Internet Engineering Task Force
INTERNET-DRAFT
draft-schulzrinne-iptel-arch-00.ps

Henning Schulzrinne
Columbia U.

June 1, 2002
Expires: December 2002

Internet Telephony Architecture Roadmap

Status of this Memo

Copyright Notice

Contents

1 Introduction **2**

2 The Public Switched Telephone Network **3**

3 Differences from Traditional Telephone Systems **4**

4 Architecture **5**
4.1 Data . . . 5
4.1.1 DTMF Carriage . . . 5
4.1.2 Dealing with Packet Loss . . . 5
4.1.3 Header Compression . . . 6
4.1.4 Voiceband Data and Fax . . . 6
4.1.5 Adaptive Applications . . . 7
4.2 Control . . . 7
4.2.1 Session Setup . . . 7
4.2.2 Streaming Media and Multimedia Messaging . . . 10
4.2.3 Programming Interfaces . . . 11
4.2.4 Resource Session Setup . . . 12
4.3 Interworking with the PSTN . . . 14
4.3.1 Signaling protocol translation . . . 14
4.3.2 Calls from the PSTN to Internet telephones . . . 14
4.3.3 Calls from VoIP devices to the PSTN . . . 15

5 Security Considerations **16**
5.1 Caller Identification 16
5.2 Call Blocking Without Revealing the Caller Identity 17
5.3 Caller Anonymity 17
5.4 Denial of Service 18
5.5 Unsolicited Calls 18

6 Acknowledgements **18**

Abstract

Interactive voice and video communication is slowly migrating to Internet protocols and the public Internet. However, third-generation wireless networks are likely to use Internet protocols extensively. This memo summarizes the IETF protocol architecture for providing such services in both landline and wireless environments based on IETF-specified protocols. It describes the basic functionality of the major components and provides an overview of how the various protocols can be used together to provide Internet telephony services. It also points out where additional standardization efforts may be needed. It is also intended for those seeking an introduction to the emerging architecture.

1 Introduction

The most basic of electronic communications services, telephony, is likely to undergo a profound technological transformation in the next decade or so, with a transition from circuit-switched services to one based on packet switching. This memo summarizes the basic challenges to the Internet architecture and operational IP networks as well as the current state of standardization developments within the IETF. Not all of these standardization efforts are equally mature; also, other organizations such as ETSI or ITU may pursue different avenues. Given that terminology and backgrounds of those working on Internet topics and those coming from a telephony environment still differ significantly, this document tries to avoid assuming an in-depth knowledge of traditional telephony technology.

Unlike other services like email delivery or web access, Internet telephony is not a single core protocol, but rather requires the use of about half a dozen different protocols in the transport and application layer, not counting protocols needed for authentication, authorization and accounting (AAA) and quality-of-service. Some of these protocols may also be used for other services beyond Internet telephony.

Also, Internet telephony is strongly influenced by the existing architecture, services and user expectations of the billion-node public switched telephone network. In a sense, the situation is similar to the early days of email deployment, where large amounts of efforts had to be expended on connecting to non-Internet messaging services. However, in this case, the service is considered a critical infrastructure, with public safety implications.

The document has several goals and audiences:

- Introduce and survey the protocols making up the IETF Internet telephony architecture;

- Indicate how the protocols cooperate in typical call scenarios, both from Internet host to Internet host as well as to and from PSTN-connected devices;

- Summarize some of the open technical issues that remain to be addressed.

The document is aimed at tying together the often disjoint efforts happening in about half a dozen working groups of the IETF. Creating modular protocols that make minimal assumptions about the rest of the protocol space is a virtue [1], but it can make understanding the overall picture a bit difficult. This document attempts to offer an introduction to those new to the area of Internet telephony, but also indicates where additional standardization and documentation efforts may be needed.

There are a large number of public policy issues connected to the evolution of Internet telephony. We will only allude to them in the context of emergency call services (Section **??**), but other issues include lawful intercept and the administration of the telephone numbering system.

In this document, we will use the term voice-over-IP and Internet telephony interchangeably, although neither captures the full extent of the service. Internet telephony is not limited to the "big-I" Internet, but often used in private IP-based networks. It is also not limited to voice, but in almost all cases, extends to other media, either for pure IP-based services or when communicating with, for example, ISDN-based teleconferencing end points.

The document is organized as follows. In Section 2, we briefly summarize the operational modes for Internet telephony. Section **??** highlights some of the unique challenges that Internet telephony poses to the existing deployed Internet, while Section 3 tries to show that using PSTN models does not adequately account for the fundamental differences between the existing PSTN and the Internet architecture. Section 4 attempts to provide a high-level overview of the core Internet telephony protocols and how they work together in typical calls. A number of open issues are discussed in Section **??**, with a section on security issues (Section 5) rounding out the document.

2 The Public Switched Telephone Network

In this document, we will use the term Public Switched Telephone System (PSTN) to refer to the existing, circuit-switched telephone system. This system consists of transmission facilities, either TDM-based (e.g., using DS0 through T-3 and higher circuits) or ATM (AAL1)-based, controlled by signaling systems, generally using Signaling System #7 (SS7), a specialized packet networking protocol stack. Land-line end terminals are connecting either using analog lines, with hook-switch and tone signaling, or digitally, via ISDN, with its own user-to-network signaling protocols (Q.931). Wireless terminals can also be connected either via analog radio transmission (e.g., using the first-generation wireless technology AMPS in the U.S.) or digitally, through a variety of time-division, frequency-division or code-division multiplexed "second generation" systems. Some people also refer to this as the "legacy phone system", "black telephones", "plain old telephone service" (POTS). Since the telephone system is run by private companies in many countries, the term Global Switched Telephone System (GSTN) is also found instead of PSTN, but used infrequently outside of formal documents.

We will use the terms Internet telephony and voice-over-IP (VoIP) interchangeably. Even though telephony and voice-over-IP imply a service limited to speech, it must be emphasized that almost all IETF developments are inherently multi-media. (Megaco is somewhat of an exception as it focuses on gateways to voice-only systems.)

The initial transformation of the telephone system appears to be taking place along several separable paths:

Wide-area signaling: Here, traditional telephony signaling protocols are simply carried over IP and a reliable transport protocol such as SCTP [2], as discussed within the IETF SIGTRAN working group. This has no further impact on either the Internet architecture or the telephone system.

Wide-area transport: In this model, telephone switches are connected by IP "trunks", replacing traditional TDM or ATM AAL1 circuits for voice transport. Again, the impact on end users or Internet architecture is minimal. Since the number of such trunks is relatively small, per-flow resource reservation appears feasible if it is considered desirable to carry VoIP and other Internet traffic over the same network. Already, about 5-10% of international phone traffic uses IP transport, so the transformation of the phone system is more than just theoretical.

PBX: Private branch exchanges (PBXs) are being replaced by LAN-based Internet telephony systems, primarily to avoid having to maintain two separate wiring plants and management infrastructures. In the future, enhanced services may also provide additional motivation. Devices in these systems are reached from the outside as if they were regular telephones. As long as these systems connect to the PSTN just like PBXs, the impact on the Internet and phone system are minimal. Also, this transformation requires the least amount of protocol support in the areas of naming, gateway location, billing or quality of service.

IP access networks: Providers of DSL and cable modem Internet access networks have an economic incentive to also offer VoIP services over that infrastructure. This is particularly attractive for residences and small businesses with multiple lines, as the number of "lines" in an IP-based system can be far greater for the same amount of resources. If the network provides quality of service guarantees, some form of charging is likely required.

Tail-end hop off: In tail-end hop-off mode, Internet hosts place phone calls to traditional PSTN subscribers, with gateways translating between them. The primary problem here is locating the gateway, as described in Section 4.3.3.

3 Differences from Traditional Telephone Systems

Internet telephony differs in a number of fundamental ways from traditional telephony.

