Interaction of Call Setup and Resource Reservation Protocols in Internet Telephony

Henning Schulzrinne
Columbia University

Jonathan Rosenberg
Bell Laboratories

Jonathan Lennox
Columbia University

June 11, 1999

Abstract

In the Internet, call signaling, security association and resource reservation are handled by separate protocols and likely traverse different paths. However, for reliable service, the three functions may need to be coupled during call setup. We describe and compare several approaches to coupling, based on either single-phase setup or two-phase setup mechanisms. Our discussion is based on the Session Initiation Protocol (SIP), but also applies to other signaling protocols with similar properties.

1 Introduction

Before establishing a basic phone call, at least three things have to happen in both Internet-based and circuit-switched telephone systems: the called party has to be alerted, the path of voice bits has to be established and resources may have to be set aside for this call. In the circuit-switched network, the signaling protocol, such as SS7, handles all three aspects. In Internet telephony, responsibility for the three functions is divided between the session establishment protocol such as SIP [1, 2] or H.323 [3], the routing protocol such as BGP [4][1], and a resource reservation protocol such as RSVP [5].

In a circuit-switched telephone call, the three signaling actions are interleaved, so that the phone does not ring until there is a path available to the caller. Indeed, a one-way backward voice channel from called party to calling party is available as soon as the phone rings at the called party. If desired, the calling switch can send a test tone to the remote end, and receive confirmation of a working bidirectional connection.

SIP currently intentionally does not involve itself with the reservation of resources. This is necessary, since the paths of SIP and resource reservation messages generally only coincide at the end points. [2]

There are several possibilities of how to handle resource reservations in IP telephony calls. The easiest solution is to avoid per-call resource reservation altogether, and simply rely on class-based service differentiation [6]. Secondly, if per-call resource reservation is desired, the reservation can be established as soon as the remote IP address and media types of the other side is known, but without interfering with further call progress. The caller knows the callee's media destinations generally after receiving a 200 OK message, but the information can also be included in a provisional response such as 180 (Ringing). This method is acceptable only if the probability that the resource reservation fails is very low or if the participants prefer to use best-effort service to receiving a "fast busy" signal, requiring them to defer the call. The third approach is to interleave signaling and resource reservation, so that the callee's phone only rings once resources have been successfully reserved.

[1] Here, We are glossing over the differences between circuit routing and datagram routing.

[2] Indeed, it is quite possible that they do not even coincide there, as the signaling end point can direct media packets to another IP address.

Recently, it has been proposed [**?**] to use the Session Description Protocol [7] for indicating which media streams should reserve resources. It also allows to indicate whether resource are to be reserved for incoming or outgoing data and whether success of the reservation is mandatory or optional. An example of an SDP message extended by these parameters is shown in Fig. **??**. The example shows the SDP contained in the INVITE that Alice sends to Bob. Here, Bob should only participate in the audio session if the audio data from Bob to Alice can use reserved resources. For the video stream, Alice would like bidirectional reservation, but wants to continue the call even if reservation fails to find the necessary resources. The example also demonstrates that similar considerations apply to setting up security associations.

Embedding resource reservation information into the session description has the advantage that it can become part of the negotiation. For example, the callee may decide that he doesn't want to pay for resource reservation, or only for certain media. Table 1 lists the possible combinations of sender and receiver QOS indications and the ensuing resource reservation direction. $A \rightarrow B$ means that data flowing from caller A to callee B uses reserved network resources. It may also be helpful to indicate the available resource reservation protocols, in case there are several alternatives.

	B: 180 or 183			
A: INVITE	send	recv	sendrecv	none
send	N/A	$A \rightarrow B$	N/A	–
recv	$B \rightarrow A$	N/A	N/A	–
sendrecv	$A \rightarrow B$	$B \rightarrow A$	$A \leftrightarrow B$	–
none	–	–	–	–

Table 1: Resource reservation negotiation (N/A: not allowed)

We assume that all of the proposals below make use of this method for end-to-end resource reservations.

```
v=0
o=alice 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP Seminar
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
```

Figure 1: Example of an SDP message extended by QOS indications

So far, we have silently assumed that resources are reserved end-to-end. However, it is also possible to imagine *segmented resource reservation*. Here, resources are reserved "close to" the end system, but not end-to-end. For example, resources may be reserved within the same subnet, the same AS or the same network provider. Segmented resource reservation has the advantage that it allows both sides to deploy resource reservation independently. Also, if the reservation stays local, there may be less of a need for strong authentication, since it is easier to assure the validity of IP addresses, for example. On the other hand, RSVP messages can be tunneled so that only willing nodes need to participate in the resource reservation. There is currently no generally accepted segmented resource reservation protocol, although one could imagine using RSVP for this, by simply setting the PATH destination address to a virtual end point at the egress router of the zone.

