Internet Telephony: Architecture and Protocols an IETF Perspective

Henning Schulzrinne
Dept. of Computer Science
Columbia University
hgs@cs.columbia.edu

Jonathan Rosenberg
Bell Laboratories
Lucent Technologies
jdrosen@bell-labs.com

July 2, 1998

Abstract

Internet telephony offers the opportunity to design a global multimedia communications system that may eventually replace the existing telephony infrastructure. We describe the upper-layer protocol components that are specific to Internet telephony services: the Real-Time Transport Protocol (RTP) to carry voice and video data, and the Session Initiation Protocol (SIP) for signaling. We also mention some complementary protocols, including the Real Time Streaming Protocol (RTSP) for control of streaming media, and the Wide Area Service Discovery Protocol (WASRV) for location of telephony gateways.

1 Introduction

Internet telephony, also known as voice-over-IP or IP telephony (IPtel), is the real-time delivery of voice (and possibly other multimedia data types) between two or more parties, across networks using the Internet protocols, and the exchange of information required to control this delivery. Internet telephony offers the opportunity to design a global multimedia communications systems that may eventually replace the existing telephony infrastructure, without being encumbered by the legacy of a century-old technology.

While we will use the common term of "Internet telephony", it should be understood that the addition of other media, such as video or shared applications, does not fundamentally change the problem. Also, unlike in traditional telecommunications networks where distribution applications (radio, TV) and communications applications (telephone, fax) are quite distinct in terms of technology, user interface, communications devices and even regulatory environments, this is not the case for the Internet. The delivery of stored ("streaming") media and telephone-style applications can share almost all of the underlying protocol infrastructure. Indeed, in the model presented here, the same end user application may well serve both. This affords new opportunities to integrate these two modes. Consider, for example, the simple act of rolling a VCR into a conference room to either playback or record a session. The equivalent function requires specialized equipment in the phone system, but is easy to implement in the system to be described here. As another example, it may be desirable to jointly view stored or live broadcast events, in a kind of "virtual couch", where friends, distributed across the Internet, could share the same movie, as well as side comments and conversations.

Internet telephony differs from Internet multimedia streaming primarily in the control and establishment of sessions, the "signaling". While we generally assume that a stored media resource is available at a given location, identified at different levels of abstraction by a URI (Uniform Resource Identifer) or URL (Uniform Resource Locator), participants in phone calls are not so easily located. Personal mobility, call

delegation, availability, and desire to communicate make the process of signaling more complex. On the other hand, streaming media can make use of time-axis manipulations, such as fast-forward or absolute positioning. In the Internet, the Real Time Streaming Protocol (RTSP) [1] is the standard protocol for controlling multimedia streams. Section 5 describes the Session Initiation Protocol (SIP) [2] co-developed by the authors for signaling Internet telephony services.

Both RTSP and SIP are part of a protocol stack that has recently emerged from the Internet Engineering Task Force (IETF) - the *Internet Multimedia Conferencing Architecture* [3] . The protocols encompass both IPtel services and stored media services in an integrated fashion. Figure 1 depicts the stack, along with other protocols likely to be used for both Internet streaming media and Internet telephony. Unlike circuit-switched telephony, Internet telephony services are built on a range of packet switched protocols. For example, the functionality of the SS7 ISUP and TCAP [4] telephony signaling protocols encompass routing, resource reservation, call admission, address translation, call establishment, call management and billing. In an Internet environment, routing is handled by protocols such as the Border Gateway Protocol (BGP) [5], resource reservation by the Resource Reservation Protocol (RSVP) [6] or other resource reservation protocols [7]. SIP, described here, translates application-layer addresses, establishes and manages calls. There is currently no Internet telephony billing protocol in the Internet, although RADIUS [8] or DIAMETER [9], in combination with SIP authentication, may initially serve that purpose for calls through Internet telephony gateways. Having a number of different protocols, each serving a particular function, allows for modularity, flexibility, simplicity, and extensibility. End systems or network servers that only provide a specific service need only implement that particular protocol, without interoperability problems. Furthermore, protocol components can be reused in other applications, avoiding re-invention of specific functions in each application.

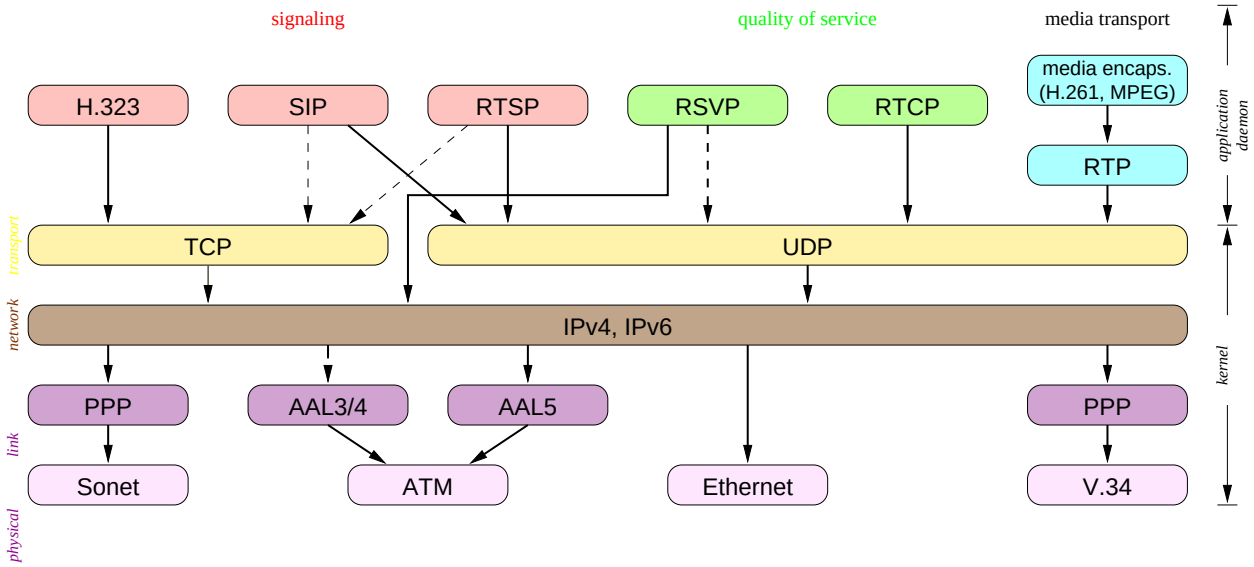

Figure 1: Internet telephony protocol stack

Even though the term Internet telephony is often associated with point-to-point voice service [10], none of the protocols described here are restricted to a single media type or unicast delivery. Indeed, one of the largest advantages of Internet telephony compared to the Plain Old Telephone System (POTS) is the transparency of the network to the media carried, so that adding a new media type requires no changes to the network infrastructure. Also, at least for signaling, the support of multiparty calls differs only marginally from two-party calls.

The protocols mentioned, and the rich services they provide, are just one part of the picture for IP telephony. As it was designed for data transport, the Internet currently offers only best effort service. The

result is that voice packets suffer heavy losses and significant delays when there is network congestion, making the speech quality poor. A number of efforts in the area of Quality of Service (QoS) management are underway in order to address this issue.

The remainder of the paper is organized as follows. Section 2 discusses the differences between the general switched telephone network (GSTN, i.e., the current phone system) and the Internet telephony architecture. Section 3 then discusses how these differences translate into some of the advantages of Internet telephony. We then discuss the key protocol components of Internet telephony. Section 4 discusses the Real Time Transport Protocol (RTP), and addresses some of the efforts underway to resolve the QoS problems with voice transport. Section 5.1 discusses SIP and section 6 discusses some additional protocols, including a call processing language and service location, which form pieces of the overall puzzle. We then put them all together in section 7. We conclude in section 8 and present ideas for future work.

2 Differences between Internet Telephony and the GSTN

Internet telephony differs in a number of respects from the GSTN, both in terms of architecture and protocols. These differences affect the design of telephony services.

Fundmamentally, IP telephony relies on the "end-to-end" paradigm for delivery of services. Signaling protocols are between the end systems involved in the call; network routers treat these signaling packets like any other data, ignoring any semantics implied by them. Note, however, that IP telephony can make use of "signaling routers" (which are effectively proxies) to assist in functions such as user location. In this case, these proxies can be used for routing only of initial signaling messages. Subsequent messages can be exchanged end-to-end. As a consequence of the end-to-end signaling paradigm, call state is as well end to end, as are instantiation of many telephony features.