Separation of signaling and data paths: While modern digital telephone systems use out-of-band signaling, with a physically separate signaling network, signaling and voice streams coincide at various switches in the datapath as session setup and resource reservation (here, circuit establishment) are bundled into the same protocol. In the Internet, data, resource reservation signaling and session signaling share the same infrastructure, but data and session signaling generally coincide only at the end points. Resource reservation, if used, is a separate protocol and may or may not follow the data path hop-by-hop. For example, for diff-serv with bandwidth brokers, session setup will traverse the same set of AS as media streams, but may not necessarily follow the same path within each AS.

The separation of call state and resource state may also lead to calls that are very long-lived, while using no network resources. For example, groups of people working together may maintain an "open line" that allows instant communication, without explicit call setup, but where voice data flows only sporadically. (This is known as "hoot and holler" in trading floor applications.) Such calls may last effectively indefinitely.

The separation of calls into a session state and a resource state also removes the traditional notion of "lines" from end systems. The number of simultaneous call appearances of an end system is no longer limited by the bandwidth of the access link, but rather the maximum number of simultaneously active calls, i.e., calls where caller or callee are actually generating non-silence voice packets.

Uniform signaling protocols: In the PSTN, there is a strong distinction between user-to-network (UNI) and network-to-network (NNI) signaling protocols, where in-band tones and Q.931 serve as UNI protocols and the SS7 suite as NNI protocols. This distinction is not made for Internet telephony, where the same protocols are used in either case for both session and resource setup.

Trust model: SS7 does not incorporate any means of carrier-to-carrier authentication. Data, such as originating phone number, are trusted by the terminating carrier.

Traffic model: Voice traffic in Internet telephony is generally silence-suppressed, while silence suppression is rarely used in the PSTN. This increases efficiency, but also leads to burstier traffic.

4 Architecture

The IETF protocol architecture for Internet telephony can be roughly divided into three parts: data transport (Section 4.1), control (“signaling”) (Section 4.2) and elements needed for interoperation with the PSTN (Section 4.3).

4.1 Data

Regardless of whether H.323 or SIP or Megaco is used for signaling, existing VoIP systems use RTP [3, 4] to carry voice samples, including the architectures based on H.323 and SIP (see below). (This is not true for other multimedia data, where proprietary transport protocols are more common.) RTP provides information to do resequencing, payload format description, talkspurt detection, intramedia and intermedia synchronization and application-layer encryption. However, it appears that many applications are only implementing a subset; in particular, RTCP, the protocol part of RTP used for participant information and QoS feedback, is not as widely implemented. This is partially due to the fact that participant information in two-party calls is available via the signaling protocol. However, RTCP is useful even in two-party calls to measure the currently available quality of service, in particular round-trip delay, loss and delay jitter.

Three extensions of RTP are relevant for VoIP: carriage of DTMF tones [5], forward error correction [6, 7] and header compression [8].

4.1.1 DTMF Carriage

DTMF and other in-band signaling tones used in the PSTN may not survive coding with high-compression audio codecs, such as G.723.1 or G.729. Also, since reliable detection of such tones requires non-trivial digital signal processing, it is desirable to limit detection to gateways, rather than having it be carried out in IP end systems. Thus, instead of coding waveforms, RFC 2833 describes a simple encoding that carries DTMF and other signaling tones as indices rather than waveforms.

4.1.2 Dealing with Packet Loss

There are two basic approaches to reducing application-visible errors. First, we can transmit two (or more) different, interleaved streams of audio or video data, consisting of a primary encoding and a lower-rate “fill-in”’ codec which is used only if a packet from the primary encoding has been lost [9]. This works best if there is a large difference in rates between the primary and secondary codec and if the secondary codec can be used to speed up the convergence of the encoding parameters of the primary codec. For low bit-rate codecs such as G.732.1, G.729 or AMR, only very low-rate codecs such as LPC-10 (2.4 kb/s) yield acceptable overhead of about 25%. This mechanism also adds substantial CPU overhead at both sender and receiver, as each audio packet has to be encoded twice and may have to decoded twice in case of loss.

Alternatively, we can interleave redundant audio blocks from the same codec, using classical forward error correction techniques. This incurs low computational overhead, but may have larger delay depending on the block size [10]. There currently does not seem to be a comparison of delay, effort and perceptual quality for similar loss rates across the two methods.

4.1.3 Header Compression

Due to the delay requirements and loss sensitivity, audio packets are generally much shorter than typical non-ACK data packets. For example, a G.729 packet will contain 10 bytes of payload for each 10 ms of packetization interval, with 20 to 50 ms being typical packetization intervals. For IPv4 networks, the total header size, however, is 40 bytes, with 20, 8, and 12 bytes contributed by IPv4, UDP and RTP, respectively. For IPv6 networks, the overhead increases to 60 bytes. If encapsulation is used, this header overhead can increase further. For both IPv4 and IPv6, link-layer overhead (e.g., 26 bytes for Ethernet) and encapsulation overhead (e.g., for IPv6-in-IPv4 tunnels) may need to be added. Thus, even for maximum-length voice frames, the total transport efficiency will likely be no more than 50%.

While it is possible to compress the RTP header by itself by only transmitting first- and second-order differences, it is more effective to compress IP, UDP, and RTP at once (CRTP, [11]). CRTP transmits differences between successive packets; receivers request state updates if a packet has been lost. Unlike RTP-only compression, however, CRTP is obviously only useful for single links and must be implemented by routers. Since RTP packets have no distinct port numbers or IP protocol types, routers must use heuristics to detect whether a particular packet stream is using RTP.

Unfortunately, the CRTP header compression mechanism does not work well when the access link suffers from high packet loss rates or delays [12], common, for example, for wireless links. Robust header compression [8] addresses these issues by including periodic state updates.

4.1.4 Voiceband Data and Fax

In some cases, voiceband data has to be carried across a voice-over-IP infrastructure, particularly once an organization has gotten rid of all traditional PSTN connectivity. Voiceband data consists primarily of fax (effectively, a 9.6 kb/s or, less commonly, 14.4 kb/s modem) and data modems. Fax can be carried in four modes, reflecting different time scales for the confirmation of the fax content. The four modes are modem mode, real-time fax, session mode and store-and-forward mode [13].

In modem mode, an analog-to-digital converter simply sends the fax modem tones across the network, treating them as normal voice. This mode is rarely used since packet loss is likely to cause visible artifacts.

Real-time fax typically uses the T.38 [14] protocol to carry the standard fax compressed data across a UDP or TCP connection. T.38 sessions can be set up using H.323 or SIP [15, 16].

Store-and-forward fax transport TIFF graphics in MIME bodies [17, 18] via SMTP.

Session-based fax, where confirmation of delivery is received at the end of the session and where end systems are connected directly, has apparently not been specified.

Voice-band modems require no further protocol support, with G.711 as the likely audio codec. Clearly, the modem modes for fax and data are, at best, stop-gap measures, as they are extremely bandwidth-inefficient, turning an 80 kb/s IP channel (G.711 plus packetization overhead) into a 33.6 kb/s fax or modem channel. (Most fax machines operate at even lower speeds of 9.6 or 14.4 kb/s.)

4.1.5 Adaptive Applications

In the absence of resource reservation, it is generally agreed that all Internet services should share resources fairly. None of the deployed systems appear to implement traffic adaptation, although some commercial streaming systems do. However, beyond this basic agreement, a definition of fairness is harder to come by. For example, one could argue that applications should be TCP-friendly, i.e., send at roughly the same rate as a TCP connection on the same path. However, a pure per-five-tuple equivalence does not adequately reflect user value. Are the (typically) four TCP connections set up for a web page supposed to have the same or four times the throughput of a single audio connection? Does each audio or video layer count as one connection or do all layers count as one or does this depend on the particular implementation choices of the application? If routers were to start enforcing fairness on the basis of IP addresses and port pairs, it appears likely that applications would be encouraged to "spread" their traffic across several different port pairs. (Currently, layered codecs are defined to be spread across several different multicast addresses [19].)

