendrecv
a=secure:optional sendrecv
```

This SDP indicates that B should not continue its involvement in the session until resources for the audio are reserved from B to A, and a bi-directional security association is established for the video. B can join the sessions once these preconditions are met, but should reserve resources and establish a bidirection security association for the whiteboard.

4 Usage with SIP

4.1 Overview

In the case of SIP, the caller prepares an SDP message body for the INVITE describing their desired QoS and security preconditions, and the desired directions. For SIP, send means the direction of media from originator (whichever entity created the SDP) to recipient (whichever entity received the SDP in a SIP message), and recv is from recipient to originator. In an INVITE, the UAC is the originator, and the UAS is the recipient. The roles are reversed in the response.

If the recipient of the INVITE (UAS) is capable and willing to perform the coupling (by coupling, we mean making the security and QoS establishment a precondition to session participation), it MUST return either a 180 or 183 (hereby referred to as 180/3) response containing SDP, along with the qos attribute, for each coupled stream. This SDP MUST be a subset of the preconditions indicated in the INVITE. Unlike normal SIP processing, the UAS MUST NOT alert the

called user at this point (unless the SDP in the 180/3 indicated no mandatory coupled streams).

Table 1 illustrates the allowed values for the coupling tag in the 180/3. Each row represents a value of the coupling in the INVITE, and each column the value in the 180/3. An entry of N/A means that this combination is not allowed. A value of A->B (B->A) implies that the coupling is for resources reserved (or security established) from A to B (B to A). A value of A<->B means that the coupling is for resource reservation or security establishment in both directions. The value in the response is the one used by both parties.

	B: 180 or 183			
A: INV	send	recv	sendrecv	none
send	N/A	A->B	N/A	--
recv	B->A	N/A	N/A	--
sendrecv	A->B	B<-A	A<->B	--
none	--	--	--	--

Table 1: Allowed values of coupling

Table 2 illustrates the allowed values for the strength tag in the request and response. A "Y" means the combination is allowed, and a "N" means it is not. The value in the response is the one used by both parties.

	B: 180 or 183		
A: INV	mandatory	optional	none
mandatory	Y	Y	Y
optional	N	Y	Y
none	N	N	Y

Table 2: Allowed values of strength parameter

The 180/3 is received by the UAC. The UAC should treat a 180/3 without SDP, or with SDP and without any qos parameters in any stream, as an indication that the callee is unable or unwilling to couple. As such, it should proceed with normal call setup procedures. If the 180/3 contained SDP with mandatory qos parameters, the UAC

SHOULD NOT generate local ringback (in the case of 180), or play media from the remote party (in the case of 183) until the mandatory preconditions are met.

Once preconditions are met, the UAS alerts the user, and the UAC either provides ringback (in the case that a 180 was received) or plays media from the remote party (in the case of 183), and the SIP transaction completes normally.

Note that this extension requires usage of reliable provisional responses [10]. This is because the 180/3 contains SDP with information required for the caller to initiate reservations from it towards the callee.

4.2 Details for RSVP

Assuming the callee has inserted the qos tag in the 180/3 and sent it, it should immediately start generating PATH messages for each stream it marked as send or sendrecv in the 180/3 (both mandatory and optional). When the RESV message for the stream arrives at the callee, the callee makes note of it. When RESV messages have arrived for all mandatory streams which the callee marked as send, and if the callee didn't mark any mandatory streams as sendrecv or recv, it alerts the user.

When the caller receives the SDP in the 180/3, it immediately begins sending PATH messages for all streams marked as recv or sendrecv in the 180/3.

The caller will begin to receive PATH messages from the callee for streams marked in the 180/3 as send or sendrecv. The caller SHOULD begin sending RESV messages for these streams. Reservation confirmations MUST be requested.

The callee will begin to receive these PATH messages from the caller. It should send RESV messages for these streams. Reservation confirmations MUST be requested.

The caller should either generate local ringback (in the case of 180) or media from streams it receives (in the case of 183) when the following conditions are met:

- o For all streams marked as mandatory recv in the 180/3, a RESV was received

- o For all streams marked as mandatory send in the 180/3, a reservation confirmation was received

- o For all streams marked as mandatory sendrecv, a RESV and reservation confirmation were received

The callee should begin to generate local ringback once all the following conditions are met:

- o For all streams marked as mandatory recv in the 180/3, a reservation confirmation was received

- o For all streams marked as mandatory send in the 180/3, a RESV was received

- o For all streams marked as mandatory sendrecv, a RESV and a reservation confirmation were received

These rules basically ensure that ringing occurs only after all required reservations have been made. In the case of bidirectional reservations, the call setup requires 3.5 RTT (as compared to 1.5 without any preconditions).

4.2.1 Examples

The basic message flow for a case where the caller marks a single audio stream as sendrecv, and the callee marks the 180/3 as sendrecv, is shown in Figure 1.