The Internet itself is both multi-service and service-independent. It provides packet-level transport, end-to-end, for whatever services are deployed at the end systems through higher layer protocols and software. This leads to tremendous flexibility and extensibility. New application level services, such as the web, email, and now IP telephony, can be created and deployed by anyone with access to the Internet.

IP Telephony separates call setup from reserving resources. In the Internet, protocols such as RSVP [6] are used to reserve resources. These protocols are application independent, and reservations may take place before or after actual flow of data begins. When used after the flow of data begins, the data will be treated as best effort. As a result, IP telephony can be used without per-call resource reservation in networks with sufficient capacity. On the other hand, removing the "atomicity" of call setup found in the current telephone system also breaks the assumption of "all or nothing" call completion: since call setup and resource reservation are distinct, one may succeed, while the other may fail. If resources are reserved first, the caller may incur a cost for holding those resources while "the phone rings", even though the call is not answered. (If the network does not charge for reservations that are not actually used, the network becomes vulnerable to denial-of-service attacks, where the attacker can block others from making reservations.) Also, in the phone system, the resource needs for a single leg of a call are known ahead of time; this is clearly not the case for Internet telephony calls, where the callee may choose to communicate with only a subset of the media offered by the caller.

Due to the limited signaling abilities of GSTN end systems, GSTN addresses (phone numbers) are overloaded with at least four functions: end point identification, service indication, indication of who pays for the call and carrier selection. The GSTN also ties call origination with payment, except as modified by the address (800 numbers) or specific manual features (such as collect calls). IPtel addresses, which are formulated as URL's, are used solely for endpoint identification and basic service indication. The other functions, such as payment and carrier selection, are more readily handled by the protocols, such as RSVP and RTSP, which carry the addresses.

Internet telephony offers a larger degree of freedom to allocate functions between network servers and user-supplied and operated end systems. For example, because of the end-to-end signaling, phone services such as distinctive ringing based on call urgency, media-based endpoint selection, and cryptographically authenticated caller id (in addition to features available from POTS phones, such as speed dialing and caller ID) may be accomplished trivially by even standalone IP telephones, resulting in better scaling.

The phone system employs different signaling protocols between a user and the "network" (User-Network Interface – UNI) as compared to between (implicitly trusted) network elements (Network-Network Interface – NNI). This makes certain features (such as mumber translation) unavailable to end users, or leads to classifying end users as "networks", with the attendant security implications in the access to databases and network resources. The UNI/NNI distinction does not exist on the Internet, both at the level of data transport and signaling. SIP can set up calls between two end systems just as easily as creating a "channel" on an aggregated RTP session (see Section 4.3).

The open, multi-service, end-to-end nature of the Internet also means that various components of telephony services can be provided by completely different service vendors (of course, agreement on protocols is necessary for interoperability). For example, one vendor can provide a name to IP address mapping service, another can provide voice-mail, another can provide mobility services, while yet another can provide conferencing services. Furthermore, the end-to-end nature of the Internet means that anyone with an Internet connection can run and operate such a service. This leads to an easy-entry, highly competitive marketplace for all Internet services, such as IP telephony.

This separation of functionality also simplifies the number portability problem. As an organization may provide just a name mapping service, a user can change other service providers (such as their voicemail provider or ISP) without a change in name.[1] Changing name providers may require a change in name. However, automated white-pages services, such as those run by Four11, allow another layer of indirection which further alleviates the number portability problem.

3 Features of Internet Telephony

The architectural differences described in the previous section lead to a number of advantages from both a user perspective and a carrier perspective [12], beyond just "cheaper phone calls":

Adjustable quality: While IPtel currently has the reputation for tin-can quality (due to low bitrate codecs, in part) there is no reason (except lack of bandwidth) why the same technology cannot supply audiophile-quality music. Because the Internet is not a service-specific network, the media exchanged is chosen entirely by end systems. As such, end systems can control the amount of compression based on network bandwidth [13] or the content to be transmitted. For example, music-on-hold (as it is not speech) is not suitable for very low bit-rate speech codecs.

Security: The Internet has the reputation as being insecure, even though it is probably easier to tap a telephone demarc box than a router. SIP can encrypt and authenticate signaling messages; RTP supports encryption of media. Together, these provide cryptographically secure communications.

User identification: Standard POTS and ISDN telephone service offers caller id indicating the number (or, occasionally, name) of the caller, but during a bridged multi-party conference, there is no indication of who is talking. The real-time transport protocol (RTP) [14] used for Internet telephony easily supports talker indication in both multicast and bridged configurations and can convey more detailed information if the caller desires.

[1] There is a more difficult issue of maintaining the same IP network address when changing providers [11].

User interface: Most POTS and ISDN telephones have a rather limited user interface, with at best a two-line liquid crystal display or, in the public network, cryptic commands like "*69" for call-back. Advanced GSTN features such as call-forwarding are rarely used or customized, since the sequence of steps is typically not intuitive. This is due in part to the limited signaling capabilities of end systems, and the general notion of "network intelligence" as compared to "end system intelligence". As IP telephony end systems have much richer signaling capabilities, the graphical user interface offered by Internet telephony can be more readily customized and offer richer indications of features, process and progress.

Computer-telephony integration: Because of the complete separation of data and control paths and the separation of phone equipment from the PC's controlling them, computer-telephony integration (CTI) [15] is very complex, with specs [16] running to 3,300 pages. Much of the call handling functionality can be easily accomplished once the data and control path pass through intelligent, network-connected end systems.

Feature Ubiquity: The current phone system offers very different features depending on whether the parties are connected by the same Private Branch Exchange (PBX), reside within the same local calling area or are connected by a long-distance carrier. Even trivial features such as caller ID only work for a small fraction of international calls, for example. A PBX may not allow a call to be forwarded outside that PBX, or cause the forwarded call to still go through the PBX. Internet telephony does not suffer from this problem. This is because the Internet protocols are internationally used, and because services are defined largely by the end systems.

Multimedia: As pointed out, adding additional media such as video, shared whiteboards or shared applications is much easier in the Internet environment compared to the POTS and ISDN, as multiplexing is natural for packet networks. This makes signaling protocols simpler as well, as issues such as B-channel allocation and synchronization are non-existent in the Internet.

Carriers benefit as well:

Silence suppression and compression: Sending audio as packets makes it easy to suppress silence periods, further reducing bandwidth consumption, particularly in a multi-party conference or for voice announcement systems. Unlike the GSTN, which generally does such silence suppression across trans-continental links, IP telephony performs silence suppression at the endpoints. Furthermore, as packet networks are much more suitable to multiplexing, no network support is required to take advantage of end system silence suppression. This leads to a reduction in cost to perform the silence suppression (as it's distributed to end systems where it can be done cheaply), and an improvement in the scope of its benefits. Furthermore, compression can be used at end systems to reduce bandwidth consumption across the entire network. Unfortunately, compression is at odds with enhanced voice quality services described above. However, there exist codecs which can compress wideband speech to 16 kb/s, which can give both excellent voice quality and reduced bandwidth compared to the GSTN. Note that silence suppression and compression compensate for the decreased efficiency of packet switching.

Shared facilities: The largest operational savings can probably come from provisioning and managing a single, integrated network, rather than separate voice, data and signaling networks.

Advanced services: From first experiences and protocols, it appears to be far simpler to develop and deploy advanced telephony services in a packet-switched environment than in the GSTN [17, 18]. Internet protocols, such as SIP [2] that support standard CLASS (Custom Local Area Signaling Services) [19]

features (such as Call Forward No Answer) take only a few tens of pages to specify. They can perform the functions of both the user-to-network signaling protocols such as Q.931 as well as the network signaling (ISUP, Signaling System 7).

Separation of voice and control flow: In telephony, the signaling flow, even though carried on a separate network, has to "touch" all the intermediate switches to set up the circuit. Since packet forwarding in the Internet requires no setup, Internet call control can concentrate on the call (rather than connection) functionality. For example, it is easy to avoid triangle-routing when forwarding or transferring calls; the transferring party can simply inform the transferred party of the address of the transferred-to party, and the two can contact each other directly. As there is no network connection state, none needs to be torn down.

4 RTP for Data Transport

Real-time flows such as IPtel voice and video streams have a number of common requirements that distinguish them from "traditional" Internet data services:

Sequencing: The packets must be re-ordered in real time at the receiver, should they arrive out of order. If a packet is lost, it must be detected and compensated for without retransmissions.