Also, fairness may come at a price if adaptation requires the frequent exchange of feedback messages. The additional traffic of these messages, not useful for reliability as in TCP, may effectively double the per-conversation data rate. RTCP-based feedback adds generally less than 5% overhead and has other uses, but because the feedback interval is on the order of seconds, not round-trip times, applications controlled by such feedback can never react in exactly the same way as TCP applications. Also, it remains to be seen whether rapid changes in sending rates are appropriate for media content, as it could lead to distracting "fading" effects of images sharpening and blurring leading non-technical users to believe that "something is wrong with the set". (There are also anecdotal indications that user perception of audio and video quality is largely driven by the worst quality during a call.)

Adaptation may also be appropriate even if resource reservation is used [20], to maximize the overall utility of scarce resources. Generally, for multimedia, lower bit rates have higher per-bit utility. However, unless users can be assumed to be cooperative, such mechanisms require congestion-based pricing.

4.2 Control

4.2.1 Session Setup

Session setup establishes end-to-end, application-layer sessions, such as a phone call, a group of related call "legs" or a conference.

(The term "session" is unfortunately not defined consistently across IETF protocols. For example, a SIP session can consist of several RTP sessions. Several sessions can be used to invite to a single multicast conference session.)

There are three basic session setup mechanisms, namely peer-to-peer, master-slave and announcement-based.

Peer-to-Peer In the peer-to-peer system, the two communicating end systems exchange signaling information, possibly with the help of intervening "proxy" servers that, for example, hide end system location changes. (In email, the network location of the mail recipient is hidden from the mail sender via a well-known "meeting point" identified by the email address, the mail store, accessed from different locations via mail retrieval protocols such as POP [21] or IMAP [22]. Due to the real-time nature of session signaling, this approach is not suitable for Internet telephony.)

There are two peer-to-peer session setup protocols being implemented and deployed, namely the H.323 suite of protocols [23] and the Session Initiation Protocol (SIP) [24]. It is beyond the scope of this document to describe the differences, see [25, 26]. As an IETF-designed protocol, we will focus here on SIP. SIP is also the likely candidate for setting up multimedia sessions in the IP multimedia domain within third-generation wireless networks, being pursued within the 3GPP and 3GPP2 forums, and as the protocol to interconnect MGCs.[1]

SIP is a text-based protocol that borrows the basic syntax from HTTP, i.e., it uses a request line, followed by RFC822-style headers and a message body, typically containing a session description, such as SDP [19], but also possibly other information such as HTML or plain text explanations of call failures. Negotiation of session parameters takes place in a simple offer-response exchange, with the initial INVITE request containing the capabilities of the calling end system, and the response the capabilities of the called end system. This allows each side to tailor the data its send to the capabilities of the other side, but as discussed below, the scheme has its limits once more than a simple enumeration of media and codecs are required. End systems can request changes of call parameters by re-issuing INVITE requests during a call. With additional end system capabilities, this also allows a crashed end system to resume calls that existed before the crash. (This approximates the soft-state capability of protocols such as IGMP or RSVP, but refreshes that enable soft state [27] are an extension of SIP, not part of the core.)

SIP entities are identified by SIP URLs. It is anticipated that in the long term, the email address can also be converted mechanically into a SIP URL (e.g., alice@example.com becomes sip:alice@example.com). Telephone numbers can be reached in two ways, either by identifying the desired gateway (sip:+1-415-555-1234@mygw.com;user=phone), translation entity (sip:+1-415-555-1234@example.com, if the caller is in the example.com domain) or by a tel URL [28], as in tel:+1-415-555-1234. The latter needs to be resolved to a SIP URL by a server designated by the caller or locally via ENUM (see Section 4.3.3), since the URL contains no domain name. For SIP URLs, the request is sent to the entity identified by the domain name, using SRV DNS lookups (see below). Each entity that receives such a request can in turn translate the SIP URL and proxy the request to the next server. A request header tracks the locations visited to prevent loops. As an example, a request addressed to sip:alice@example.com may be translated by the example.com SIP server to sip:alice@example.uk, where it may become sip:alice@support.example.uk. The original destination of the request is maintained in headers of the request that remain unchanged as the request traverses the proxies. This is very roughly similar to the progress of an SMTP message from MTA to MTA.

SIP is the first major protocol that uses DNS SRV records [29] to locate servers based on a given SIP URL. Thus, all servers are identified by the domain, not by a host name or a service-specific hostname (such as sip.example.org). The use of SRV records also supports simple randomized load balancing across proxy servers, reducing the need for layer-four routing and various DNS round-robin schemes.

Since call setup delays are important, SIP chose to implement its own reliability mechanism, with shorter initial retransmit timers than standard TCP configurations. Also, the overall message volume for UDP is lower than for a TCP-based solution, which would typically (without transaction TCP) need an initial three-way handshake and then the actual exchange of application-layer signaling messages. However, SIP can use reliable transport protocols such as TCP or SCTP if their congestion control or reliability mechanisms are desired. To minimize the number of messages exchanged for a standard call setup, SIP implements two different reliability mechanisms, a four-message one for INVITE that accomodates long delays between initial request and final response and a standard two-message request-response exchange for simple requests that are expected to be answered immediately. (With hindsight, a single mechanism even at the cost of additional messages may have been preferable.)

[1] 3G systems will provide multimedia services in both the circuit-switched and packet-switched modes.

One of the distinguishing characteristics is that SIP has built-in support for proxies, in terms of authentication and the ability to detect routing loops. The use of explicitly configured "outbound" proxies will hopefully reduce the incentive for playing various "transparent" intercept tricks. Proxies can also "fork" requests, i.e., send several outgoing requests based on a single incoming request. The response forwarded upstream is the best response received from all the branches. With minor exceptions, only a single response can be forwarded upstream. This mechanism simplifies the common case of trying to locate a user identified by a generic address at a number of possible locations.

SIP has a built-in end system registration mechanism that temporarily binds an IP address to a generic user address. Each SIP user agent periodically sends REGISTER requests to its home registrar, the logical entity tracking user locations. Issues that occur when the user is visiting another network that is separated by a firewall from the home network are discussed in [30]. One solution is to create a temporary, but globally unique identifier that is registered with the registrars in the visited network and the home network.

Normally, only the first request within a call leg transits all proxies, while other requests, including the request terminating the call, travel directly between the two end systems. However, proxies can force requests to visit them by inserting a Record-Route header field in the request.

SIP is currently being extended in a number of ways. For example, a mechanism for indicating that a particular system supports a given extension has been proposed. A refresh mechanism requests that the other side periodically re-send INVITE requests to indicate liveness of the signaling connection or to refresh a connection with a crashed end system.

Services like call transfer and multi-party calls require SIP extensions. The current model is that a new request, REFER [31], asks the recipient to issue another request, such as an INVITE, to a third party. Since that request may not fail or succeed immediately, the initiator can also request that it be sent notifications on progress, using the SUBSCRIBE-NOTIFY mechanism described below.

As a text protocol, SIP is somewhat less efficient than some binary protocol encodings, although the majority of space is consumed by textual identifiers rather than protocol elements and thus difficult to reduce even if information were labeled by binary instead of textual tags. (To reduce the overhead, some of the more commonly used SIP header fields have one-letter abbreviations, reducing the per-header field labeling overhead to three bytes, compared to about four bytes for a typical TLV-style protocol.) A typical signaling request consumes about 300 bytes. If each message is encoded individually, standard Lempel-Ziv or similar compression can reduce the space required by about 25%. Better compression is likely if the same dictionary is reused across calls, as may be feasible if there is a semi-permanent TCP signaling connection between a node and an outbound proxy. Even uncompressed, the signaling overhead for a call is likely to represent only about one second of talk time even for compressed codecs. A number of efforts are underway to reduce the overhead significantly, probably by introducing a "shim" between the transport layer and SIP that compresses the SIP message between two SIP entities via a dictionary.