2 Requirements

Internet telephony systems have to satisfy at least the same service expectations as the current telephone service – retraining 200 million telephone customers is not a particularly viable option. The following requirements are given. For "caller", we also use the terms "originating station" or M_O; for "callee", "terminating station" and M_T.

The basic requirements was described above: the called party is alerted only once the resource reservation has succeeded. We summarize the requirements below, sorted roughly in order of importance:

Resource reservation: While Internet delays and packet loss may, on average, be sufficient for Internet telephone calls, having to suspend a call until the "Internet weather clears up" is not acceptable. Thus, we assume that resources have to be reserved at least for a subset of the media within a call. (Reservation can imply either per-flow or per-aggregate reservation. For simplicity of presentation, our examples in this note assume RSVP [5].)

The necessity of resource reservation is based on the assumption that, during overload periods, b% call blocking is preferable to an equivalent amount of packet loss. For b up to around 10%, this is probably not true, but once the overload exceeds that amount, all calls would get degraded to uselessness. (Clearly, the overload case only occurs when all media streams have been reduced to the minimal acceptable quality since, generally speaking, a blocked call is more annoying than a mobile-phone-quality call.)

There is a slight additional advantage to allowing modest quality degradation, in that it "increases" the call capacity of the system, as opposed to simply moving calls to a later time or a competitor.

No call defects: Unless the callee specifically desires otherwise, a call should not ring, be picked up and then fail due to lack of available network resources. This requires that network resources are reserved *before* the callee picks up.

No charge for incomplete calls: There should be no charge for incomplete calls, i.e., including the time that the callee phone rings. However, given the previous requirements, this implies that the network will reserve resources before media data flows and before the caller is charged for the reservation. We refer to the point in time where resource usage is billable as the *commit time*. Unless care is taken, this may allow two conspiring end systems to block network resources for an extended period of time, thus effecting a denial-of-service attack.

Post-pickup delay: After the terminating station picks up the receiver, a voice path to the caller must be available almost immediately, as otherwise the "hello" greeting of the callee may get truncated or dropped. As a guideline, a voice path must be available within about 100 ms of the time that the callee picks up the receiver. This time limit is based on the use of speaker phones or mobile phones, where the called party may speak almost immediately after hitting the "talk" button. Clearly, with traditional phones where a receiver needs to be lifted to one's ear, the post-pickup delay can be much longer.

Since the 200 response must traverse the proxy servers that the original INVITE request used, the delay of the 200 response may well exceed this bound of 100 ms and may well arrive after the first few voice packets.

Announcements: The originating station needs to be able to receive audio announcements before ringback, in particular when connecting to the PSTN. It is desirable that announcements can make use of reserved resources rather than being carried best-effort. If they are carried using reserved resources, the caller should not pay for them. In some approaches, no voice packets are allowed to reach the caller until the caller has committed the reserved resources.

Accounting and preventing theft of service: A party must not be able to send IP packets using reserved resources without the packets being accounted for. The ease of accounting is subject to debate. It appears possible to simply use RADIUS [8, 9] for this purpose. In that scenario, a router or other device simply sends an accounting request (`Acct-Status-Type`, type "start") when an RSVP RESV message or similar resource reservation request has been received. When the reservation is terminated, the device sends an `Acct-Status-Type` parameter of type "stop". The following parameters then provide the necessary information:

`Acct-Input-Packet`	number of packets received
`Acct-Output-Packets`	number of packets sent
`Acct-Input-Octets`	number of bytes received
`Acct-Output-Octets`	number of bytes sent
`Acct-Session-Time`	duration of active session

(This is already implemented for voice calls, for example, in the Cisco 2600/3600 router.)

The authors believe that per-byte rather than per-minute charging is appropriate for Internet voice and multimedia services, since the relationship between time and number of bits transmitted can change drastically depending on the codec, audio silence suppression and the use of VBR video.

Note, however, that in addition to volume-based charges, a per-minute charge to account for holding reserved resources may be necessary to prevent users from blocking resources without penalty.

Ability to charge by the minute: It appears desirable to allow users to be charged for reservations by the minute, with the charge depending on the nominal reservation bandwidth, in addition to by byte or packet volume. Time-based charging is easier for users to understand and avoids the problem that similar phone calls generate widely diverging charges, depending on the activity of silence suppression. (On the other hand, per-byte charges encourage the use of silence detection and will yield reasonably predictable charges if either caller or callee pays for media flowing in *both* directions. For "dutch treat" charging, it encourages listening rather than talking...)

Segmented QOS support: In some network architectures, only the access networks support resource reservation, possibly different ones for both sides. We cannot rely on having resource reservation requests be visible to the intermediate network. However, there does not appear to be any reason for intermediate networks to drop RSVP or other reservation protocol requests, even if it does not act upon them. (Indeed, it is not clear why an access network provider should choose to contract with such a network violating the most basic of service level agreements, namely to carry valid IP packets end-to-end. Such a policy would likely interfere with the deployment of resource reservation.)