Intra-media synchronization: The amount of time between when successive packets are to be "played-out" must be conveyed. For example, no data is usually sent during silence periods in speech. The duration of this silence must be reconstructed properly.

Inter-media synchronization: If a number of different media are being used in a session, there must be a means to synchronize them, so that the audio that is played out matches the video. This is also known as lip-sync.

Payload identification: In the Internet, it is often necessary to change the encoding for the media ("payload") on the fly to adjust to changing bandwidth availability or the capabilities of new users joining a group. Some kind of mechanism is therefore needed to identify the encoding for each packet.

Frame indication: Video and audio are sent in logical units called frames. It is necessary to indicate to a receiver where the beginning or end of a frame is, in order to aid in synchronized delivery to higher layers.

These services are provided by a *transport protocol*. In the Internet, the Real Time Transport Protocol [14] is used for this purpose. RTP has two components. The first is RTP itself, and the second is RTCP, the Real Time Control Protocol. Transport protocols for real time media are not new, dating back to the 1970's [20]. However, RTP provides some functionality beyond resequencing and loss detection:

Multicast-friendly: RTP and RTCP have been engineered for multicast. In fact, they are designed to operate in both small multicast groups, like those used in a three-person phone call, to huge ones, like those used for broadcast events.

Media independent: RTP provides services needed for generic real time media, such as voice and video. Any codec-specific additional header fields and semantics are defined for each media codec in separate specifications.

Mixers and translators: Mixers are devices which take media from several users, mix or "bridge" them into one media stream, and send the resulting stream out. Translators take a single media stream, convert it to another format, and send it out. Translators are useful for reducing the bandwidth requirements of a stream before sending it over a low-bandwidth link, without requiring the RTP source to reduce its bitrate. This allows receivers that are connected via fast links to still receive high quality media. Mixers limit the bandwidth if several sources send data simultaneously, fulfilling the function of conference bridge. RTP includes explicit support for mixers and translators.

QoS feedback: RTCP allows receivers to provide feedback to all members of a group on the quality of the reception. RTP sources may use this information to adjust their data rate, while other receivers can determine whether Quality of Service (QoS) problems are local or network-wide. External observers can use it for scalable quality-of-service management.

Loose Session Control: With RTCP, participants can exchange identification information such as name, email, phone number, and brief messages.

Encryption: RTP media streams can be encrypted using keys that are exchanged by some non-RTP method, e.g., SIP or the Session Description Protocol (SDP) [21].

4.1 RTP

RTP is generally used in conjunction with the User Datagram Protocol (UDP), but can make use of any packet-based lower-layer protocol. When a host wishes to send a media packet, it takes the media, formats it for packetization, adds any media-specific packet headers, prepends the RTP header, and places it in a lower-layer payload. It is then sent into the network, either to a multicast group or unicast to another participant.

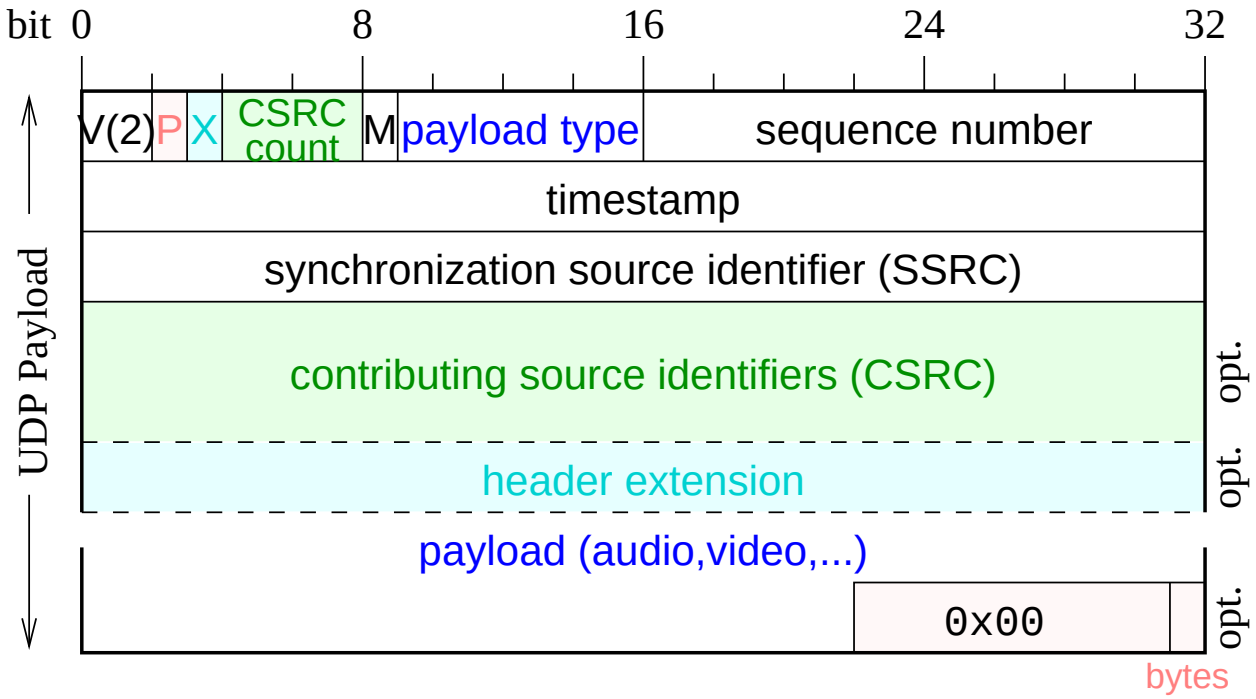

Figure 2: RTP fixed header format

The RTP header (Fig. 2) is 12 bytes long. The V field indicates the protocol version. The X flag signals the presence of a header extension between the fixed header and the payload. If the P bit is set, the payload is padded to ensure proper alignment for encryption.

Users within a multicast group are distinguished by a random 32-bit synchronization source SSRC identifier. Having an application-layer identifier allows to easily distinguish streams coming from the same translator or mixer and associate receiver reports with sources. In the rare event that two users happen to choose the same identifier, they redraw their SSRCs.

As described above, a mixer combines media streams from several sources, e.g., a conference bridge might mix the audio of all active participants. In current telephony, the participants may have a hard time distinguishing who happens to be speaking at any given time. The Contributing SSRC (CSRC) list, whose length is indicated by the CSRC length field, lists all the SSRC that "contributed" content to the packet. For the audio conference, it would list all active speakers.

RTP supports the notion of media-dependent framing to assist in the reconstruction and playout process. The marker M bit provides information for this purpose. For audio, the first packet in a voice talkspurt can be scheduled for playout independently of those in the previous talkspurt. The bit is used in this case to indicate the first packet in a talkspurt. For video, a video frame can only be rendered when its last packet has arrived. Here, the marker bit is used to indicate the last packet in a video frame.

The payload type identifies the media encoding used in the packet. The sequence number increments sequentially from one packet to the next, and is used to detect losses and restore packet order. The timestamp, incremented with the media sampling frequency, indicates when the media frame was generated.

The payload itself may contain headers specific for the media; this is described in more detail below.

4.2 RTCP: Control and Management

The Real Time Control Protocol, RTCP is the companion control protocol for RTP. Media senders (sources) and receivers (sinks) periodically send RTCP packets to the same multicast group (but different ports) as is used to distribute RTP packets. Each RTCP packet contains a number of elements, usually a sender report (SR) or receiver report followed by source descriptions (SDES). Each serves a different function:

Sender Reports (SR) are generated by users who are also sending media (RTP sources). They describe the amount of data sent so far, as well as correlating the RTP sampling timestamp and absolute ("wall clock") time to allow synchronization between different media.

Receiver Reports (RR) are sent by RTP session participants which are receiving media (RTP sinks). Each such report contains one block for each RTP source in the group. Each block describes the instantaneous and cumulative loss rate and jitter from that source. The block also indicates the last timestamp and delay since receiving a sender report, allowing sources to estimate their distance to sinks.

