Unified Messaging using SIP and RTSP

Kundan Singh and Henning Schulzrinne
Columbia University
{kns10,hgs}@cs.columbia.edu

Abstract—**Traditional answering machines and voice mail services are tightly coupled to Plain Old Telephone Systems. It is hard to provide simple services like forwarding voice mail outside the local PBX. With the advent of Internet telephony, various Internet based voice mail service providers are likely to come up. We propose a multimedia mail architecture using the existing Internet protocols, in particular SIP (Session Initiation Protocol) and RTSP (Real-Time Streaming Protocol). We discuss the design and implementation aspects of our prototype system and the protocol issues in using SIP and RTSP.**

Keywords—**Voice mail, video mail, unified messaging, SIP, RTSP, Internet telephony.**

I. Introduction

Current voice mail systems are closed. It is hard to perform simple operations, like forwarding voice mail outside the local PBX, filtering or sorting messages. Configuration is tedious, e.g., one can not readily switch between a set of outgoing messages. Moreover, voice mail and call answering services are implemented as stand-alone proprietary systems. The service must be provided by the PBX, local phone company or the local handset or one must obtain a separate voice mail number. Getting a third-party voice mail service is difficult.

On the other hand, Internet protocols, such as electronic mail for internet messaging and SIP (Session Initiation Protocol [1], [2]) for Internet telephony, have an open architecture. Configuration is simple and the protocols are extensible. There is a clear separation between the internet service provider and the messaging or telephony service provider. Internet telephony is likely to replace the old telephone systems in near future. So, it is important to design a voice mail system for Internet telephony.

We have built an Internet voice/video mail system using existing Internet protocols, in particular, SIP and RTSP (Real-Time Streaming Protocol [3]), that allows users message access from any Internet connected device, using standard media players or user agents. SIP is used for setting up multimedia calls over Internet. RTSP controls the delivery of streaming media and provides facilities to play back, record, or perform other operations on multimedia contents. We propose to use a SIP-PSTN gateway to access the voice mail service from a PSTN phone.

Use of RTSP enables to record the message once and use the pointer or the URL when forwarding the message without actually forwarding the multimedia file. This is particularly desirable for low bandwidth situations where downloading a whole video mail is very expensive. Moreover, the voicemail can be accessed with any RTSP based media player, e.g., Apple's QuickTime.

Having a SIP interface to the voice mail system makes it readily usable in Internet telephony. The SIP server can be configured to forward the call to voice mail depending on caller address, time of day, etc.

This work was supported by a grant from Sylantro Corp.

A. Outline of the rest of the paper

The remainder of the paper is organized as follows. In Section II, we list the general requirements for an Internet voice mail service. Section III describes and discusses the limitations of existing voice messaging systems. In Section IV, we describe a mechanism to use SIP and RTSP to provide voice (and multimedia) messaging. Section V gives an insight into our implementation. Finally, we describe future work in Section VI.

II. Requirements

In this section we list the requirements for a simple voice mail system.

Recording and playback: The system must be able to record the voice message. It must be able to play back the recorded message to the user to which the message belongs. System should provide security and privacy of messages.

Ease of access: The system should provide a framework in which user can easily browse through his messages. In other words, user should be able to skip a message or part of it, read only the header for a message, pause, delete, rewind or fast-forward a message. It should be possible to have different configurable mail folders.
The system should provide a web interface for accessing the messages.

Telephony interface: The system should be able to support DTMF (Dual Tone Multi Frequency) based navigation and message retrieval. This allows the messages to be accessed using PSTN phones. It should be possible to use the voice mail system in Internet telephony environment as well as in traditional PSTN environment.

Notification: The system must be able to notify the user on arrival of a new message. The notification may be sent using electronic mail, instant messaging, or any other suitable means.

Call reclaiming: The system should support reclaiming the call by the callee, i.e., if the callee picks up the phone while the caller is recording the message, he should be able to talk to the caller.

Unified messaging: The system should handle voice messages as attachments in emails. It should be possible to forward a voice message as an email attachment. This helps in integrating it with the existing Internet mail infrastructure. The system should be readily extensible to support VPIM (Voice Profile for Internet Mail [4], [5]).