Below, we describe a number of signaling approaches allow coupling between resource reservation and session signaling. The single-phase setup in Section 4 handles resource reservations as part of a single SIP transaction. Sections 5 and 6 let the final response bypass proxies to avoid the call defect problem. Section 7 describes the use of a modified INVITE request for reserving resources, while Section 8 introduces a new request, RESERVE.

3 H.323

H.323 [3] has a similar problem, with the details depending on the version of the protocol. In version 1, the H.245 exchange containing information about the addresses of the end points and the media types follows the H.225.0 Q.931 exchange, i.e., occurs after ringing. With the fast-connect procedure of version 2, the H.245 description is ???.

4 Single-Phase Setup

As shown in Fig. 2, the caller issues an INVITE request, which reaches the callee through zero or more proxies. The session description contains an indication that the caller wishes to reserve resources, as described in the introduction.

The callee responds with an "18x Reserving Resources" message, which contains its resource reservation indication.

Next, both sides exchange resource reservation requests. For RSVP and the common case of bidirectional reservation, the caller and callee each send a PATH message and return a RESV message upon receipt of the PATH message. If a party receives a RESV message, it knows that the outgoing media streams has its resources reserved. However, it needs to request a reservation confirmation to find out if the media data it will be *receiving* has succeeded in its reservation request.

Caller and callee have the advantage of knowing all acceptable codecs, and thus can make informed assumptions about the bandwidth to be reserved. For RSVP, each party can submit revised PATH requests should an earlier one fail.

When the callee is assured that the data path has been reserved in both directions, it responds with a "180 Ringing" messsage and starts alerting the callee. When the "180 Ringing" message arrives at M_O, the caller hears locally-generated ringback tone. The ringback tone stops when M_O receives a 200 OK or receives the first media packet from the callee, whichever comes first. The details of this scheme are described in [**?**]. Note that the 18x resource reservation request has to be delivered reliably [10], as otherwise the negotiation mechanism would fail.

In this and all schemes below, the resource reservation has to time out if it does not get committed. Otherwise, malfunctioning or malicious end systems could reserve resources indefinitely and block other callers from using the network. Also, the network will likely have to impose a per-subscriber limit of reserved but uncommitted resources.

There are at least two mechanisms that could be used to commit resources and start call charges:

RSVP extension: The resource reservation protocol could be extended by a commit message. The caller sends the commit message upon receipt of 200 or the first audio packet. Audio packets that arrive at a router with reserved, but uncommitted resources are treated as best-effort, to prevent fraud. This solution still leaves the first few voice packets vulnerable, as the commit message is always one round-trip time behind the first packet arrival.

First media packet: Routers could commit to resources and charge for them as soon as the first data packet arrives.