Since SIP is an out-of-band signaling protocol, it will have to be supported explicitly by NATs, through ALGs, and firewalls [32, 33, 34].

SIP typically sets up sessions between two end systems. However, it is also possible to have a third party set up a session between it and two other SIP user agents. These two user agents then exchange media directly, but have a signaling relationship only with the third-party controller [35, 36].

Master-Slave In the master-slave system, either one or both end systems, media gateways (MGs), in a two-party call are controlled by a media gateway controller (MGC). The media gateways can be individual Internet-connected telephones, small (residential) gateways connected to an Ethernet port on a cable or DSL modem, or large (trunk) gateways terminating circuit-switched trunk groups. If the two VoIP end systems are controlled by the same MGC, no other signaling protocols are needed. This is typically the case for closed, single-provider systems, such as a cable-based Internet service provider that wants to offer Internet telephony services internally. However, this approach scales poorly to large numbers of end systems, as a single (logical) controller has to be aware of the state of all end systems. It has been suggested that in larger or cross-provider networks, peer-to-peer signaling protocols set up calls between MGCs, so that the MGCs look like peer-to-peer signaling end points. An MGC with that configuration is often called a softswitch, although the term is not sufficiently precise to be technically useful.

It may also be useful to have individual SIP end systems act as MGCs. For example, a PC could control a local dedicated audio I/O device, also known as an Internet telephone, if its local operating system and other hardware are more amenable to low-latency audio transport. Thus, a single logical end system within a call or conference can be split into several physical devices, each individually network-attached. (See also the discussion below on conference bus protocols for an alternate approach.) In a rough sense, this recalls the separation between user-network (UNI) and network-network (NNI) protocols common in the PSTN world.

There are two master-slave control protocols in common use, namely the Media Gateway Control Protocol (MGCP) [37] and Megaco/H.248 [38, 39], with the latter being the standards-track effort, jointly developed with ITU SG 16. Among other syntactic differences, MGCP operates on “connections”, while Megaco has “contexts”, with several terminations.

When controlling end systems, MGCP and Megaco attempt to model the behavior of a standard telephone. They assume a 12-button user interface [40], ring and other tones and an unstructured text display. (In contrast, for example, a SIP call displays labeled information on caller name, address, subject, organization, urgency and referenced calls.) Thus, the software on the device has limited ability to provide services such as call filtering and cannot provide services such as call forwarding or call transfer. Such services need to be provided by the MGC. Typically, each such end system is controlled by a single MGC, although it may be possible to assign different virtual lines to different MGCs, allowing a single phone to be reachable and to make outgoing calls from different service providers (see Section **??**).

Session Announcement A second mechanism for session establishment is the announcement of sessions via multicast, similar to a distributed TV directory. Currently, SAP [41] is used to carry Session Description Protocol (SDP) [19] messages describing sessions or other session announcement multicast addresses [42]. Participants can be invited to SAP-announced sessions via SIP, simply by copying the session description from the SAP announcement into a SIP INVITE request.

This model is only efficient if local caches store session announcement, which are then queried by users, rather than users having to wait for session announcements to trickle in. Given that many current global sessions have a likely audience of a few hundred, worldwide distribution of the announcement at rates of even once a minute is not likely to be efficient, compared to more application-layer filtered mechanisms such as email or web pages.

4.2.2 Streaming Media and Multimedia Messaging

While generally considered a separate application, streaming media, i.e., the one-way delivery of mostly stored multimedia content to one or more recipients, Internet telephony can make use of streaming media, e.g., for the delivery of voicemail (or general multimedia mail) messages and announcements. For multimedia mail, treating these typically large messages as streaming media instead of as attachments decreases the delivery delay particularly for low-bandwidth clients, as they do not have to download the whole attachment via POP or IMAP. It is also more efficient if, as is common, the recipient decides after listening to the first few seconds that they are not interested in the content.

For delivery of multimedia messages to large groups, streaming is also more storage-efficient. Instead of storing one copy for each recipient on the mail server, the streaming media server only needs to store one copy and deliver it on demand. (For very large audiences and popular content, content distribution networks will replicate the content to servers closer to the recipient, but the number of copies will still be significantly smaller than the audience size under most circumstances.)

However, like any external reference, such as to an ftp or http URL, storage as a pointer to a streaming media object raises the issue of stale references and garbage collection and leaves control of the message lifetime primarily at the sender side. (Recipients can obviously copy the streaming media content locally.)

Currently, streaming media and call setup use two separate, but similar, protocols, RTSP [43] and SIP, respectively. Both are in a sense remote request/response protocols and it would be possible to extend SIP, for example, to incorporate all of the RTSP methods for playback control. Also, some of the concept of SIP, such as request routing and forking, may well be applicable to locating stored content in content distribution networks. However, given the installed base and alternate ways of achieving similar goals, for example by using the RTSP redirection facilities for locating the best instance of a streaming media object, such an effort may not be worthwhile. For systems that need both interactive and streaming media, a single parser can parse the basic request and header format for both protocols.

4.2.3 Programming Interfaces

A major motivation for deploying Internet telephony is the promise of easier "service creation", i.e., the ability of equipment vendors, carriers, enterprise customers and users to create new telephony services. Currently, despite attempts in the so-called Intelligent Network (IN) area dating back to the 1980s, telephony services have largely remained the same for about a decade, and their creation is limited mostly to vendors and some large carriers. Carriers and end users in particular hope to replicate the model of the web, where static content (HTML pages) was very quickly followed by a wide variety of widely-used dynamic content creation mechanisms, both proprietary and semi-standardized.

APIs: The oldest approach to service creation, with ancestors dating back to the Telephony API (TAPI) [44] and Java TAPI (JTAPI) [45] for end-system services, is the use of application programming interfaces. Current examples include Parlay [46] and JAIN [47]. Both approaches attempt to hide the detailed underlying protocol mechanisms from the programmer by postulating a common call model. The hope is that Internet and PSTN calls can be treated in almost the same manner.

sip-cgi: As one of the attempts to leverage familiar web concepts to Internet telephony, sip-cgi [48] builds on the cgi-bin model [49]. Each request or response invokes a per-user script that is passed information about the request or response via environment variables. A cgi-bin script only needs to provide the response body and occasionally the response code to the server; sip-cgi can invoke a larger range of actions, from responses to the original request to proxying the request to one or more destinations.

Sip-cgi is language-independent, but just as for cgi-bin, it is more difficult to provide a limited execution environment that allows untrusted user scripts to access the necessary registration information or selected databases without opening up the whole local server operating system to attack.

Sip-cgi scripts have only a very rudimentary mechanism to keep transaction or call state. As in cgi-bin, such state needs to be kept by the script itself, but unlike cgi-bin, sip-cgi offers a way for the script to have the server store opaque state information for a call.

SIP requests are passed to sip-cgi scripts as environment variables, one for each header field, with the script being responsible for parsing them.

servlets: SIP servlets [50] are an adaptation of HTTP servlets [51]. Servlets are invoked for requests and responses, but they can keep transaction information in class data. Servlets are written in Java, but the servlet engine can be added to servers written in other languages. Unlike sip-cgi, servlets can access SIP information in parsed form.

CPL: The call processing language (CPL) [52] is a special-purpose language for call handling in proxy and redirect servers. It is encoded as XML, making it easy to use tools such as XSL to render its structure. An instance of a CPL script is invoked for each INVITE request and then handles all branches within the transaction. Beyond request logging, there is no mechanism to keep call state. CPL intentionally does not provide the full features of a programming language. For example, it does not have loops or general-purpose variables. CPL does offer non-recursive subroutines. This makes it much easier to bound the runtime of a CPL script and check its correctness prior to execution.