Source Descriptor (SDES) packets are used for session control. They contain the CNAME (Canonical Name), a globally unique identifier similar in format to an email address. The CNAME is used for resolving conflicts in the SSRC value and associate different media streams generated by the same user. SDES packets also identify the participant through its name, email, and phone number. This provides a simple form of session control. Client applications can display the name and email information in the user interface. This allows session participants to learn about the other participants in the session. It also allows them to obtain contact information (such as email and phone) to allow for other forms of communication (such as initiation of a separate conference using SIP). This also makes it easier to contact a user should he, for example, have left his camera running.

If a user is leaving, he includes a *BYE* message. Finally, *Application (APP)* elements can be used to add application-specific information to RTCP packets.

Since the sender reports, receiver reports, and SDES packets contain information which can continually change, it is necessary to send these packets periodically. If the RTP session participants simply sent RTCP

packets with a fixed period, the resulting bandwidth used in the multicast group would grow linearly with the group size. This is clearly undesirable. Instead, each session member counts the number of other session members it hears from (via RTCP packets). The period between RTCP packets from each user is then set to scale linearly with the number of group members. This ensures that the bandwidth used for RTCP reports remains fixed, independent of the group size. Since the group size estimate is obtained by counting the number of other participants, it takes time for each new participant to converge to the correct group size count. If many users simultaneously join a group, as is common in broadcast applications, each user will have an incorrect, and very low estimate of the group size. The result is a flood of RTCP reports. A back-off algorithm called reconsideration [22] is used to prevent such floods.

4.3 Payload Formats

The above mechanisms in RTP provide for services needed for the generic transport of audio and video. However, particular codecs will have additional requirements for information that needs to be conveyed. To support this, RTP allows for payload formats to be defined for each particular codec. These payload formats describe the syntax and semantics of the RTP payload. The particular semantic of the payload is communicated in the RTP payload type indicator bits. These bits are mapped to actual codecs and formats via a binding to names, registered with the Internet Assigned Numbers Authority (IANA), and conveyed out of band (through SDP, for example). Any number of bindings can be conveyed out of band; this allows an RTP source to change encoding formats mid-stream without explicit signaling. Furthermore, anyone can register a name (so long as it has not been used), and procedures are defined for doing so. This allows for RTP to be used with any kind of codec developed by anyone.

RTP media payload formats have been defined for the H.263 [23], H.261 [24], JPEG [25] and MPEG [26] video codecs, and a host of other audio and video encoders are supported with simpler payload formats [27].

Furthermore, RTP payload formats are being defined to provide some generic services. One format, for redundant audio codings [28], allows a user to transmit audio content using multiple audio codecs, each delayed from the previous, and of a lower bitrate. This allows for lost packets to be recovered from subsequent packets, albeit with a lower quality codec. Another payload format is being defined for parity and Reed Solomon like forward error correction (FEC) mechanisms, to allow for recovery of lost packets in a codec-independent manner [29]. Yet another format is being introduced to multiplex media from multiple users into a single packet [30]. This is particularly useful for trunk replacement between Internet telephony gateways, where it can provide a significant reduction in packet header overheads.

4.4 Resource Reservation

Given the importance of telephony services, we anticipate that a significant fraction of the Internet bandwidth will be consumed by voice and video, that is, RTP-based protocols. Due to its tight delay constraints, IPtel streams are also likely candidates for guaranteed QOS. Unfortunately, existing proposals such as RSVP [6] are rather complex, largely due to features such as receiver orientation and support for receiver diversity that is likely to be of limited use for Internet telephony. We have proposed [7] to dispense with a separate resource reservation protocol altogether and simply use RTCP messages to tell routers along the path to reserve sufficient resources. To determine the amount of bandwidth for the reservation, RTCP sender reports (SR) can be used as is (by observing the difference between two subsequent SR byte counts), or an additional field can be inserted that specifies the desired grade of service in more detail. RTCP messages carrying reservation requests are marked for special handling by a router alert option [31]. The receiver reports back whether the reservation was completely or only partially successful.

Similar proposals of simplified, sender-based resource reservation protocols can be found (albeit not using RTCP) in the Scalable Reservation Protocol (SRP) [32].

The issue of reservations for IPtel services has another facet: if a reservation fails, the call can still take place, but using best effort transport of the audio. This is certainly preferable to a fast-busy signal, where no communications at all are established. This introduces the possibility of completely removing the admission decision all together. On each link in the network, sufficient bandwidth for voice traffic can be allocated. Time honored approaches to telephone network engineering can be used to determine the amount of allocation needed. The remaining bandwidth on links will be used for other services, and for best effort. Incoming packets can be classified as voice or non-voice. Should there be too many voice connections active at any point in time (possible since there is no admission control), the extra calls can simply become best effort. Good network engineering can place reasonably low probabilities on such an event, and the use of "spillover" best effort bandwidth can eliminate admission control to handle these unlikely occurrences.

The above mechanism is generally referred to as part of the class of IP service referred to as *differentiated services*. These rely on user subscription (to facilitate network engineering) and packet classification and marking at network peripheries. From the above discussion, IPtel is a prime candidate for such a service. In fact, the *premium service* [33] provides almost no delay and jitter for its packets, making it ideal for real time voice and video.

5 Signaling: Session Initiation Protocol

A defining property of IPtel is the ability of one party to signal to one or more other parties to initiate a new call. For purposes of discussing the Session Initiation Protocol, we define a call is an association between a number of participants. The signaling association between a pair of participants in a call is referred to as a connection. Note that there is no physical channel or network resources associated with a connection; the connection exists only as signaling state at the two endpoints. A session refers to a single RTP session carrying a single media type. The relationship between the signaling connections and media sessions can be varied. In a multiparty call, for example, each participant may have a connection to every other participant, while the media is being distributed via multicast, in which case there is a single media session. In other cases, there may be a single unicast media session associated with each connection. Other, more complex, scenarios are possible as well.

An IP telephony signaling protocol must accomplish a number of functions:

Name translation and user location involve the mapping between names of different levels of abstraction, e.g., a common name at a domain and a user name at a particular Internet host. This translation is necessary in order to determine the IP address of the host to exchange media with. Usually, a user has only the name or email address of the person they would like to communicate with (j.doe@company.com, for example). This must be translated into an IP address. This translation is more than just a simple table lookup. The translation can vary based on time of day (so that a caller reaches you at work during the day, and at home in the evening), caller (so that your boss always gets your work number), or the status of the callee (so that calls to you are sent to your voice mail when you are already talking to someone), among other criteria.

Feature negotiation allows a group of end systems to agree on what media to exchange and their respective parameters, such as encodings. The set and type of media need not be uniform within a call, as different point-to-point connections may involve different media and media parameters. Many software codecs are able to receive different encodings within a single call and in parallel, for example, while being restricted to sending one type of media for each stream.

Any call participant can invite others into an existing call and terminate connections with some (*call participant management*). During the call, participants should be able to transfer and hold other users.

The most general model of a multi-party association is that of a full or partial mesh of connections among participants (where one of the participants may be a media bridge), with the possible addition of multicast media distribution.

Feature changes make it possible to adjust the composition of media sessions during the course of a call, either because the participants require additional or reduced functionality or because of constraints imposed or removed by the addition or removal of call participants.

There are two protocols that have been developed for such signaling operations, the Session Initiation Protocol (SIP) [2], developed in the IETF, and H.323 [34], developed by the ITU. In the next section, we will discuss SIP, while H.323 is covered by Toga et al. within this issue.

5.1 SIP Overview

SIP is a client-server protocol. This means that requests are generated by one entity (the client), and sent to a receiving entity (the server) which processes them. As a call participant may either generate or receive requests, SIP-enabled end systems include a protocol client and server (generally called a user agent server). The user agent server generally responds to the requests based on human interaction or some other kind of input. Furthermore, SIP requests can traverse many *proxy servers*, each of which receives a request and forwards it towards a next hop server, which may be another proxy server or the final user agent server. A server may also act as a *redirect server*, informing the client of the address of the next hop server, so that the client can contact it directly. There is no protocol distinction between a proxy server, redirect server, and user agent server; a client or proxy server has no way of knowing which it is communicating with. The distinction lies only in function: a proxy or redirect server cannot accept or reject a request, whereas a user agent server can. This is similar to the Hypertext Transfer Protocol (HTTP) model of clients, origin and proxy servers. A single host may well act as client and server for the same request. A connection is constructed by issuing an INVITE request, and destroyed by issuing a BYE request.

As in HTTP, the client requests invoke *methods* on the server. Requests and responses are textual, and contain header fields which convey call properties and service information. SIP reuses many of the header fields used in HTTP, such as the entity headers (e.g., Content-type) and authentication headers. This allows for code reuse, and simplifies integration of SIP servers with web servers.