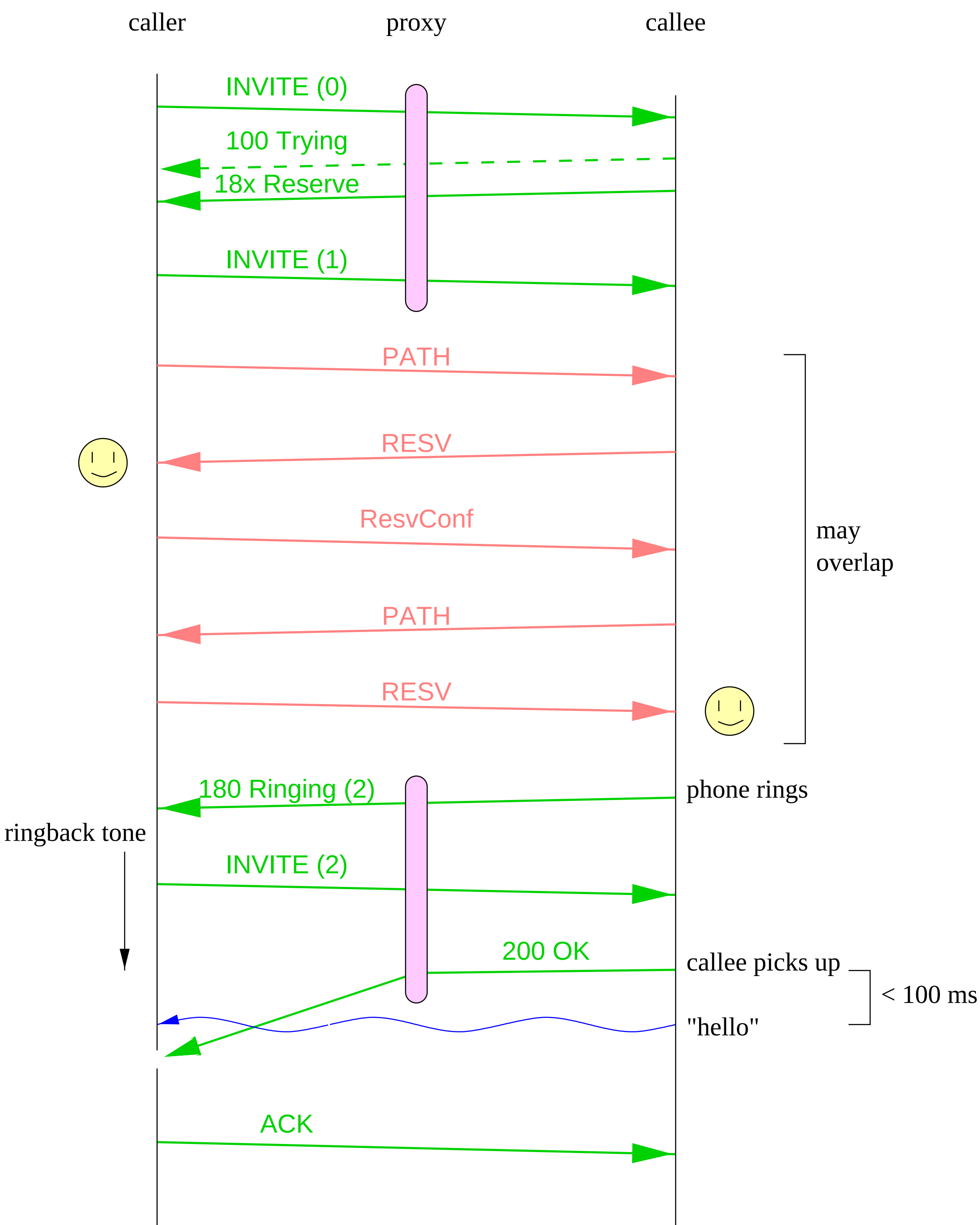

Figure 2: Single-phase call setup

4.1 Disadvantages

In the scheme described above, to avoid the delay in answering caused by a slow 200 response, incoming media packets must stop the playing of ring-back at the caller. This appears feasible, but can be awkward to implement, particularly in a multimedia call. For example, it may be desirable to stop ringback when the first video packet arrives, but the video packets may reach a completely different application than the one playing back ringback tone, so that some form of coupling between applications is needed. Even for audio only, the ringback playback may be done by a different type of application than receiving network (RTP) packets. For example, in the current Mbone-style tools, ringback is generated by playing an audio file to the sound device. Fortunately, media tools like rat [11] or nevot [12] have a "conference bus", where messages indicating talker activity are being distributed to a local multicast group, so that this coupling is possible.

There are a number of ways that routers can detect and signal the commitment of resources. Their feasibility depends on the ability to adjust router behavior. For example, the first media packet to make use of a filter spec could trigger a message from the packet engine to the control processor, which could timestamp the event. This event would then, e.g., via DIAMETER, signal the beginning of the chargeable time. It may not even be necessary to report events in real-time. If not, a simple hardware addition could record the timestamp of the first packet, with a periodical "sweep" process gathering the data. Alternatively, the initial packets could be marked by the router alert option, which would commit the resources, at the cost of greater delay and processing overhead. Alternatively, the router can simply count packets or bytes and pass this information to the accounting systems, e.g., using RADIUS, when the reservation is terminated. (RADIUS records [9] already contain byte counts.)

As described, all single-phase setup mechanism, including the ones described below in Section 5 and 6, have the disadvantage that for segmented QOS, the callee does not know whether the caller has succeeded in reserving resources. Thus, it might ring, but the call can still fail. There are two ways to prevent this problem. First, if provisional responses are transmitted reliably end-to-end, rather than hop-by-hop, as in [10], the callee could send another 18x response, "quality assured", which the caller acknowledges only if its resource reservation has succeeded. If the caller's reservation failed, it terminates the call with a BYE instead. Alternatively, the caller can delay confirmation of the 18x (Reservation Requested) message until its reservation has succeeded, but this would just delay the start of the M_T resource reservation. This approach has an additional disadvantage, in that delay in completing the caller's resource reservation causes the first proxy to retransmit the 180QA message.

Counting all messages required for confirming receipt of RPRs, the first mechanism requires nine messages. We can save two messages by starting ringback upon receipt of the 180QA response rather than sending another 18x (Ringing) response.

5 Call Setup with Express 200

This approach bends the SIP rules for sending 200 responses. Instead of sending the 200-response via proxies, the UAS sends it directly to the UAC indicated in the Contact header. This behavior is indicated by a Require header. Once the 200 response arrives, the UAC takes the appropriate action to commit network resources and allow the flow of QOS-assured media packets.

Having the 200 response bypass proxies causes searches at proxies to continue, so the UAC has to send a CANCEL request unless it wants to receive spurious final responses. There is still the possibility of a race condition, where a proxy times out and passes an error response (300 or greater) upstream before the CANCEL reaches it. Thus, the UAC has to be prepared to acknowledge, but otherwise ignore, this response. Since 200 responses are retransmitted end-to-end rather than hop-by-hop, there is no danger of generating extranous responses.