VoiceXML: VoiceXML [53] is an XML DTD that provides interactive voice response services in PSTN and IP-connected end systems. It is executed in a “browser” that receives voice or DTMF inputs from the caller and then invokes scripts that can retrieve other scripts. It roughly models the notion of HTML forms, with voice and DTMF input replacing GUI-based forms. Unlike CPL, VoiceXML has typical programming language constructs such as variables and loops. The current version does not offer significant call control features and thus serves a different niche than APIs (such as Parlay or JAIN), sip-cgi, or CPL. The W3C is currently designing an enhanced version with improved end system call control features, such as call bridging.

Table 1 summarizes some of these differences for the call control mechanisms discussed above.

	APIs	servlets	sip-cgi	CPL
Language-independent	no	Java only	yes	own
Protocols	PSTN/SIP/H.323	SIP	SIP	SIP, H.323
Model	call	req./tran.	request	request
Secure	no	mostly	no, but can be	yes
End user service creation	no	yes	power users	yes
GUI tools w/portability	no	no	no	yes
Call initiation	yes	no	no	no
Multimedia	some	yes	yes	yes

Table 1: Internet telephony programming models

4.2.4 Resource Session Setup

Quality of service issues are described in RFC 2990 [54]. Here, we mention some specific concerns related to IP telephony.

IP telephony is generally well-suited for simple priority mechanisms with global rate allocations. For example, all marked UDP traffic could be assigned a separate AF class [55] or be assigned to the EF class

[56]. RFC 2598 indicates that with a simple priority queue (PQ), the 90%th percentile of the jitter is generally less than half a packet time. (Global rate allocations, e.g., via some variant of round-robin or WFQ scheme, prevent the IP telephony class from starving the best effort class.)

The bandwidth of the audio portion of individual VoIP calls is well-known and constant, as long as one ignores silence suppression. (With silence suppression, delay models become much more complicated [**?**, 57, 58], but the activity factor of around 40-50% remains relatively constant.)

Also, the overall telephone call volume is statistically predictable, given a certain user population and call activity, with a rich body of estimation models dating back to Erlang. Thus, with modest simplifications, the overall bandwidth need can be predicted accurately.

However, the simplicity and predictability seem to be limited to networks with two classes of traffic, rather than a multi-layer hierarchy [CITATION?]. It has been argued [Quotable evidence?] that most of the quality-of-service problems that make Internet telephony across wide-area networks unreliable can be traced to the access links, where such simple prioritization schemes are viable without the typical concerns about settlements and detailed accounting. However, this only works if the access link is dimensioned that IP telephony calls always find sufficient bandwidth. All IP telephones known to the author support a setable DSField, allowing immediate implementation.

At an access link, simple aggregate statistics [59] may well suffice as a basis for aggregate (rather than per-flow or per-user) accounting and billing.

New DiffServ scheduling techniques, such as the ABE (Alternative Best-Effort) service [60], offer a low-latency service suitable for Internet telephony.

On the IntServ side, current per-flow resource reservation mechanisms are less than ideally suited for Internet telephony. Among other problems, they lack any mechanism for indicating charges and billing information, including provision for common models where a single party pays for both directions of data transferred. They are overly complex for the simple problem of resource allocation, where sender-based protocols are likely to be simpler, only requiring a two-message request-response mechanism. Implementation complexity matters in particular for Internet telephony as many end systems are embedded systems, with very modest memory and OS resources. Efforts emerging from the NSIS BOF may address this issue.

A simple architecture, or profile of an existing resource reservation mechanism, that allows end systems to request unicast bandwidth resources within the assigned DiffServ class at the egress router or within a single administrative domain, combined with overdimensioning in the backbone, may address the practical QOS needs of Internet telephony. (For example, one could imagine a profile of RSVP that uses only PATH messages.)

Based on the above, it appears unlikely that inter-domain resource reservation is needed. One can only speculate as to whether it would be implementable and deployable if it were needed. Since existing telephone switches can easily store and manage the call state for the PSTN, using relatively low-performance processors of about 1 MIPS, it is not clear that common concerns about the scalability of per-flow resource reservations (as opposed to per-flow queueing) are grounded in implementation reality. The major problem is not likely to be keeping state in routers, but rather authenticating and aggregating inter-provider reservation records. However, this does not seem significantly more complicated than inter-provider dial-in roaming arrangements [61, 62].

MPLS [63] offers another, sub-IP, solution to provide traffic segregation between low-bandwidth, low-jitter and best-effort traffic. However, MPLS appears to be limited to intra-provider traffic engineering. It warrants further discussion whether IP-layer provisioning could accomplish some of the same goals with lower overall system complexity.

4.3 Interworking with the PSTN

Interworking with the PSTN requires

- translating media (audio) between the constant bit stream of a PSTN circuit and IP packets;
- translating call signaling between IP signaling and the numerous PSTN signaling protocols that have evolved over the years (Section 4.3.1);
- translating between identifiers, such as telephone numbers and SIP URIs;
- determining the transition point between the circuit and packet-switched environments.

4.3.1 Signaling protocol translation

In circuit telephony, there are two distinct types of signaling protocols, namely for the user-to-network interface (UNI) and the network-to-network interface (NNI). This division reflects the traditional asymmetry between end system and network capabilities, as well as a notion of trusted networks vs. untrusted end systems.

When connecting VoIP and the PSTN, the VoIP "cloud" can, viewed from the PSTN side, look either like a group of end systems (or a PBX) or like a peer network. For the former, ISDN signaling using Q.931 appears to be preferred over more traditional channel-associated signaling (CAS). Basic voice calls can be translated fairly easily in both directions, but much of the enhanced information in SIP session setup messages, for example, has no equivalent in Q.931 and will thus be lost.

The dominant NNI signaling protocol in modern circuit-switched telephone systems is ISUP, the ISND User Part, of the Common Channel Signaling Protocol #7. Translation between SIP and ISUP has been described [64]; difficulties arise with some of the older features of the telephone system, such as overlap signaling [65]. (In overlap signaling, a signaling message is sent for each digit dialed, reflecting the absence of a dial-string termination indication on landline phones. This allows the originating exchange to be ignorant of the structure and length of telephone numbers.)

4.3.2 Calls from the PSTN to Internet telephones

In calls from the PSTN to Internet devices, the gateway needs to map telephone numbers to SIP or H.323 URLs. This can be done in a variety of ways, depending on where the gateway is located with respect to the numbers dialed. In the simplest case, the gateway simply terminates a set of numbers, acting no differently than a PBX or local telephone switch. Thus, there is a single gateway for each group of telephone numbers. The gateway can either generate SIP calls using the `tel` URL [28] and have another proxy translate these into `sip` URLs or generate SIP URLs directly, based on the number dialed. For example, if extension 4321 is dialed for the `example.com` gateway, the gateway generates the URL `sip:4321@example.com;user=phone`. The IP telephony devices for that number then need to register, using SIP, with the example.com server.

Alternatively, the gateway can use the ENUM lookup mechanism [66] to map E.164 [67] telephone numbers, i.e., globally unique numbers including the country code, to one or more URLs. The ENUM mechanism [66] uses DNS NAPTR records to map E.164 numbers, reversed from the E.164 "big-endian" into the DNS "little-endian" hierarchy order. For example, the number +1-201-555-6789 turns into a lookup of 9.8.7.6.5.5.5.1.0.2.1.e164.arpa. Each ENUM entry can contain any number of URLs, typically SIP, H.323 and mailto URLs [68, 69], but also HTTP URLs to map telephone numbers to the corresponding web page.

Unfortunately, secure updates of DNS are still not widely available [70] and scalability requires reasonably long TTL values, so that this mechanism is currently most likely to be suitable for relatively static information rather than updating, say, current SIP Contact information. [ISSUE? Partial updates of information.]

