Voice Over Internet Protocol (VoIP)

BUR GOODE, SENIOR MEMBER, IEEE

Invited Paper

During the recent Internet stock bubble, articles in the trade press frequently said that, in the near future, telephone traffic would be just another application running over the Internet. Such statements gloss over many engineering details that preclude voice from being just another Internet application. This paper deals with the technical aspects of implementing voice over Internet protocol (VoIP), without speculating on the timetable for convergence.

First, the paper discusses the factors involved in making a high-quality VoIP call and the engineering tradeoffs that must be made between delay and the efficient use of bandwidth. After a discussion of codec selection and the delay budget, there is a discussion of various techniques to achieve network quality of service.

Since call setup is very important, the paper next gives an overview of several VoIP call signaling protocols, including H.323, SIP, MGCP, and Megaco/H.248. There is a section on telephony routing over IP (TRIP). Finally, the paper explains some VoIP issues with network address translation and firewalls.

Keywords*—*H.323, Internet telephony, MGCP, SIP, telephony routing over IP (TRIP), voice over IP (VoIP), voice quality.

NOMENCLATURE

ACD	Automatic call distributor.
ALG	Application level gateway.
ATM	Asynchronous transfer mode, a cell-switched communications technology.
BGP-4	Border gateway protocol 4, an interdomain routing protocol.
BRI	Basic rate interface (ATM interface, usually 144 kb/s).
Codec	Coder/decoder.
CR-LDP	Constrained route label distribution protocol.
DiffServ	Differentiated services.
DHCP	Dynamic host configuration protocol.
DSL	Digital subscriber line.
DTMF	Dual tone multiple frequency.
EF	Expedited forwarding.
FTP	File transfer protocol.
FXO	Foreign Exchange Office.
H.323	An ITU-T standard protocol suite for real-time communications over a packet network.
H.225	An ITU-T call signaling protocol (part of the H.323 suite).
H.235	An ITU-T security protocol (part of the H.323 suite).
H.245	An ITU-T capability exchange protocol (part of the H.323 suite).
HTTP	Hypertext transfer protocol.
IANA	Internet assigned numbers authority.
IETF	Internet engineering task force.
IntServ	Integrated services Internet.
ITAD	Internet telephony administrative domain.
ITSP	Internet telephony service provider.
ITU	International Telecommunications Union.
IP	Internet protocol.
IS-IS	Intermediate system-to-intermediate system, an interior routing protocol.
LAN	Local area network.
LDP	Label distribution protocol.
LS	Location server.
LSP	Label switched path.
LSR	Label switching router.
Megaco/H.248	An advanced media gateway control protocol standardized jointly by the IETF and the ITU-T.
MG	Media gateway.
MGCP	Media gateway control protocol.
MOS	Mean opinion score.
MPLS	Multiprotocol label switching.
MPLS-TE	MPLS with traffic engineering.
NAT	Network address translation.
OSPF	Open shortest path first, an interior routing protocol.
PBX	Private branch exchange, usually used on business premises to switch telephone calls.
PHB	Per hop behavior.
PRI	Primary rate interface (ATM interface, usually 1.544 kb/s or 2.048 Mb/s).

Manuscript received March 20, 2002; revised May 14, 2002.

The author is with AT&T Labs, Weston, CT 06883 USA (e-mail: bgoode@att.com).

Digital Object Identifier 10.1109/JPROC.2002.802005.