Calls in SIP are uniquely identified by a call identifier, carried in the Call-ID header field in SIP messages. The call identifier is created by the creator of the call and used by all call participants. Connections have the following properties: The *logical connection source* indicates the entity that is requesting the connection (the originator). This may not be the entity that is actually sending the request, as proxies may send requests on behalf of other users. In SIP messages, this property is conveyed in the From header field. The *logical connection destination* contained in the To field names the party who the originator wishes to contact (the recipient). The *media destination* conveys the location (IP address and port) where the media (audio, video, data) are to be sent for a particular recipient. This address may not be the same address as the logical connection destination. *Media capabilities* convey the media that a participant is capable of receiving and their attributes. Media capabilities and media destinations are conveyed jointly as part of the payload of a SIP message. Currently, the Session Description Protocol (SDP) [21] serves this purpose, although others are likely to find use in the future [2]. SDP expresses lists of capabilities for audio and video and indicates where the media is to be sent to. It also allows to schedule media sessions into the future and schedule repeated sessions.

SIP defines several methods, where the first three manage or prepare calls and connections: INVITE invites a user to a call and establishes a new connection, BYE terminates a connection between two users in a call (note that a call, as a logical entity, is created when the first connection in the call is created, and

[2] In fact, H.245 capability sets can be carried in SDP for this purpose

destroyed when the last connection in the call is destroyed), and OPTIONS solicits information about capabilities, but does not set up a connection. STATUS informs another server about the progress of signaling actions that it has requested via the Also header (see below). ACK is used for reliable message exchanges for invitations. CANCEL terminates a search for a user. Finally, REGISTER conveys information about a user's location to a SIP server.

SIP makes minimal assumptions about the underlying transport protocol. It can directly use any datagram or stream protocol, with the only restriction that a whole SIP request or response has to be either delivered in full or not at all. SIP can thus be used with UDP or TCP in the Internet, and with X.25, AAL5/ATM, CLNP, TP4, IPX or PPP elsewhere. Network addresses within SIP are also not restricted to being Internet addresses, but could be E.164 (GSTN) addresses, OSI addresses or private numbering plans. In many cases, addresses can be any URL; for example, a call can be "forwarded" to a mailto URL for email delivery.

5.2 Message Encoding

Unlike other signaling protocols such as Q.931 and H.323, SIP is a text-based protocol. This design was chosen to minimize the cost of entry. The data structures needed in SIP headers all fall into the parameter-value category, possible with a single level of subparameters, so that generic data coding mechanisms like Abstract Syntax Notation 1 (ASN.1) offer no functional advantage. Many parameters are textual

Unlike the ASN.1 Packed Encoding Rules (PER) and Basic Encoding Rules (BER), a SIP header is largely self-describing. Even if an extension has not been formally documented, as was the case for many common email headers, it is usually easy to reverse-engineer them. Since most values are textual, the space penalty is limited to the parameter names, usually at most a few tens of bytes per request. (Indeed, the ASN.1 PER-encoded H.323 signaling messages are larger than equivalent SIP messages.) Besides, extreme space efficiency is not a concern for signaling protocols.

If not designed carefully, text-based protocols can be difficult to parse due to their irregular structure. SIP tries to avoid this by maintaining a common structure of all header fields, allowing a generic parser to be written.

Unlike, say, HTTP and Internet email, SIP was designed for character-set independence, so that any field can contain any ISO 10646 character. Since SIP operates on an 8-bit clean channel, binary data such as certificates does not have to be encoded. Together with the ability to indicate languages of enclosed content and language preferences of the requestor, SIP is well suited for cross-national use.

5.3 Addressing and Naming

To be invited and identified, the called party has to be named. Since it is the most common form of user addressing in the Internet, SIP chose an email-like identifier of the form *"user@domain"*, *"user@host"*, *"user@IP_address"* or *"phone-number@gateway"*. The identifier can refer to the name of the host that a user is logged in at the time, an email address or the name of a domain-specific name translation service. Addresses of the form *"phone-number@gateway"* designate GSTN phone numbers reachable via the named gateway.

SIP uses these addresses as part of SIP URLs, such as `sip:j.doe@example.com`. This URL may well be placed in a web page, so that clicking on the link initiates a call to that address, similar to a `mailto` [35] URL today.

We anticipate that most users will be able to use their email address as their published SIP address. Email addresses already offer a basic location-independent form of addressing, in that the host part does not have to designate a particular Internet host, but can be a domain, which is then resolved into one or more possible domain mail server hosts via Domain Name System (DNS) MX (mail exchange) records.

This not only saves space on business cards, but also allows re-use of existing directory services such as the Lightweight Directory Access Protocol (LDAP) [36], DNS MX records (as explained below) and email as a last-ditch means of delivering SIP invitations.

For email, finding the mail exchange host is often sufficient to deliver mail, as the user either logs in to the mail exchange host or uses protocols such as the Internet Mail Access Protocol (IMAP) or the Post Office Protocol (POP) to retrieve their mail. For interactive audio and video communications, however, participants are typically sending and receiving data on the workstation, PC or Internet appliance in their immediate physical proximity. Thus, SIP has to be able to resolve *"name@domain"* to *"user@host"*. A user at a specific host will be derived through zero or more translations. A single externally visible address may well lead to a different host depending on time of day, media to be used, and any number of other factors. Also, hosts that connect via dial-up modems may acquire a different IP address each time.

5.4 Basic Operation

The most important SIP operation is that of inviting new participants to a call. A SIP client first obtains an address where the new participant is to be contacted, of the form *name@domain*. The client then tries to translate this domain to an IP address where a server may be found. This translation is done by trying, in sequence, DNS Service (SRV) records, MX, Canonical Name (CNAME) and finally Address (A) records. Once the server's IP address has been found, the client sends it an INVITE message using either UDP or TCP.

The server which receives the message is not likely to be the user agent server where the user is actually located; it may be a proxy or redirect server. For example, a server at `example.com` contacted when trying to call `joe@example.com` may forward the INVITE request to `doe@sales.example.com`. A Via header traces the progress of the invitation from server to server, allows responses to find their way back and helps servers to detect loops. A redirect server, on the other hand, would respond to the INVITE request, telling the caller to contact `doe@sales.example.com` directly. In either case, the proxy or redirect server must somehow determine the next hop server. This is the function of a *location server*. A location server is a non-SIP entity which has information about next hop servers for various users. The location server can be anything – an LDAP server, a proprietary corporate database, a local file, the result of a finger command, etc. The choice is a matter of local configuration. Figures 3 and 4 show the behavior of SIP proxy and redirect servers, respectively.