The ENUM mechanism raises a number of policy issues beyond the scope of this document, such as who "owns" telephone numbers and which entities should control various levels of the E.164 hierarchy [71].

4.3.3 Calls from VoIP devices to the PSTN

When placing calls from Internet devices to the PSTN, the end system has to locate the appropriate Internet telephony gateway (ITGW). Since every ITGW can place calls to (almost) every telephone number, we effectively have overlaid two fully-connected networks on top of each other, with a large number of feasible connection points.

Given the likelihood that gateways will charge for their services, the number of feasible gateways may well be much smaller, due to the need to establish a business relationship with the gateway prior to placing a call. This relationship can be direct, i.e., from the caller to the operator of the ITGW, or indirect, mediated by a clearing house.

One likely design criteria for such a network of ITGW is that as many numbers as possible are reachable as local calls. It appears difficult to estimate how many gateways would be needed to make every outbound call a local call, but as a first-order estimate, large ISPs have on the order of 400 to 500 POPs in the United States. Thus, it appears that a relatively small table, e.g., for about 276 area codes and lists of exchanges in the United States, would be sufficient as long as only a single provider is used. (Similar tables are currently commonly found in dialing software.)

For small sets of gateways, the Service Location Protocol [72] may be suitable to advertise all gateways, with a few service announcements each reflecting the different call costs. While SLP is normally only suitable for service discovery within a local network, DNS SRV records can advertise directory agents (DAs) for domains [73]. A broker could gather advertisements from a range of local DAs and in turn advertise them.

If information about ITGWs is more dynamic, e.g., due to load balancing, use of multiple providers and failover, a "routing" protocol is useful [74]. TRIP [75] offers such services. TRIP is a distribution protocol similar to BGP that distributes aggregated call routing information. The call routing information consists of E.164 prefixes, properties and the associated signaling proxy. Each aggregation point is then visited by call setup requests.

Consider an example (Fig. 1): A caller in Los Angeles wants to reach a number with the area code 212 in Manhattan. It generates a SIP request to the URL tel:+1-212-555-1234. The SIP request is directed to the locally configured outbound proxy. (The outbound proxy is discovered using DHCP or via a well-known multicast address.) The caller may first try to translate the phone number to a SIP URL, using ENUM, or delegate the task to the outbound proxy. If the ENUM lookup produces another telephone number, the lookup recurses.

The outbound proxy may try an ENUM lookup (Section 4.3.2) to see if that number is reachable via SIP. If it is, the request URI is rewritten accordingly and the request is routed to the SIP destination. TRIP is not used in this case. For the remainder of the section, we assume that either there is no ENUM entry or the ENUM entry itself points to another E.164 number.

Alternatively, if the caller wishes to use a particular carrier, it might address the call to sip:+1-212-555-1234@papabell.com, using DNS SRV records to reach the Papabell SIP server. The Papabell carrier then uses TRIP information to choose the appropriate gateway.

This outbound proxy also listens for TRIP announcements and builds a table of E.164-prefix/proxy pairs. For example, it might have an entry for +1-212, pointing to the SIP proxy nyc.gw.com. The proxy then rewrites the request URI to sip:+1-212-555-1234@nyc.gw.com and routes it to that server. Alternatively, it might have entries for +1 from a number of carriers, which in turn run TRIP inside their own network. In that case, the SIP request would visit the external proxy for the carrier, and then be handed along a hierarchical chain of SIP proxies until it reaches the proxy or SIP user agent responsible for the ITGW. A single proxy or SIP user agent can be responsible for any number of gateways. The ITGW or its SIP user agent (if MGCP or Megaco are used) indicates the IP address for the voice data in the session description. Note: The actual voice data, encapsulated in RTP packets, will travel directly to the gateway, end-to-end, without going through any of these proxies. Thus, only the initial signaling request and the TRIP data is handled by the proxies and the associated TRIP agents.

TRIP information is periodically exchanged between location servers (LS) that peer with each other using TCP connections, similar to BGP. Peering is set up manually. The protocol between the location server and the proxy has not been defined. The choice depends on whether the query is to be answered within the location server or if the announcements are to be made available to the SIP proxy. For querying, SLP [72] may be suitable, if they answer can be expressed as a URL. If the proxy wants access to the TRIP data itself, it may be easier to simply define a listen-only mode. (SLP may then be useful for discovering the TRIP server itself.)

Unlike BGP, it is quite likely that there will be any number of TRIP domains (Internet Telephony Administrative Domains, ITADs), not necessarily connected to each other. TRIP can serve as a routing protocol between internal and external peers.

A subset of TRIP [76] is used by ITGWs to announce their prefixes to TRIP location servers.

TRIP is a relatively complex protocol, as it inherits a lot of BGP functionality. It remains to be seen which parts of it are actually used in practice. Also, depending on the complexity of the attributes, aggregation may not be possible, so that essentially every gateway is exported.

Note that SIP and TRIP are independent. TRIP is not needed for end-to-end Internet telephony calls; also, TRIP can be used for other signaling protocols, such as H.323.

5 Security Considerations

Beyond the usual security considerations for Internet services, Internet telephony poses some particular security challenges. Some of these are discussed in the Security Considerations sections of the RTP [3] and SIP [24] specifications.

5.1 Caller Identification

Current PSTN systems allow users to identify the calling terminal, either by number or by subscriber name, where the caller can suppress this identifying information if it is a non-800 call. This identification serves three different purposes, namely recognizing known callers, e.g., to allow them to bypass call filters, to be able to identify harassing callers and to identify when the same telephone subscriber calls back. Note that the telephone system only identifies the telephone line, not the actual caller.

Providing reliable caller identification is somewhat more difficult for Internet telephony services. It can be provided by the caller or by the outbound proxy. Various types of user information and their verification status can be included in the SIP request via the SIP privacy extension [77]. The caller can sign the INVITE request with a personal certificate, recognized either by the callee, PGP-style, or signed by a CA. Given the relative lack of use of PGP certificates and the difficulty of obtaining personal certificates, this option may not be viable in the near term. Also, most existing personal certificates only certify that a particular user can be reached at the email address contained in the certificate. With the use of throw-away email addresses, this does not provide much protection against harassment calls, but is sufficient to recognize known callers. It is also not likely to be suitable for low-complexity embedded systems.

The outbound proxy can also certify the caller's identity by signing the request with its private key. As the number of providers is likely to be smaller than the number of individuals, this provides a level of service roughly similar to secure web pages. The callee then has to decide whether to trust the domain that is signing the request. The outbound proxy may use proxy authentication with a local secret to verify the caller's identity or may use local authentication, such as information derived from RADIUS or DIAMETER-based AAA systems.

Alternatively, the SIP request can use basic or digest authentication [78], with either a personal, per-callee user identity and password or a password for the whole destination domain. In the first case, the caller and callee have arranged for a shared secret and user name ahead of time, similar to how current web pages arrange for logins. However, this requires each caller to remember or note down a user name and password for each callee. As another approach, the caller can use the same login name, his SIP URI, everywhere. Each called domain can generate a random password, which is automatically emailed to the user's address, assuming that the SIP address can be used as an email address. This is about as secure as the typical personal certificates, but only provides identification of individuals already known to the callee. It still requires the callee to copy this password from the email to the Internet telephony client.

A third approach to identification extends the current web cookie model to Internet telephony. Once the caller has reached the callee, the UAS returns a cookie to the caller, which is then reused upon subsequent calls [79]. This only helps with the third mode of identification, distinguishing repeat callers, and only works if the caller calls from the same terminal. This mechanism remains to be standardized.

5.2 Call Blocking Without Revealing the Caller Identity

In the PSTN, both caller and callee trust The Telephone Company. This trust allows it to offer call blocking without the callee getting to know the caller's identity. Such a service is difficult to provide in Internet telephony systems as the caller is not likely to trust the destination domain.