This mode of operation takes at least five messages: the standard three messages for the call setup plus two messages for the CANCEL request. More likely, M_T sends both 100 and 180 messages, for a total of seven messages.

There exists another race condition, where the timeout occurs before the express-200 reaches the UAC. However, this condition can occur even in the current version of SIP. Note that even sending a CANCEL request and waiting for a response before declaring a time-out will not prevent this race as the CANCEL response is hop-by-hop. For these hopefully rare circumstances as well as for any unknown call identifier, a UAC should probably simply send a BYE request to avoid being hit with the 200 retransmissions.

This mechanism breaks services that rely on proxy servers maintaining call state, including firewalls, NATs and ACDs. If a proxy inserts a Record-Route header, it won't see the 200, but will see an ACK. This will probably still work, but proxies might just drop it as a spurious ACK.

Thus, given the problems enumerated, this solution is unlikely to be satisfactory.

6 One-Phase Call Setup with Dual 200

This mechanism operates like the single-phase setup, but avoids the delay of routing 200 responses through proxies by sending *two* 200 responses, an "express" 200 and a"local" 200. The express-200 travels directly to the address contained in the Contact header of the INVITE request[3], bypassing any proxies. The local-200 travels the usual path indicated by the Via headers, making stops at all proxies. The first one terminates ringback and may trigger messages that allow audio to flow to the caller. The local-200 is ignored and simply serves to stop searches at intermediate proxies. The caller sends an ACK in the normal manner for the local-200. The UAS retransmits local-200's until the ACK has been received. Only the local-200 is retransmitted since the difference in delay makes no difference if the express-200 has been lost.

This scheme requires that the UAC can distinguish the two different 200 responses, so that it acknowledges only the local-200. Possibilities include the use of a different status code, assuming that it would only be used for the express-200 or a special header field.

If the local-200 were lost, it could result in searches not being terminated. Compared to the "express-200" scheme above, it avoids the need for sending a CANCEL request and thus only takes four messages.

This scheme recovers somewhat faster compared to the other schemes when the (express) 200 is lost, as the local-200 will likely follow within the processing time of the intermediate proxies. This is likely to be shorter than the 500 ms first retransmission interval for the other one-phase and the two-phase proposal.

Another variation on this theme is to send a special provisional "200 on the way" response that bypasses proxies.

7 Two-Phase Setup with Pre-Ring INVITE

It has been proposed [**?**] to solve the reservation problem via a two-phase call setup mechanism, illustrated in Fig. 3. (Our description here differs in the details from the proposal in [**?**] to increase its generality.) M_O generates two INVITE/ACK cycles. The first, called pre-ring, causes M_O and M_T to initiate resource reservation. The INVITE request is distinguished by a Request-Disposition token and/or a Require header from a regular INVITE request.

If end-to-end resource reservation is desired, the same SDP indication is used as in the one-phase cases. Even for segmented QOS, however, M_O has to wait for the 200 response from M_T, as setting up the filter likely requires knowing the destination IP address.

[3]The Contact header is now suggested for INVITE, but will become mandatory when SIP moves to Draft standard status.

Once M_O has succeeded with its resource reservation, it sends a second INVITE request. The second INVITE is sent directly to M_T, identified in a Contact header contained in the first 2xx response, bypassing any proxies[4]. Once M_T has succeeded with its reservation, M_T then starts ringing and sends a 180 (Ringing) response to trigger ringback at M_O.

To improve backward compatibility with SIP systems that only use a single-phase INVITE, we suggest that systems using the two-phase approach return a different success status, say 201 (Created) for the non-ringing INVITE. That way, M_O will know immediately whether it is dealing with a one-phase system, even if the 180 Ringing response gets lost. Otherwise, it would have to use a Require header, adding an additional round-trip time when connecting to single-phase systems.