Scaleability: The system should be scaleable and should be able to handle thousands of users. Use of a single server may not be appropriate.

Protocol compliance: The system must be compatible with existing Internet multimedia protocols, and should be designed with little or no modification in the current infrastructure.

Other requirements, though not crucial, may also be listed, e.g., support for mailing list, integration with email, message expiry, and so on.

III. Related work

Efforts to design Internet based messaging date back to the early days of the Internet when file transfers were used. Electronic mail changed the way people communicate today in the Internet. Unified messaging further reduces the gap between Internet text based messaging and multimedia based communication.

Profiles have been defined for Internet messaging to support voice. In particular, VPIM [5] supports the interchange of voice messages between voice mail systems, unified messaging systems, email servers and desktop client applications. The basic architecture is to carry the voice attachments in the electronic mail. None of these address the integration of Internet telephony with the voice messaging system. Moreover, carrying the voice bits across the low bandwidth links while forwarding the messages is not desirable. It also requires special purpose client applications which can understand the profiles.

SIP has recently gained momentum, and various schemes have been proposed to forward a call to a voice mail server. The Common Gateway Interface for SIP [6] or the Call Processing Language [7], [8] can be used to configure the SIP server to use an external voice mail service. Campbell and Sparks [9] suggest the use of SIP Request-URI to carry service control information related to voice mail. However, integration of SIP based Internet telephony with the voice mail service is still an active research topic.

IV. Architecture

Fig. 1 shows an example of the voice mail recording. A SIP server handles all the users in a particular domain, e.g., cs.columbia.edu. Different users register their current location with the SIP server, so that the server can contact the user on receipt of an incoming call. The voice mail server also registers its location on behalf of all the users it is serving. From the SIP server's perspective, their are two active locations for every user, one is his actual SIP based phone and the other is the voice mail server.

When another user (hgs@cs.columbia.edu) calls kns10@cs.columbia.edu, the SIP server proxies the call to both the locations. If the user picks up the phone, the branch to the voice mail server is cancelled and a normal SIP call proceeds between hgs and kns10.

The voice mail server is configured to wait for some time, say 10 seconds, before accepting the call. So, if kns10 does not pick up the phone in 10 seconds, the voice mail server is going to accept the call on his behalf. Before accepting the call, the voice mail server sets up the media path with the RTSP server, so that the RTSP server can prompt hgs to leave a message and

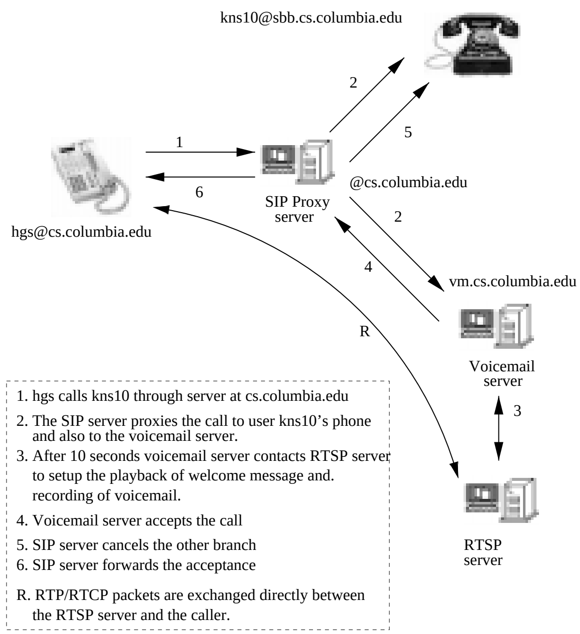

Fig. 1. Forwarding the call to voice mail

the RTSP server can start recording the voice message. Media stream is directly exchanged between the caller (hgs) and the RTSP server.

Once the caller has finished recording, he hangs up. The voice mail server informs the RTSP server to stop recording.