5.3 Caller Anonymity

Telephone services have had a long tradition of strong anonymity where callers can be assured that the callee has no means of identifying the caller or determining that two calls are from the same location. (The latter is not true when calling from a payphone.) Without application-layer support, true anonymity is difficult to provide in Internet telephony, as IP addresses in call signaling and media streams reveal much about the caller's identity, even if the caller name is not disclosed in From SIP header fields. This differs from email, where reasonable anonymity can be achieved even without anonymizers by acquiring a temporary throw-away web-based email address. In SIP-based systems, the concept of an outbound proxy or a Route-designated second proxy could be used to introduce a network address translator that makes all signaling and media packets appear from that proxy. However, unlike for email, the choice of an anonymizer should probably either be randomized or chosen to minimize the delay penalty of triangle routing.

5.4 Denial of Service

Internet telephony is subject to additional denial of service threats. If an intruder can guess the IP address and port of one of the participants, he can easily send an RTP stream to that address, causing it to be mixed with the real conversation. Guessing the port number is not too difficult, particularly as it appears that certain applications restrict their port number range to be small, to simplify firewall configuration. IPsec AH may be used, but incurs a significant overhead, XX bytes per packet. If the SSRC of the packet stream is known, the recipient can filter the corresponding packets. Guessing the 32-bit SSRC is sufficiently unlikely, as the rogue sender has no information when he has hit the right value. (Obviously, the recipient should not return RTCP receiver reports to the rogue sender unless a sufficient number of packets has been received.) The receiver can discern the legitimate SSRC by listening to the first RTP packet, as long as the rogue sender is not already sending DOS streams. It may also be advisable to include this information in a yet-to-be-defined SDP option.

5.5 Unsolicited Calls

Since SIP requests can easily be generated automatically and possibly without cost to the caller, “phone spam” is a likely outcome, where either just signaling messages or calls with automated voice or video messages are generated. It may be possible to avoid at least automated systems by including a “Turing test” in the call setup (“please press 142 to connect to Alice”), but blocking of unsigned calls and various user identification mechanisms are also needed, as described earlier.

6 Acknowledgements

Brian Carpenter, Gonzalo Camarillo and Jonathan Lennox provided detailed comments. The discussion of the IESG and IAB influenced the description of SIP and TRIP.

References

[1] B. Carpenter and Ed, “Architectural principles of the internet,” RFC 1958, Internet Engineering Task Force, June 1996.

[2] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, “Stream control transmission protocol,” RFC 2960, Internet Engineering Task Force, Oct. 2000.

[3] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: a transport protocol for real-time applications,” RFC 1889, Internet Engineering Task Force, Jan. 1996.

[4] H. Schulzrinne, “RTP profile for audio and video conferences with minimal control,” RFC 1890, Internet Engineering Task Force, Jan. 1996.

[5] H. Schulzrinne and S. Petrack, “RTP payload for DTMF digits, telephony tones and telephony signals,” RFC 2833, Internet Engineering Task Force, May 2000.

[6] C. Perkins and O. Hodson, “Options for repair of streaming media,” RFC 2354, Internet Engineering Task Force, June 1998.

[7] J. Rosenberg and H. Schulzrinne, “An RTP payload format for generic forward error correction,” RFC 2733, Internet Engineering Task Force, Dec. 1999.

[8] C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannu, L.-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, and H. Zheng, “RObust header compression (ROHC): framework and four profiles: RTP, UDP, ESP, and uncompressed,” RFC 3095, Internet Engineering Task Force, July 2001.

[9] C. Perkins, I. Kouvelas, O. Hodson, V. Hardman, M. Handley, J. C. Bolot, A. Vega-Garcia, and S. Fosse-Parisis, “RTP payload for redundant audio data,” RFC 2198, Internet Engineering Task Force, Sept. 1997.

[10] J. Rosenberg, L. Qiu, and H. Schulzrinne, “Integrating packet FEC into adaptive voice playout buffer algorithms on the Internet,” in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (Tel Aviv, Israel), Mar. 2000.

[11] S. Casner and V. Jacobson, “Compressing IP/UDP/RTP headers for low-speed serial links,” RFC 2508, Internet Engineering Task Force, Feb. 1999.

[12] M. Degermark, H. Hannu, L.-E. Jonsson, and K. Svanbro, “Evaluation of CRTP performance over cellular radio links,” *IEEE Personal Communications Magazine*, vol. 7, Aug. 2000.

[13] L. Masinter, “Terminology and goals for internet fax,” RFC 2542, Internet Engineering Task Force, Mar. 1999.

[14] International Telecommunication Union, “Procedures for real-time group 3 facsimile communication over IP networks,” Recommendation T.38, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, June 1998.

[15] J. Mule and J. Li, “SIP real-time fax call flow examples and best current practice,” Internet Draft, Internet Engineering Task Force, Oct. 2001. Work in progress.

[16] E. Wedlund, W. Jiang, and H. Schulzrinne, “SDP extensions for fax over IP using T.38,” Internet Draft, Internet Engineering Task Force, Dec. 1998. Work in progress.

[17] L. McIntyre, S. Zilles, R. Buckley, D. Venable, G. Parsons, and J. Rafferty, “File format for internet fax,” RFC 2301, Internet Engineering Task Force, Mar. 1998.

[18] K. Toyoda, H. Ohno, J. Murai, and D. Wing, “A simple mode of facsimile using internet mail,” RFC 2305, Internet Engineering Task Force, Mar. 1998.

[19] M. Handley and V. Jacobson, “SDP: session description protocol,” RFC 2327, Internet Engineering Task Force, Apr. 1998.

[20] X. Wang and H. Schulzrinne, “Pricing network resources for adaptive applications in a differentiated services network,” in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (Anchorage, Alaska), Apr. 2001.

[21] J. Myers and M. Rose, “Post office protocol - version 3,” RFC 1725, Internet Engineering Task Force, Nov. 1994.

[22] M. Crispin, “Internet message access protocol - version 4rev1,” RFC 2060, Internet Engineering Task Force, Dec. 1996.

[23] 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.

[24] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, “SIP: session initiation protocol,” RFC 2543, Internet Engineering Task Force, Mar. 1999.

[25] H. Schulzrinne and J. Rosenberg, “A comparison of SIP and H.323 for Internet telephony,” in *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, (Cambridge, England), pp. 83–86, July 1998.

[26] I. Dalgic and H. Fang, “Comparison of H.323 and SIP for IP telephony signaling,” in *Proc. of Photonics East*, (Boston, Massachusetts), SPIE, Sept. 1999.

[27] S. Donovan and J. Rosenberg, “SIP session timer,” Internet Draft, Internet Engineering Task Force, Oct. 2001. Work in progress.

[28] A. Vaha-Sipila, “URLs for telephone calls,” RFC 2806, Internet Engineering Task Force, Apr. 2000.

[29] A. Gulbrandsen, P. Vixie, and L. Esibov, “A DNS RR for specifying the location of services (DNS SRV),” RFC 2782, Internet Engineering Task Force, Feb. 2000.

[30] H. Schulzrinne, “SIP registration,” Internet Draft, Internet Engineering Task Force, Apr. 2001. Work in progress.

[31] R. Sparks, “SIP call control - transfer,” Internet Draft, Internet Engineering Task Force, July 2001. Work in progress.

[32] C. Martin and A. Johnston, “SIP through NAT enabled firewall call flows,” Internet Draft, Internet Engineering Task Force, July 2001. Work in progress.

[33] J. Rosenberg, D. Drew, and H. Schulzrinne, “Getting SIP through firewalls and NATs,” Internet Draft, Internet Engineering Task Force, Feb. 2000. Work in progress.

[34] J. Rosenberg, J. Weinberger, and H. Schulzrinne, “NAT friendly SIP,” Internet Draft, Internet Engineering Task Force, July 2001. Work in progress.