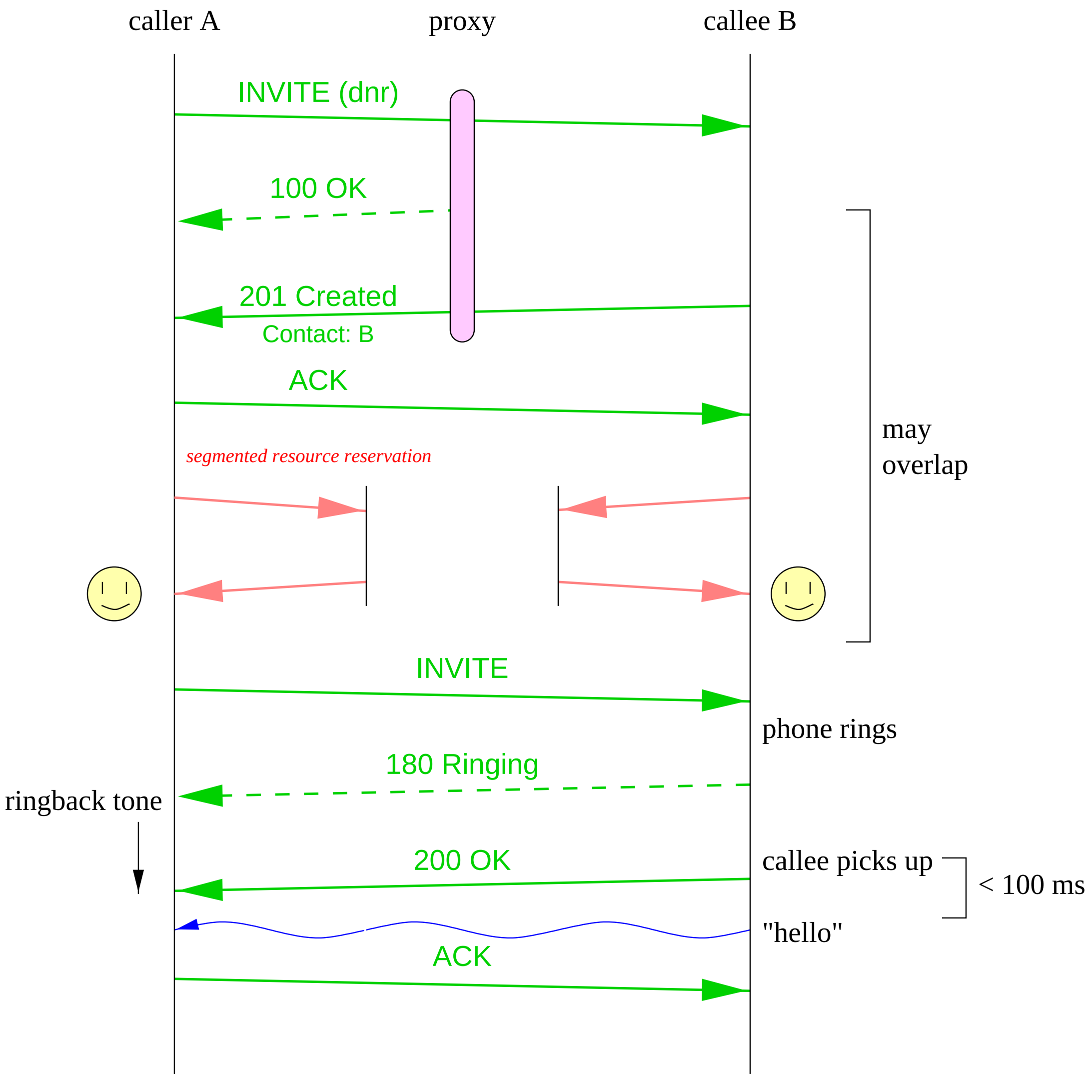

Figure 3: Two-phase setup with INVITE request

[4]unless, of course, a proxy has insisted upon being included by inserting a Record-Route header.

7.1 Advantages

The scheme does not *require* RPRs since the reservation negotiation is wrapped into a standard reliable INVITE exchange. Its message complexity is as low as the most message efficient one-pass schemes, with seven messages. The message complexity increases to eight if a reliable ringing indication is desired. If the 180 is lost without RPR, the caller would hear nothing but silence until the callee picks up – which would likely lead him to abandon the call.

Since the second INVITE request bypasses proxies, it imposes a somewhat lower message processing burden on the signaling infrastructure and may slightly reduce the post-dial delay.

Also, since the invitation is broken into two separated transactions, the time that a proxy has to maintain transaction state is reduced.

7.2 Disadvantages

The scheme has a number of potential disadvantages:

No-answer handling: Since the second INVITE is not seen by a proxy, a no-answer (CFNA) situation needs to be handled by the end system. This is somewhat inconvenient, as it requires that no-answer scripts have to be executed by the end system rather than a terminating proxy. (For one-phase SIP, the same script can handle the busy and no-answer case.) For call-forward busy, the 600 response usually occurs on the first INVITE, but if the "busy" response is triggered by a human being, it won't occur until the callee's phone rings.

The CFNA and CFB problems can be avoided to some extent if the outbound proxy forces itself into the signaling path via a Record-Route header. This works only, however, if the no-answer handling does not get confused by seeing the first INVITE succeed and thus assumes that the call has been answered. Scripts not designed to be operating in this two-phase environment may well excuse themselves from handling subsequent INVITE requests in the same call, as is possible using the sip-cgi specification [13], and may thus not operate correctly.

Lost 200s: If the response to the ringing INVITE is lost, M_O will not be able to commit resources in a timely manner. Either media data from the callee will be dropped or receive only best-effort treatment. This problem is less likely if these responses can share the same resource reservation as the actual media data.