Few things are hidden in the above example. Since the RTP (Real-time Transport Protocol [10]) media packets are directly sent between the caller and the media storage server, the delay is minimum. Also, the voice mail server can notify kns10 about the new message using electronic mail, or some other means.

There are other ways in which the voice mail can be implemented. For instance, the SIP user agent of kns10 can be configured to redirect the call to the voice mail server if there is no response from kns10 within 10 seconds or if his phone is busy. A different request-URI can be used to specify the purpose of calling the voice mail server [9]. For instance, kns10-deposit@vm.cs.columbia.edu can be used to leave a voice message, and kns10-6532-retrieve@vm.cs.columbia.edu to retrieve the voice message number 6532.

Another approach is to configure the SIP server to forward the call to the voice mail server using a CPL [7] script as shown in Fig. 2.

There are trade-offs in using these architectures. Having the voice mail server register with the SIP server on behalf of the user is very simple and does not need knowledge of SIP-CPL by the SIP server or any intelligence in the SIP user agent. The proxy behaviour of the SIP server is sufficient. On the other

```
<?xml version="1.0" ?>
<!DOCTYPE cpl SYSTEM "cpl.dtd">

<cpl>
 <subaction id="voicemail">
  <location url=
    "sip:kns10@vm.cs.columbia.edu">
   <redirect />
  </location>
 </subaction>

 <incoming>
  <address-switch field="origin"
     subfield="host">
   <address
     subdomain-of="cs.columbia.edu">
    <location url=
      "sip:kns10@cbb.cs.columbia.edu">
     <proxy>
      <busy>
        <sub ref="voicemail" />
      </busy>
      <noanswer>
        <sub ref="voicemail" />
      </noanswer>
      <failure>
        <sub ref="voicemail" />
      </failure>
     </proxy>
    </location>
   </address>
   <otherwise>
    <sub ref="voicemail" />
   </otherwise>
  </address-switch>
 </incoming>
</cpl>
```

Fig. 2. CPL script for forwarding a call to the voice mail

hand, using the user agent timeout to redirect the call to the voicemail server requires some intelligence in the user agent, which may not be desirable for embedded low cost SIP enabled devices. Using CPL seems to be the best approach, providing much finer control on the call processing, however it needs CPL enabled servers.

A. *Voice mail server*

The voice mail server has both SIP and RTSP parts. On one side it can receive Internet telephony calls using SIP, and on the other side it behaves as a RTSP client and can perform playback, recording and other control on the voice mail residing at the remote RTSP server.

The RTSP server acts as a storage server for the mails. Separating the voice mail server from the storage server helps in building scaleable systems. For example, a single voice mail server can serve all students of an university, while using the departmental RTSP servers for load balancing. Since the voice mail server does not have to handle the media stream, processing speed is not a bottle-neck.

B. *Retrieving the voice mail*

Retrieving the voice mail is much simpler. Since both RTSP and SIP can be used, user has a wider range of tools. For instance, existing RTSP based media players can be used to directly play the voice messages from the RTSP server. On the other hand, the user can use a SIP user agent to talk to the voice mail server (e.g., using request-URI kns10-6532-retrieve@vm.cs.columbia.edu) and retrieve the message. This approach allows for retrieval of voice messages from a PSTN phone through a SIP-PSTN gateway. The third, and the most preferred approach is to access the voice mail from a web page using a HTTP client, like Netscape.

C. *Telephony interface*

With the existing widespread use of PSTN, it is likely that some of the voice mail systems serve the telephone users, and not the Internet users. This places a new challenge to allow configuration, and retrieval of voice mails from a telephone using DTMF commands.