```
                A                              B

                ----------INV----------------->


                <------180 w/SDP---------------
                <.....PATH..B2A................

                ..........RESV.B2A............>
                ..........PATH.A2B............>

                <,,,,,,,,RESVCONF B2A,,,,,,,,,,
     LOCAL RB   <........RESV...A2B............

                ,,,,,,,,,,,,RESVCONF A2B,,,,,,>  RINGING

                <--------------- 200 OK--------

                --------------ACK ------------>
```

In the next example, there is a bidirectional audio stream from A to B. However, the caller (A) adds the qos attribute, but indicates only recv coupling. This means audio flows in both directions, but the SIP and RSVP are tied together only for reservations from B to A. The call flow is:

```
             A                             B

             ----------INV----------------->


             <------180 w/SP----------------
             <.....PATH..B2A................

             ..........RESV.B2A............> RINGING
             ..........PATH.A2B............>

  LOCAL RB   <,,,,,,,,RESVCONF B2A,,,,,,,,,,
             <--------------- 200 OK--------

             <........RESV...A2B............
             ,,,,,,,,,,,,RESVCONF A2B,,,,,,>

             --------------ACK ------------>
```

Note that in this example, the ringing at B occurs after the B to A reservation has completed. There is still an A to B reservation, but it takes place on slower time scales, and is not coupled to the SIP messaging. Similarly, A alerts the user with local ringback when the B to A confirmation has been received, but before the A to B reservation has been received.

In the next example, the caller has a single bidirectional audio stream which it marks with sendrecv coupling. However, the called party (B) doesn't understand this attribute. So, its 180/3 contains no SDP. The caller realizes that the callee doesn't understand the coupling, and proceeds with normal setup. As such, it provides local ringback immediately. The call flow is thus:

```
             A                             B

             ----------INV-----------------> RINGING
```

```
     LOCAL RB   <------180 w/o SDP-------------
                <--------------- 200 OK--------

                --------------ACK ------------>
```

5 Usage with SAP

5.1 QoS Preconditions with RSVP

In the case of send coupling, a session participant should not play out audio for a stream until resources have been reserved for it. Session senders SHOULD send PATH messages, and participants should send RESV messages for those streams. A participant should not play out audio until reservation confirmations are received. Thus, if a participant receives audio from a new source, it does not play that audio out until it has seen a PATH message, sent a reservation for it, and gotten a confirmation.

In the case of receive coupling, a session participant should not actually send audio until it has gotten at least one reservation for that audio stream. A participant should therefore send PATH messages before sending media. In most cases, since reservations are shared, a sender will only see a single RESV message anyway.

In the case of sendrecv coupling, a participant follows both procedures above.

5.2 Security Preconditions with IPSEC

Since SAP is primarily used for announcing multicast sessions, and IPSEC does not currently support multicast, a SAP session originator MUST NOT mark any streams with security preconditions.

6 SIP Extensions

There are two behaviors a UA might take when some of the mandatory pre-conditions fail. In the first case, the UA proceeds with the call anyway. In the second case, the UA attempts to terminate the call. Different applications will require differing behaviors. To support either, we define a new SIP header, called Failure-Conditions. This header contains a list of tokens, each of which indicates a normally non-fatal condition which MUST cause a failure for this request. As with other SIP headers containing lists of tokens, the header may appear multiple times in a message. The header is both a request header and response header. When used in a request, it indicates the server SHOULD return a 500 class response to the request, should any

of the indicated conditions occur. When used in a response, it indicate that the client SHOULD send a BYE for this call, should any of the indicated conditions occur.

The syntax for this header is:

```
Failure-Conditions = ``Failure-Conditions'' ``:'' 1#fconditions
fconditions = (``qos'' | ``security'' | token)
```

When the value qos is present in a request, the UAS should respond with a 500 class message if any mandatory reservation fails. When the value qos is present in a response, the UAC should send a BYE for this call should any mandatory reservations fail. When the value security is present in a request, the UAS should respond with a 500 class response if any mandatory security channel cannot be established. Similarly, if the the security value is present in a response, the UAC should send a BYE for this call should any mandatory security channels fail to be established.

When a condition is not listed, the request should not fail (or a BYE should not be sent), if the condition was a mandatory condition, and it failed.

The SIP extension SHOULD be used in conjunction with a Require header. This extension is named org.ietf.sip.fail.

There may be other extensions and mechanisms for SIP that support the SDP mechanisms described here.

7 Open Issues

There are many open issues:

- o The SIP rules assume unicast. What about multicast?

- o The SIP rules are only for the case of original INVITE. What about re-INVITEs? What about original INVITE's with no SDP at all, or SDP with no m lines?

- o How is changing of the qos and security attributes in re-INVITEs handled?

- o Are the SIP rules too complex? Should we eliminate the various send, recv, and sendrecv flavors, and make it "all or nothing"?

- The mechanism works assuming that an end system application actually sees both reservations and reservation confirmations. Is this true? Do the APIs for RSVP allow an application to know when these have occurred?

- What about usage of SDP with RTSP? Megaco?

- More details on ipsec are needed.

- How long should each party wait for reservations to succeed before giving up and aborting the call?

8 Acknowledgements

The authors wish to thank Jonathan Lennox for his valuable comments on this proposal.

9 Authors Addresses

Jonathan Rosenberg
Lucent Technologies, Bell Laboratories
101 Crawfords Corner Rd.
Holmdel, NJ 07733
Rm. 4C-526
email: jdrosen@bell-labs.com

Henning Schulzrinne
Columbia University
M/S 0401
1214 Amsterdam Ave.
New York, NY 10027-7003
email: schulzrinne@cs.columbia.edu

Steve Donovan
MCI Worldcom
1493/678
901 International Parkway
Richardson, Texas 75081
email: steven.r.donovan@mci.com

10 Bibliography

[1] M. Handley and V. Jacobson, "SDP: session description protocol," Request for Comments (Proposed Standard) 2327, Internet Engineering

Task Force, Apr. 1998.

[2] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: session initiation protocol," Request for Comments (Proposed Standard) 2543, Internet Engineering Task Force, Mar. 1999.

[3] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation protocol (RSVP) -- version 1 functional specification," Request for Comments (Proposed Standard) 2205, Internet Engineering Task Force, Sept. 1997.

[4] P. P. Pan and H. Schulzrinne, "YESSIR: A simple reservation mechanism for the Internet," (Cambridge, England), July 1998. also IBM Research Technical Report TC20967.

[5] S. Kent and R. Atkinson, "Security architecture for the internet protocol," Request for Comments (Proposed Standard) 2401, Internet Engineering Task Force, Nov. 1998.

[6] P. Sijben and M. Buckley, "Establishing and controlling end-to-end qos in tiphon systems," (tiphon temporary document), ETSI Tiphon, May 1999. 13TD109.

[7] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," Request for Comments (Proposed Standard) 1889, Internet Engineering Task Force, Jan. 1996.

[8] M. Handley, C. Perkins, and E. Whelan, "Session announcement protocol," Internet Draft, Internet Engineering Task Force, June 1999. Work in progress.

[9] P. Goyal, A. Greenberg, C. Kalmanek, B. Marshall, P. Mishra, D. Nortz, and K. K. Ramakrishnan, "Integration of call signaling and resource management for ip telephony," vol. 13, May/June 1999.

[10] J. Rosenberg and H. Schulzrinne, "Reliability of provisional responses in SIP," Internet Draft, Internet Engineering Task Force, May 1999. Work in progress.