Record-Route headers: If proxies insert themselves into the signaling path via Record-Route headers, it may not be possible to meet the 100 ms post-pickup delay target. Unless the end of ringback is triggered by arriving audio packets instead of the SIP 200, this behavior cannot be avoided.

Interoperation with "classic" SIP: A SIP UAS that does not support the "do-not-ring" request disposition will treat the first INVITE like a regular, ringing INVITE and alert the user. Unless the request contains a Require header or the response is something other than 200, the caller has no way of knowing that, however, and will not enable voice data until the second INVITE request. If a Require header is used, call setup is delayed by an additional round-trip time, as the caller has to issue another, "plain" request. The use of the 201 (Created) status response is probably simpler to implement and has lower message overhead.

8 Two-Phase Setup with RESERVE Method

We can modify the approach described in the previous section by using a different SIP method for the pre-ring request, RESERVE, as shown in Fig. 3. The response to the RESERVE request contains the Contact header for the end system. The RESERVE request busies the callee phone, but is time-limited. The end system will need to assure that a single malicious caller does not repeatedly send RESERVE requests, but never follows through with an INVITE, thus blocking the callee if it acts as a "single-line" phone. It is probably safer not to block the phone itself from receiving other calls in this state. This will, on occasion, cause the INVITE request to return a busy indication, but that seems harmless.

Since proxies are required to treat all unknown methods as non-INVITE methods, this does not introduce any backward compatibility issues. A "classic" SIP UAS simply returns a 501 (Not implemented) response.

Compared to using a pre-ring INVITE as in Section 7, this approach reduces the number of messages exchanged since no ACK is required. Compared to returning 201 in the previous approach, it does, however, increase the call setup delay for systems without this extension.

Having two different methods is also less likely to confuse proxy servers that want to implement no-answer services. Such a server would insert itself into the path of the second INVITE by placing a Record-Route header into the RESERVE request, albeit at the risk of missing the 100-ms window.

In summary, this approach appears to be cleanest from a protocol perspective, since it avoids overloading the same method with different semantics.

9 Announcements

There are two possibilities for handling announcements (such as "The number you have dialed has been changed. The new number is . . . ") from the PSTN side:

1. Have the caller open up its voice network connection upon receiving either the first 1xx response or a specific 18x response.

2. Since the call will not complete to the original callee, return a "200 OK", and treat it like a normal successful call. (However, the session description returned would not list any media types or only send-only entries, if RTCP ports need to be specified.) As a variation, one could also return a different 2xx response, to indicate to the caller that the voice heard is not that of the callee.

The second approach has the advantage that it is backward compatible with any existing SIP client, even those that do not understand session descriptions in 1xx responses. It also works much better with proxies, as it will terminate searches. (It would be rather confusing if the caller heard both a message indicating failure of the call, mixed with the callee picking up or another message from a different branch.)

For announcements, the same resource reservation mechanism as for end-to-end calls can be used, with the additional caveat that the flow of voice data should not trigger charges. For end-to-end reservation, this can be accomplished by having the PATH message include an appropriate authentication data policy element [14] that is recognized as a no-charge client by the intermediate routers.

However, for a gateway into the PSTN, the resource reservation would likely take place before the call is placed on the PSTN side. Thus, the resource reservation cannot take into account that the call will be answered by an announcement. (This information is available in ISUP and [?] ISDN messages.) Unless announcement calls are removed from the billing database after the fact, the only feasible solution appears to be an update of the PATH message as soon as the gateway recognizes that it will be handling an announcement.

Scenarios where a call is switched between a progress announcement and a paying call require significantly more effort if resource reservation is used. In those cases, it may well be easier to at least logically transfer the call.