A SIP-PSTN gateway can be used to provide the telephony access to the voice mail server as shown in Fig. 3. We need

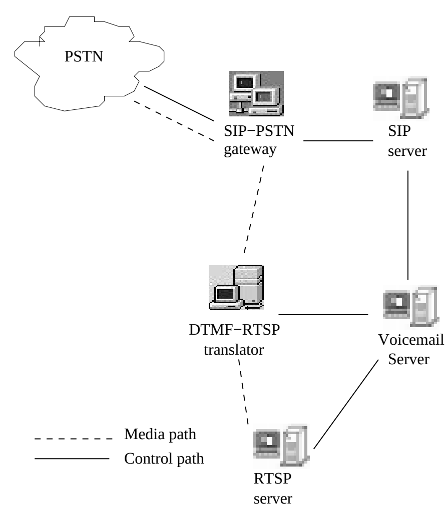

Fig. 3. PSTN access to voice mail

another module, called as DTMF-RTSP translator, which will translate the DTMF commands to the appropriate RTSP messages to be sent to the storage server. Now the media stream from the gateway goes to the RTSP server through the translator. The translator receives the signaling information for the call from the voice mail server.

D. *Call reclaiming*

Another implicit requirement for the voice mail system is to allow reclaiming an already transferred call. If the callee arrives and picks up the phone when the voice mail is being recorded, the system should provide an option for the users to stop the recording and continue talking in a normal call. This is not trivial if the voice mail system is outside the local handset.

The current architecture can use the SIP multiparty conference architecture [11] to support call reclaiming. In the previous example (in Fig. 1), when the call gets transferred to the voice mail server, the voice mail server invites the intended user, kns10@sbb.cs.columbia.edu, in the existing call. If kns10 picks up the phone while the voice mail is being recorded, he joins the existing call to form a three party conference between the caller (hgs), the voice mail server and himself. The voice mail server then drops out of the conference by sending a SIP BYE.

It might be desirable to have the user decide whether to stop the recording or not. The caller may not want to repeat the long message if he has already recorded most of it.

The voice mail server uses the SIP request-URI to identify the purpose of the call. For instance, if the call is directly made to the voice mail server to leave an announcement or a reminder in user's mail box, the server should not try to contact the intended recipient.

E. *Deletion of messages*

The architecture assumes that the RTSP server does the storage of the voice mails. However, there is no explicit mechanism to delete a resource in RTSP, in its current form.

One option is to define a new method, say DELETE, to delete a resource or a media file on the RTSP server.

The other approach is to pretend as if you are recording the file, but terminate the RTSP connection without actually recording anything. To be more specific, an RTSP SETUP with record mode is sent to the server, immediately followed by an RTSP TEARDOWN, without sending a RECORD message. The RTSP server interprets this as a command to delete the file. Even otherwise, the recorded file will be empty, and of no use.

While the first method is more formal and straight forward, it requires modification in RTSP. We have implemented both the methods.

V. Implementation

We have successfully demonstrated an Internet voice mail system based on the above architecture using our SIP and RTSP servers. The system is likely to serve the Computer Sceince Department at Columbia University, replacing the existing PSTN based voice mail and answering machines. In this section we discuss some of the implementation features.

The voice mail server registers with the SIP server on behalf of every user, as discussed earlier. This allows for centralized configuration at the server, when serving different types of user agents (Desktop based client or SIP enabled ethernet phone device). The voice mail server sends email notification to the intended recipient regarding new voice mail arrivals. The template of the notification mail is configurable. This allows the user to modify the email message format as per his taste. For instance, user may just put a clickable RTSP URL in his email notification if he prefers to to use a RTSP client, or the HTTP link to his voice mail inbox, or both.

Fig. 4. Example voice mail inbox

Every user is given a voice mail account and a voice mail web page. The web interface is similar to existing web based mail services, like Hotmail. A sample screen dump of an user's inbox is shown in Fig. 4.

The web interface is through CGI (Common Gateway Interface) and is written in Tcl (Tool Command Language). The Tcl script resides in every user's home area. Users have an option to customize formatting of their voice mail web page. Web hosting of voice mails for users outside the Computer Science department is also supported.

Basic features like folder management, password change, customizing the voice response, deleting messages, sorting the messages based on different parameters (e.g., date, subject, size), and so on are already implemented. Forwarding of the voice message as a MIME (Multipurpose Internet Mail Extensions [12], [13], [14], [15], [16]) attachment to electronic mail is also supported. This helps in integration of electronic mail and voice mail services.