[35] J. Rosenberg, J. Peterson, H. Schulzrinne, and G. Camarillo, “Third party call control in SIP,” Internet Draft, Internet Engineering Task Force, Nov. 2001. Work in progress.

[36] G. Camarillo, “Third party call control with SDP preconditions,” Internet Draft, Internet Engineering Task Force, July 2000. Work in progress.

[37] M. Arango, A. Dugan, I. Elliott, C. Huitema, and S. Pickett, “Media gateway control protocol (MGCP) version 1.0,” RFC 2705, Internet Engineering Task Force, Oct. 1999.

[38] F. Cuervo, N. Greene, C. Huitema, A. Rayhan, B. Rosen, and J. Segers, “Megaco protocol version 0.8,” RFC 2885, Internet Engineering Task Force, Aug. 2000.

[39] T. Taylor, “Megaco errata,” RFC 2886, Internet Engineering Task Force, Aug. 2000.

[40] P. Blatherwick, R. Bell, and P. Holland, “Megaco IP phone media gateway application profile,” RFC 3054, Internet Engineering Task Force, Jan. 2001.

[41] M. Handley, C. Perkins, and E. Whelan, “Session announcement protocol,” RFC 2974, Internet Engineering Task Force, Oct. 2000.

[42] R. Finlayson, “Describing session directories in SDP,” Internet Draft, Internet Engineering Task Force, Jan. 2001. Work in progress.

[43] H. Schulzrinne, A. Rao, and R. Lanphier, “Real time streaming protocol (RTSP),” RFC 2326, Internet Engineering Task Force, Apr. 1998.

[44] Microsoft, “Ip telephony with TAPI 3.0,” white paper, Microsoft, Redmond, Washington, 1999.

[45] Sun Microsystems, “The java telephony API,” Feb. 1997.

[46] P. Group. http://www.parlay.org.

[47] Sun Microsystems, “JAIN APIs for integrated networks.” Available at http://java.sun.com/products/jain/.

[48] J. Lennox, H. Schulzrinne, and J. Rosenberg, “Common gateway interface for SIP,” RFC 3050, Internet Engineering Task Force, Jan. 2001.

[49] http://hoohoo.ncsa.uiuc.edu/cgi/interface.html.

[50] A. Kristensen and A. Byttner, “The SIP servlet API,” Internet Draft, Internet Engineering Task Force, Sept. 1999. Work in progress.

[51] J. D. Davidson and D. Coward, “Java servlet specification v2.2,” Dec. 1999.

[52] J. Lennox and H. Schulzrinne, “CPL: A language for user control of internet telephony services,” Internet Draft, Internet Engineering Task Force, Nov. 2001. Work in progress.

[53] VoiceXML Forum, “Voicexml home page.” http://www.voicexml.org/.

[54] G. Huston, “Next steps for the IP QoS architecture,” RFC 2990, Internet Engineering Task Force, Nov. 2000.

[55] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, “Assured forwarding PHB group,” RFC 2597, Internet Engineering Task Force, June 1999.

[56] V. Jacobson, K. Nichols, and K. Poduri, “An expedited forwarding PHB,” RFC 2598, Internet Engineering Task Force, June 1999.

[57] J. N. Daigle and J. D. Langford, “Models for analysis of packet voice communications systems,” *IEEE Journal on Selected Areas in Communications*, vol. SAC-4, pp. 847–855, Sept. 1986.

[58] W. Jiang and H. Schulzrinne, “Analysis of on-off patterns in VoIP and their effect on voice traffic aggregation,” in *International Conference on Computer Communication and Network*, (Las Vegas, Nevada), Oct. 2000.

[59] F. Baker, K. Chan, and A. Smith, "Management information base for the differentiated services architecture," Internet Draft, Internet Engineering Task Force, Nov. 2001. Work in progress.

[60] P. Hurley, J.-Y. L. Boudec, P. Thiran, and M. Kara, "ABE: providing a low-delay service within best effort," *IEEE Network*, vol. 15, May 2001.

[61] P. Calhoun and C. Perkins, "Mobile IP network access identifier extension for IPv4," RFC 2794, Internet Engineering Task Force, Mar. 2000.

[62] B. Aboba and M. Beadles, "The network access identifier," RFC 2486, Internet Engineering Task Force, Jan. 1999.

[63] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," RFC 3031, Internet Engineering Task Force, Jan. 2001.

[64] G. Camarillo *et al.*, "ISUP to SIP mapping," Internet Draft, Internet Engineering Task Force, Mar. 2002. Work in progress.

[65] G. Camarillo, A. Roach, J. Peterson, and L. Ong, "Mapping of ISUP overlap signalling to SIP," Internet Draft, Internet Engineering Task Force, Jan. 2002. Work in progress.

[66] P. Faltstrom, "E.164 number and DNS," RFC 2916, Internet Engineering Task Force, Sept. 2000.

[67] International Telecommunication Union, "The international public telecommunication numbering plan," Recommendation E.164, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, May 1997.

[68] G. Vaudreuil and G. Parsons, "Voice profile for internet mail - version 2," RFC 2421, Internet Engineering Task Force, Sept. 1998.

[69] G. Vaudreuil and G. Parsons, "VPIM voice message MIME sub-type registration," RFC 2423, Internet Engineering Task Force, Sept. 1998.

[70] B. Wellington, "Secure domain name system (DNS) dynamic update," RFC 3007, Internet Engineering Task Force, Nov. 2000.

[71] G. Huston, "Management guidelines and operational requirements for the internet infrastructure domain ('ARPA')," Internet Draft, Internet Engineering Task Force, May 2001. Work in progress.

[72] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan, "Service location protocol," RFC 2165, Internet Engineering Task Force, June 1997.

[73] W. Zhao, H. Schulzrinne, C. Bisdikian, and W. Jerome, "The SLP service and remote discovery in SLP," Internet Draft, Internet Engineering Task Force, Mar. 2002. Work in progress.

[74] J. Rosenberg and H. Schulzrinne, "A framework for telephony routing over IP," RFC 2871, Internet Engineering Task Force, June 2000.

[75] J. Rosenberg, H. Salama, and M. Squire, "Telephony routing over IP (TRIP)," Internet Draft, Internet Engineering Task Force, Aug. 2001. Work in progress.

[76] J. Rosenberg, H. Salama, *et al.*, “Usage of TRIP in gateways for exporting phone routes,” Internet Draft, Internet Engineering Task Force, Dec. 2001. Work in progress.

[77] W. Marshall *et al.*, “SIP extensions for network-asserted caller identity and privacy within trusted networks,” Internet Draft, Internet Engineering Task Force, Mar. 2002. Work in progress.

[78] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, “HTTP authentication: Basic and digest access authentication,” RFC 2617, Internet Engineering Task Force, June 1999.

[79] D. Willis and B. Rosen, “SIP cookies,” Internet Draft, Internet Engineering Task Force, July 2001. Work in progress.

```
                    ----         ----
                   /    \       /    \               /
                  |TRIP  |     |TRIP  +-------+     /
                  |daemon+-----+daemon|       |   //
                   \    /       \    /      --+- /
                    -+--         ----      /    X
                     |                    |TRIP  |
                     |                    |daemon|
                 TRIP|212: nyc.gw.com      \    /
                     |                ...............................
call New York        |                .       |        New York     .
                 +---+--+             .  +----+--+     +-------+    .
  ------         |      |             .  |       |     |       |    .
   /  \  ------->+SIP   +-------------.->+  SIP  +....>+Media  |PSTN.
  -----\    SIP  |proxy |       SIP   .  |  UAS  |MGCP |gateway|    .
    \            +------+             .  +-------+     +---+---+    .
     \                                ...............................
LA    \                                                    |
       \                                                   |
        \                                                  |
         \                                                 |
          \                                                |
           \-----------------------------------------------+
                                audio over
                                RTP
```

Figure 1: TRIP