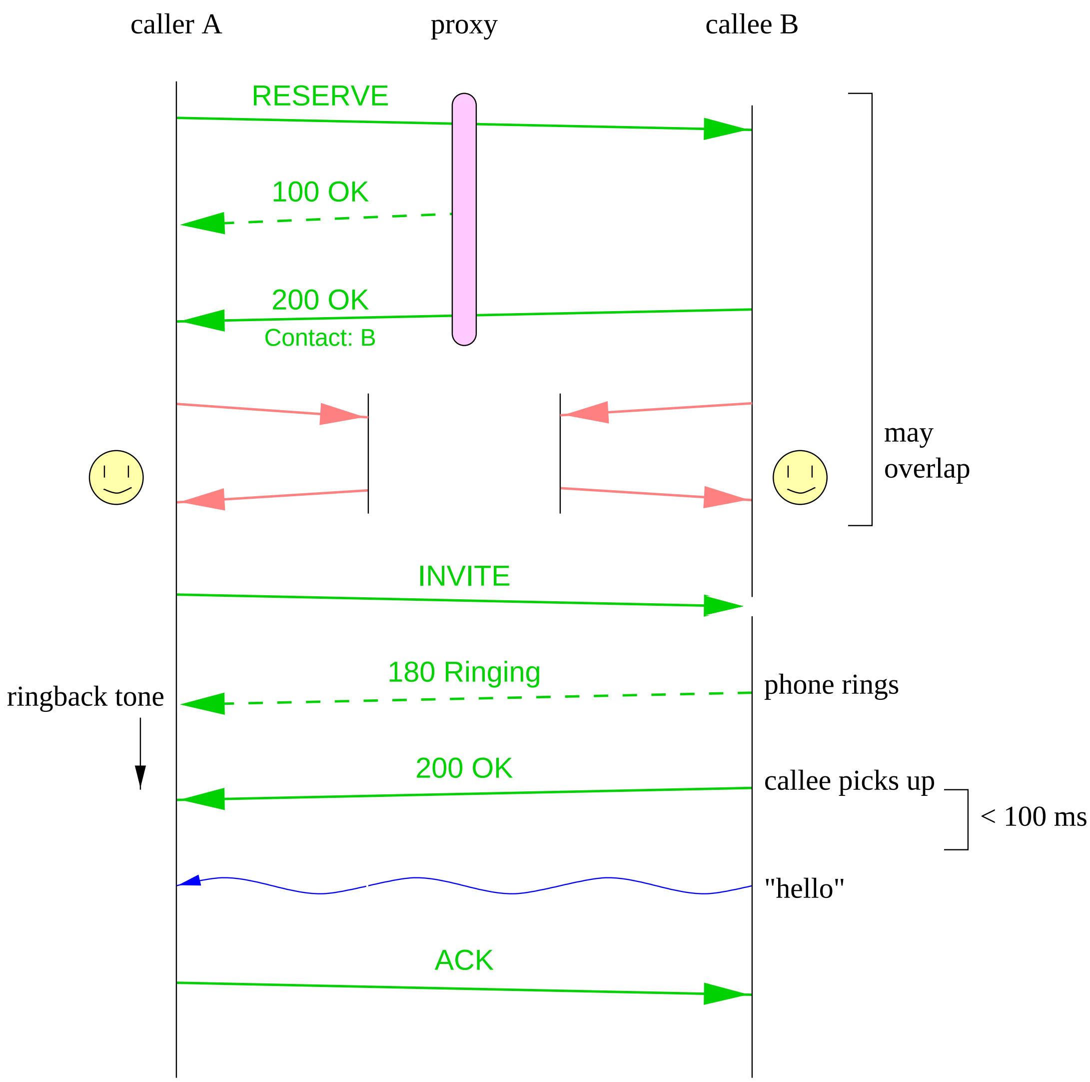

Figure 4: Two-phase setup with RESERVE request

References

[1] H. Schulzrinne and J. Rosenberg, "Internet telephony: Architecture and protocols – an IETF perspective," *Computer Networks and ISDN Systems*, vol. 31, pp. 237–255, Feb. 1999.

[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] International Telecommunication Union, "Visual telephone systems and equipment for local area networks which provide a non-guaranteed quality of service," Recommendation H.323, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, May 1996.

[4] Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," Request for Comments (Draft Standard) 1771, Internet Engineering Task Force, Mar. 1995.

[5] 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.

[6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated service," Request for Comments (Proposed Standard) 2475, Internet Engineering Task Force, Dec. 1998.

[7] M. Handley and V. Jacobson, "SDP: session description protocol," Request for Comments (Proposed Standard) 2327, Internet Engineering Task Force, Apr. 1998.

[8] C. Rigney, A. Rubens, W. Simpson, and S. Willens, "Remote authentication dial in user service (RADIUS)," Request for Comments (Proposed Standard) 2138, Internet Engineering Task Force, Apr. 1997.

[9] C. Rigney, "RADIUS accounting," Request for Comments (Informational) 2139, Internet Engineering Task Force, Apr. 1997.

[10] J. Rosenberg and H. Schulzrinne, "Reliability of provisional responses in SIP," Internet Draft, Internet Engineering Task Force, Nov. 1998. Work in progress.

[11] A. Sasse, V. Hardman, I. Kouvelas, C. Perkins, O. Hodson, A. Watson, M. Handley, J. Crowcroft, D. Harris, A. Bouch, M. Iken, K. Hasler, S. Varakliotis, and D. Miras, "Rat (robust-audio tool)," 1995.

[12] H. Schulzrinne, "Voice communication across the Internet: A network voice terminal," Technical Report TR 92-50, Dept. of Computer Science, University of Massachusetts, Amherst, Massachusetts, July 1992.

[13] J. Lennox, J. Rosenberg, and H. Schulzrinne, "Common gateway interface for SIP," Internet Draft, Internet Engineering Task Force, Nov. 1998. Work in progress.

[14] S. Yadav, R. Yavatkar, R. Pabbati, P. Ford, T. Moore, and S. Herzog, "Identity representation for RSVP," Internet Draft, Internet Engineering Task Force, Mar. 1999. Work in progress.