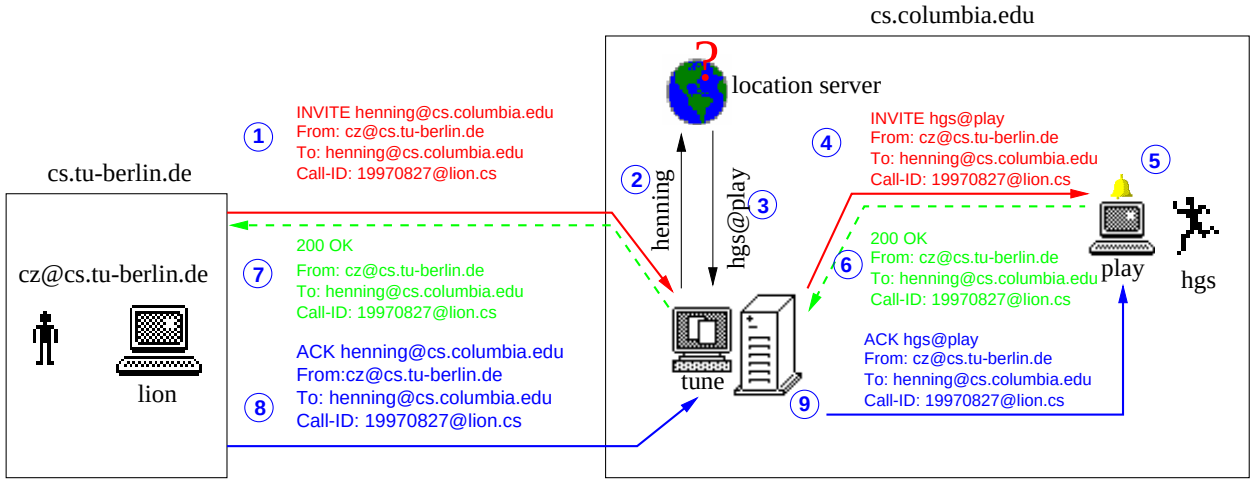

Figure 3: SIP Proxy Server Operation

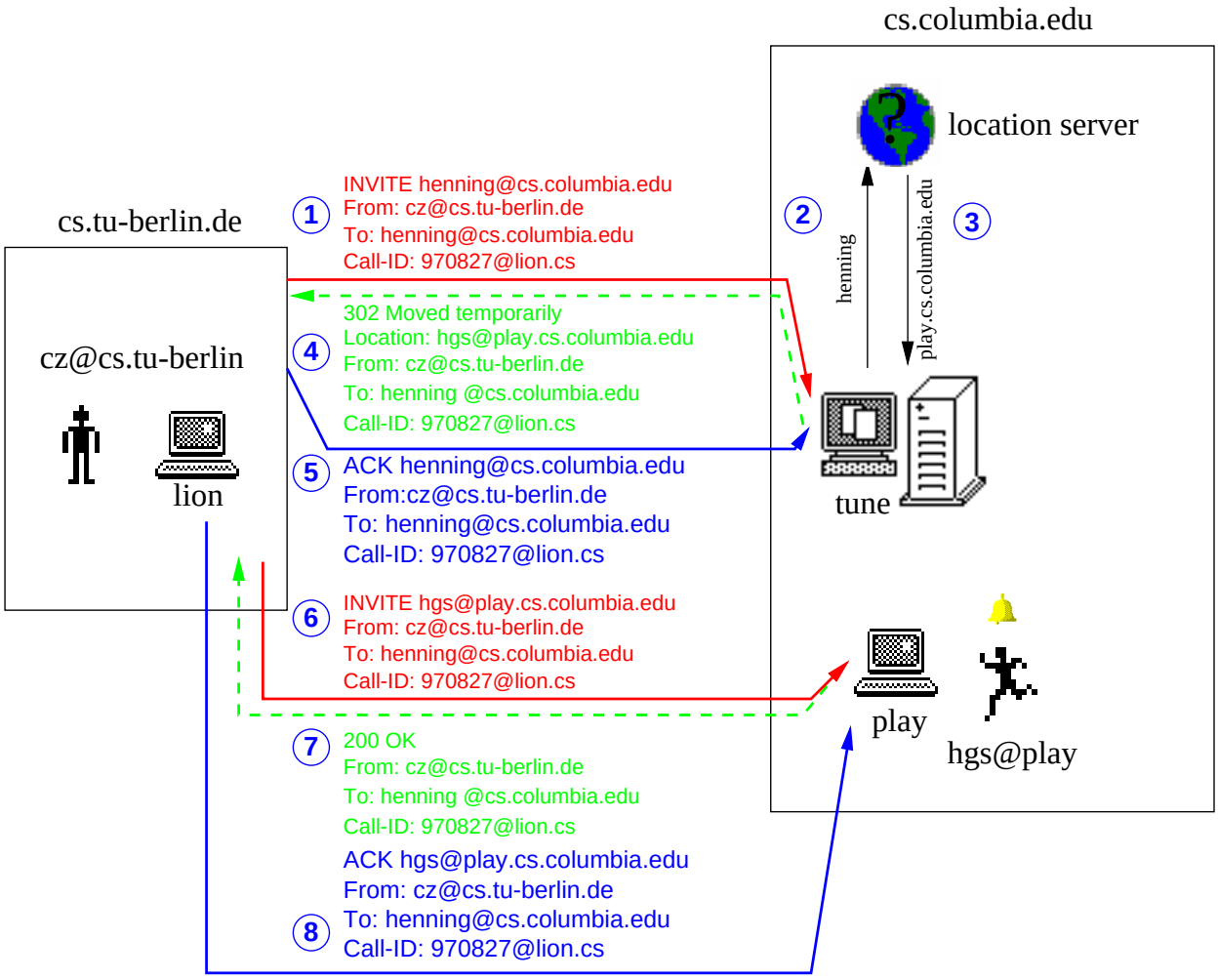

Figure 4: SIP Redirect Server Operation

Proxy servers can forward the invitation to multiple servers at once, in the hopes of contacting the user at one of the locations. They can also forward the invitation to multicast groups, effectively contacting multiple next hops in the most efficient manner.

Once the user agent server has been contacted, it sends a response back to the client. The response has a response code and response message. The codes fall into classes 100 through 600, similar to HTTP.

Unlike other requests, invitations cannot be answered immediately, as locating the callee and waiting for a human to answer may take several seconds. Call requests may also be queued, e.g., if the callee is busy. Responses of the 100 class (denoted as 1xx) indicate call progress; they are always followed by other responses indicating the final outcome of the request.

While the 1xx responses are provisional, the other classes indicate the final status of the request: 2xx for success, 3xx for redirection, 4xx, 5xx and 6xx for client, server and global failures, respectively. 3xx responses list, in a Location header, alternate places where the user might be contacted. To ensure reliability even with unreliable transport protocols, the server retransmits final responses until the client confirms receipt by sending an ACK request to the server.

All responses can include more detailed information. For example, a call to the central “switchboard” address may return a web page that includes links to the various departments in the company, providing navigation more appropriate to the Internet than an interactive voice response system (IVR).

5.5 Protocol Extensions

Since IPtel is still immature, it is likely that additional signaling capabilities will be needed in the future. Also, individual implementations and vendors may want to add additional features. SIP is designed so that the client can either inquire about server abilities first or proceed under the assumption that the server supports the extension and then "back off" if the assumption was wrong.

Methods: As in HTTP, additional methods can be introduced. The server signals an error if a method requested by a client is not supported and informs it with the Public and Allow response headers about the methods that it does support. The OPTIONS request also returns the list of available methods.

Request and response headers: As in HTTP or the Simple Mail Transport Protocol (SMTP), client and server can add request and response headers which are not crucial to interpreting the request or response without explicit indication. The entity receiving the header simply silently ignores headers that it does not understand. However, this mechanism is not sufficient, as it does not allow the client to include headers that are vital to interpreting the request. Rather than enumerating "need-to-know" non-standard headers, the SIP Required header indicates those features that the client needs from the server. The server must refuse the request if it does not understand one of the features enumerated. Feature names are either registered with the Internet Assigned Number Authority (IANA) or derived hierarchically from the feature owner's Internet domain name, giving hints as to where further information might be found. SIP uses this to ascertain whether telephony call-control functions are supported, avoiding the problem of partial implementations that have unpredictable sets of optional features.

Status codes: Status codes returned in responses are classified by their most-significant digit, so that the client knows whether the request was successful, failed temporarily or permanently. A textual status message offers a fall-back mechanism that allows the server to provide further human-readable information.

5.6 Telephony Services

SIP takes a different approach than standard telephony in defining services. Rather than explicitly describing the implementation of a particular service, it provides a number of elements, namely headers and methods, to construct services. The two principle headers used are Also and Replaces. When present in a request or response, the Also header instructs the recipient to place a call to the parties listed. Similarly, Replaces instructs the recipient to terminate any connections with the parties listed. An additional method, called STATUS, is provided to allow a client to obtain results about progress of calls requested via the Also header.

These elements, along with the basic SIP components, are easily used to construct a variety of traditional telephony services. *700, 800 and 900* services (permanent numbers, freephone and paid information services) are, from a call control perspective,simply special cases of call forwarding, governed by a database lookup or some server-specified algorithm. The charging mechanisms, which differ for the three services, are handled by other protocols in an orthogonal fashion. Unlike for Intelligent Networking (IN), the number of such lookups within a call is not limited. Call forwarding services based on user status or preferences similarly require no additional protocol machinery. As a simple example, we have implemented an automatic call forwarding mechanism that inspects a callee's appointment calendar to forward calls or indicate a more opportune time to call back.

While *call forwarding* precedes a call, *call transfer* allows to direct a call participant to connect to a different subscriber. Transfer services include blind and supervised call transfer, attendant and operator

services, auto-dialer for telemarketing, or interactive voice response. All are supported through use of the Also header combined with programmed behavior specific to the particular service. For example, blind transfer is implemented by having the transferring party send a BYE to the transferred party containing an Also header listing the transferred-to party.