VI. Conclusion and future work

We have described a multimedia mail architecture for Internet telephony, using SIP and RTSP, and shown how it meets the general requirements of a voice mail service. The architecture applies to any kind of multimedia mail, including video, because both SIP and RTSP are designed to support multimedia. Use of existing protocols helps in easy deployment, as related tools (SIP user agents, RTSP media players) are already available.

Various approaches are possible to utilize the voice mail service in the Internet telephony environment. Applicability ranges from a single user subscribed to a voice mail service to a whole university using the campus wide service. Separation of the voice mail server from the signaling and the storage servers helps in building scaleable systems.

We have also described some of the protocol issues, in particular, reclaiming a transferred call using SIP call control and deleting a mail using RTSP methods.

We have developed a prototype voice mail system, and will continue towards further university wide deployment of the system. Other implementation issues related to DTMF based voice mail access and video mail will be looked at in the future versions of our system. We can easily include fax and email messages in the web mailbox, similar to some of the IMAP (Internet Message Access Protocol [17], [18]) web interfaces.

We look at the integration of voice mail, answering machine, electronic mail, MIME attachments for various media subtypes, Internet telephony, and other Internet based services as an active research topic.

VII. Acknowledgments

We would like to thank Tarun Kapoor for his help in implementation and discussion.

References

[1] 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.

[2] 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.

[3] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol (RTSP)," Request for Comments (Proposed Standard) 2326, Internet Engineering Task Force, Apr. 1998.

[4] G. Vaudreuil and G. Parsons, "Voice profile for internet mail - version 2," Request for Comments (Proposed Standard) 2421, Internet Engineering Task Force, Sept. 1998.

[5] G. Vaudreuil and G. Parsons, "Voice profile for internet mail - version 3," Internet Draft, Internet Engineering Task Force, Feb. 1999. Work in progress.

[6] J. Lennox, J. Rosenberg, and H. Schulzrinne, "Common gateway interface for SIP," Internet Draft, Internet Engineering Task Force, May 1999. Work in progress.

[7] J. Lennox and H. Schulzrinne, "CPL: a language for user control of internet telephony services," Internet Draft, Internet Engineering Task Force, Mar. 1999. Work in progress.

[8] J. Rosenberg, J. Lennox, and H. Schulzrinne, "Programming internet telephony services," *IEEE Network*, vol. 13, pp. 42–49, May/June 1999.

[9] B. Campbell and R. Sparks, "Control of service context using SIP Request-URI," Internet Draft, Internet Engineering Task Force, Jan. 2000. Work in progress.

[10] 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.

[11] H. Schulzrinne and J. Rosenberg, "SIP call control services," Internet Draft, Internet Engineering Task Force, June 1999. Work in progress.

[12] N. Freed and N. Borenstein, "Multipurpose internet mail extensions (MIME) part one: Format of internet message bodies," Request for Comments (Draft Standard) 2045, Internet Engineering Task Force, Nov. 1996.

[13] N. Freed and N. Borenstein, "Multipurpose internet mail extensions (MIME) part two: Media types," Request for Comments (Draft Standard) 2046, Internet Engineering Task Force, Nov. 1996.

[14] K. Moore, "MIME (multipurpose internet mail extensions) part three: Message header extensions for Non-ASCII text," Request for Comments (Draft Standard) 2047, Internet Engineering Task Force, Nov. 1996.

[15] N. Freed, J. Klensin, and J. Postel, "Multipurpose internet mail extensions (MIME) part four: Registration procedures," Request for Comments (Best Current Practice) 2048, Internet Engineering Task Force, Nov. 1996.

[16] N. Freed and N. Borenstein, "Multipurpose internet mail extensions (MIME) part five: Conformance criteria and examples," Request for Comments (Draft Standard) 2049, Internet Engineering Task Force, Nov. 1996.

[17] M. Crispin, "Internet message access protocol - version 4," Request for Comments (Proposed Standard) 1730, Internet Engineering Task Force, Dec. 1994.

[18] M. Crispin, "Internet message access protocol - version 4rev1," Request for Comments (Proposed Standard) 2060, Internet Engineering Task Force, Dec. 1996.