In SIP, call setup and session parameter modification are accomplished by the same (INVITE) request, as all SIP requests are idempotent.

In an Internet environment, no "lines" are tied up by active calls when media is not being sent. This means an IP telephone can support an unlimited number of active calls at one time. This makes it easy, for example, to implement call waiting and camp-on.

5.7 Multi-Party Calls

SIP supports the three basic modes of creating multiparty conferences (and their combinations): via network-level multicast, via one or more bridges (also known as multipoint control units) or as a mesh of unicast connections. Multicast conferences require no further protocol support beyond listing a group address in the session description; indeed, the caller does not even have to be a member of the multicast group to issue an invitation. Bridges are introduced like regular session members; they may take over branches of a mesh through the Replaces header, without the participants needing to be aware that there is a bridge serving the conference. SIP also supports conferences through full-mesh, also known as *multi-unicast*. In this case, the client maintains a point to point connection with each participant. While this mode is very inefficient, it is very useful for small conferences where bridges or multicast service are not available. Full meshes are built up easily using the Also header. For example, to add a participant C to a call between users A and B, user A would send an INVITEto C with an Also header containing the address of B. Mixes of multi-unicast, multicast, and bridges are also possible.

SIP can also be used to set up calls to several callees. For example, if the callee's address is a mailing list, the SIP server can return a list of individuals to be called in an Also header. Alternatively, a single address, e.g., `sysadmin@acme.com`, may reach the first available administrator. Servers can fan-out invitation requests, including sending them out via multicast. Multicast invitations are particularly useful for inviting members of a department (`product_team@example.com`) to a conference in an extremely efficient manner without requiring central list administration.

Multicast invitations also allow a small conference to gradually and smoothly migrate to a large-scale Mbone-type conference (such as the ones described by Handley in this issue) without requiring separate protocols and architectures.

6 Additional Protocols

We have so far touched on the two major protocol components required for Internet telephony, namely RTP and SIP. However, a number of other protocol components are very useful for more rich services; we briefly mention them here.

When an IP host wishes to communicate with a GSTN endpoint, it must do so by means of an Internet Telephony Gateway (ITG). This will generally require the host, or a proxy for the host, to eventually find the IP address of a telephony gateway appropriate for terminating the call. The selection of such a gateway is not an easy problem. There are many factors which contribute to the process: cost of completing the call, billing methods supported (credit card, debit card, e-cash), signaling protocols supported (SIP, H.323), media codecs supported, service provider, etc. Generally, the client will need to provide input, indicating many of these preferences, for the selection to take place. In essence, a telephony gateway is a service, and the client desires to select a server for this service based on some criteria. Extensions to the Service

Location Protocol [37] for discovery of wide area services [38] have been proposed which seem applicable to telephony gateways [39].

As discussed in Section 5.1, SIP can be used to translate addresses as the request moves from SIP server to SIP server. This translation can be based on any criteria, such as caller and time of day. However, there is no currently defined mechanism for allowing a user to express its preference for how an address is to be translated. For example, if `alice@acme.com` calls `bob@widgets.com`, Alice will send a SIP request to the SIP server at `widgets.com`. Bob would like the calls forwarded to his PC if he is connected and if Carol calls, but the calls should be forwarded to his voice mail (through an ITG) otherwise. This requires Bob to express preferences for such translations to the server. We are currently developing a call processing syntax which can be uploaded to a server in a SIP REGISTER message. Such a language is to be standardized in the IETF iptel (IP Telephony) working group.

With widespread use of IP telephony, IP telephony voicemail is likely to follow. In order to support retrieval and recording of voicemail (which could easily include video), a protocol is necessary to give a user "VCR" like controls over the voicemail server. The Real Time Streaming Protocol (RTSP) [40] is used for this purpose. RTSP allows a client to instruct a media server to record and playback multimedia sessions, including functions such as seek, fast forward, rewind, and pause. RTSP, like SIP, is a textual protocol similar in format to HTTP. RTSP integrates easily with SIP, in fact. A user can use SIP to invite a media server or voicemail server to a multimedia session, and then use RTSP to control operation of the during the session.

7 Protocol Integration

The previous sections have described the basics of the protocol mechanisms for Internet telephony: SIP, RTP, wide area service location, and RTSP. In this section, we put the elements together and show how they can be used for a complex service.

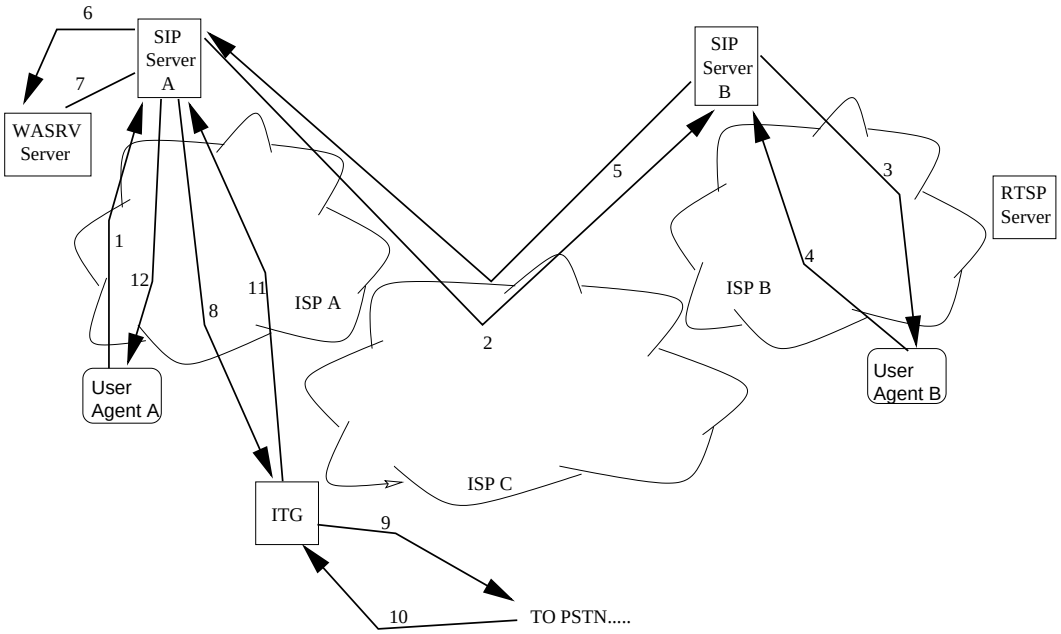

Figure 5: Internet Telephony Components

Figure 5 shows an IP network consisting of three Internet Service Providers (ISP's), A, B, and transit ISP C. ISP A provides a SIP proxy server A, and a wide area service location (WASRV) server. ISP B provides

a SIP proxy server as well, in addition to an RTSP server for voicemail. ISP C provides a telephony gateway (ITG). User A in ISP A wishes to call user B in ISP B. User A's SIP user agent client is configured to use SIP server A as a proxy for all call requests. To call B, A sends a SIP INVITE message to the SIP Server A (1), indicating the name of user B in the To field (`john_b@ispb.com`). SIP server A looks up the domain ispb.com in DNS, and obtains the IP address of server B. Acting as a proxy, it forwards the SIP INVITE to SIP server B (2). Server B checks its records, and finds a set of call processing instructions for the user. The instructions indicate that the user is first to be contacted, via proxy, at their PC. If no one answers, the server should return two alternate locations for user B to the requester: a telephone number and a voicemail server. The SIP server then follows these instructions, and forwards the INVITE message to user B's PC (3).

User B has instructed his SIP user agent server software not to accept any calls. User B's SIP user server thus responds with an error message to SIP server B (4). The SIP server then sends a redirect response to SIP server A (5). The response is in the 300 class set, and includes in the Location fields two alternate addresses. The first is a telephone URL (`tel://+1-732-555-1212`), and the second an RTSP URL for a media server. The location headers can indicate preferences, and the response indicates that the caller should try the phone URL first.

The SIP server A then tries to call the user at the GSTN number. To do this, it first queries a WASRV server (6). It provides the server with the telephone number to contact, and user preferences about billing methods (such as credit card payment). These preferences can be supplied to the SIP server by the user in any number of ways, including manual entry by a administrator. The WASRV server queries its database, and returns the address of an appropriate gateway (7).

The SIP server then sends the original SIP INVITE to the gateway (8). The gateway makes a call to the PSTN (9), but the line is busy (10). The gateway indicates this to the server via a SIP error message (11). The server A still has the RTSP URL as a potential contact point for the client. Since it does not understand how to process RTSP URL's, it returns this URL to A's user agent client via a SIP redirection response (12). User A can then contact the RTSP server, and leave a message for user B.

This example illustrates a number of different facets of the architecture. The first is the "hopping" of SIP INVITE requests between elements, with each element accessing directories (such as a WASRV server) or databases as needed. The behavior of each SIP server is dependent on its local programming and implementation. In the example here, SIP server B had been programmed with user preferences about call handling. Server A was programmed with user preferences about billing, and had access to WASRV services to complete calls to PSTN destinations. This heterogeneity allows for the definition of new services, and for differentiation among servers.

Another facet of the architecture is the smooth integration with other services (such as tel: and rtsp:). SIP allows for any type of URL to be carried in the From,To, Location, and Also fields. This allows calls to be handed off to other protocols and services when needed.

8 Conclusion and Future Work

We have presented a portion of a protocol suite for supported advanced IP telephony services on the Internet. This suite includes the Real Time Transport Protocol (RTP) for transport, the Session Initiation Protocol (SIP) for signaling, and the Real Time Streaming Protocol (RTSP) for stored media retrieval. These protocols are independent and modular, and when combined with billing, service discovery, and resource reservation protocols, form a complete architecture for future services.

However, much work remains. While IPtel promises to add greater flexibility to telephony services and speed their implementation, IPtel first has to overcome a number of well-known problems, including unpredictable quality of service in the wide area, the lack of reliable and cheap end systems, Internet unreliability, and the lack of a billing infrastructure.

The Internet also currently lacks a generally accepted reliable multicast protocol. Beyond application sharing, voting (i.e., distributed counting) and floor control (i.e., a distributed queue) require such reliability.

As mentioned earlier, gateways to existing telecommunications systems will have to play a large role in the transition to an Internet-based telecommunications infrastructure. We are currently investigating the interconnection of SIP servers with the SS7 (ISUP) protocol [4], so that a gateway can become a first-class citizen in the telephone network (in other words, interface to the telephone network using an NNI as opposed to UNI). Gateways of SIP to H.323 are also under study.

References

[1] H. Schulzrinne, "A comprehensive multimedia control architecture for the Internet," in *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, (St. Louis, Missouri), May 1997.

[2] M. Handley, H. Schulzrinne, and E. Schooler, "SIP: Session initiation protocol," Internet Draft, Internet Engineering Task Force, Mar. 1998. Work in progress.

[3] M. Handley, J. Crowcroft, C. Bormann, and J. Ott, "The internet multimedia conferencing architecture," Internet Draft, Internet Engineering Task Force, July 1997. Work in progress.

[4] T. Russell, *Signaling system #7*. New York: McGraw-Hill, 1995.

[5] Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," Request for Comments (Draft Standard) 1771, Internet Engineering Task Force, Mar. 1995. (Obsoletes RFC1654).

[6] B. Braden, 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, Oct. 1997.

[7] P. Pan and H. Schulzrinne, "Yessir: A simple reservation mechanism for the internet," Technical Report RC 20697, IBM Research, Hawthorne, New York, Sept. 1997.

[8] C. Rigney, "RADIUS accounting," Request for Comments (Informational) 2139, Internet Engineering Task Force, Apr. 1997. (Obsoletes RFC2059).

[9] A. Rubens and P. Calhoun, "DIAMETER base protocol," Internet Draft, Internet Engineering Task Force, Mar. 1998. Work in progress.

[10] International Telecommunication Union, "Terms and definitions," Recommendation B.13, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, 1988.

[11] L. Zhang, A. Mankin, J. Stewart, T. Narten, and M. Crawford, "Separating identifiers and locators in addresses: An analysis of the GSE proposal for IPv6," Internet Draft, Internet Engineering Task Force, Mar. 1998. Work in progress.

[12] H. Schulzrinne, "Re-engineering the telephone system," in *Proc. of IEEE Singapore International Conference on Networks (SICON)*, (Singapore), Apr. 1997.

[13] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS control of multimedia applications based on RTP," *Computer Communications*, vol. 19, pp. 49–58, Jan. 1996.

[14] 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.

[15] C. Strathmeyer, "Feature topic: Computer telephony," *IEEE Communications Magazine*, vol. 34, Apr. 1996.

[16] Versit Consortium, "Computer telephony integration (CTI) encyclopedia," Tech. Rep. Release 1.0, Versit Consortium, Oct. 1996.

[17] C. Low, "The internet telephony red herring," in *Proceedings of Global Internet*, (London, England), pp. 72–80, Nov. 1996.

[18] C. Low, "Integrating communications services," *IEEE Communications Magazine*, vol. 35, pp. –, June 1997.

[19] P. Moulton and J. Moulton, "Telecommunications technical fundamentals," technical handout, The Moulton Company, Columbia, Maryland, 1996. see http://www.moultonco.com/semnotes/telecomm/teladd.htm.

[20] D. Cohen, "A protocol for packet-switching voice communication," in *Proc. of Computer Network Protocols Symposium*, (Liege, Belgium), Feb. 1978.

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

[22] J. Rosenberg and H. Schulzrinne, "Timer reconsideration for enhanced RTP scalability," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (San Francisco, California), March/April 1998.

[23] C. Zhu, "RTP payload format for H.263 video streams," Request for Comments (Proposed Standard) 2190, Internet Engineering Task Force, Sept. 1997.

[24] T. Turletti and C. Huitema, "RTP payload format for H.261 video streams," Request for Comments (Proposed Standard) 2032, Internet Engineering Task Force, Oct. 1996.

[25] L. Berc, B. Fenner, R. Frederick, and S. McCanne, "RTP payload format for JPEG-compressed video," Request for Comments (Proposed Standard) 2035, Internet Engineering Task Force, Oct. 1996.

[26] R. Civanlar, G. Fernando, V. Goyal, and D. Hoffman, "RTP payload format for MPEG1/MPEG2 video," Request for Comments (Proposed Standard) 2250, Internet Engineering Task Force, Jan. 1998. (Obsoletes RFC2038).

[27] H. Schulzrinne, "RTP profile for audio and video conferences with minimal control," Request for Comments (Proposed Standard) 1890, Internet Engineering Task Force, Jan. 1996.

[28] O. Hodson, I. Kouvelas, O. Hodson, M. Handley, and J. Bolot, "RTP payload for redundant audio data," Request for Comments (Proposed Standard) 2198, Internet Engineering Task Force, Sept. 1997.

[29] J. Rosenberg and H. Schulzrinne, "An RTP payload format for generic forward error correction," Internet Draft, Internet Engineering Task Force, Mar. 1998. Work in progress.

[30] J. Rosenberg and H. Schulzrinne, "An RTP payload format for user multiplexing," Internet Draft, Internet Engineering Task Force, Jan. 1998. Work in progress.

[31] D. Katz, "IP router alert option," Request for Comments (Proposed Standard) 2113, Internet Engineering Task Force, Feb. 1997.

[32] W. Almesberger, J.-Y. L. Boudec, and T. Ferrari, "Scalable resource reservation for the internet," in *Proc. of IEEE Conference Protocols for Multimedia Systems – Multimedia Networking (PROMS-MmNet)*, (Santiago, Chile), Nov. 1997. Technical Report 97/234, DI-EPFL, Lausanne, Switzerland.

[33] K. Nichols, V. Jacobson, and L. Zhang, "A two-bit differentiated services architecture for the internet," internet draft, Bay Networks, LBNL and UCLA, Nov. 1997.

[34] 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.

[35] L. Masinter, P. Hoffman, and J. Zawinski, "The mailto URL scheme," Internet Draft, Internet Engineering Task Force, Jan. 1998. Work in progress.

[36] T. Howes, S. Kille, and M. Wahl, "Lightweight directory access protocol (v3)," Request for Comments (Proposed Standard) 2251, Internet Engineering Task Force, Dec. 1997.

[37] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan, "Service location protocol," Request for Comments (Proposed Standard) 2165, Internet Engineering Task Force, June 1997.

[38] J. Rosenberg, H. Schulzrinne, and B. Suter, "Wide area network service location," Internet Draft, Internet Engineering Task Force, Dec. 1997. Work in progress.

[39] J. Rosenberg and H. Schulzrinne, "Internet telephony gateway location," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (San Francisco, California), March/April 1998.

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