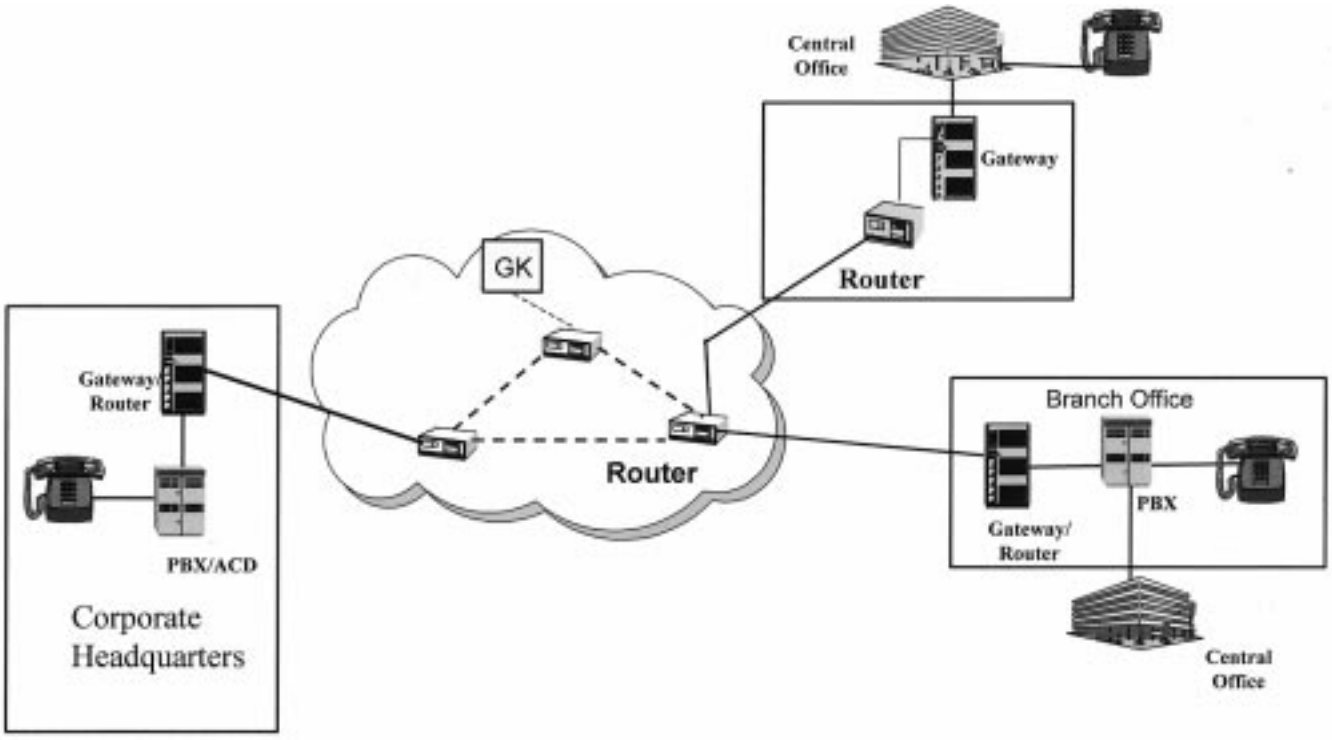

Fig. 1. Business use of VoIP.

PSTN	Public switched telephone network.
RAS	Registration, admission and status. RAS channels are used in H.323 gatekeeper communications.
RFC	Request for comment, an approved IETF document.
RSVP	ReSerVation setup protocol.
RSVP-TE	RSVP with traffic engineering extensions.
RTP	Real-time transport protocol.
RTCP	Real-time control protocol.
RTSP	Real-time streaming protocol.
QoS	Quality of service.
SDP	Session description protocol.
SG	Signaling gateway.
SIP	Session initiation protocol.
SS7	Signaling system 7.
SCTP	Stream control transmission protocol.
SOHO	Small office/ home office.
TCP	Transmission control protocol.
TLS	Transport layer security.
TDM	Time-division multiplexing.
TRIP	Telephony routing over IP.
URI	Uniform resource identifier.
URL	Uniform resource locator.
UDP	User datagram protocol.
VAD	Voice activity detection.
VoIP	Voice over Internet protocol.

I. Introduction

There is a plethora of published papers describing various ways in which voice and data communications networks may "converge" into a single global communications network. This paper deals with the technical aspects of implementing VoIP, without speculating on the timetable for convergence. A large number of factors are involved in making a high-quality VoIP call. These factors include the speech codec, packetization, packet loss, delay, delay variation, and the network architecture to provide QoS. Other factors involved in making a successful VoIP call include the call setup signaling protocol, call admission control, security concerns, and the ability to traverse NAT and firewall.

Although VoIP involves the transmission of digitized voice in packets, the telephone itself may be analog or digital. The voice may be digitized and encoded either before or concurrently with packetization. Fig. 1 shows a business in which a PBX is connected to VoIP gateway as well as to the local telephone company central office. The VoIP gateway allows telephone calls to be completed through the IP network. Local calls can still be completed through the telephone company as in the past. The business may use the IP network to make all calls between its VoIP gateway connected sites or it may choose to split the traffic between the IP network and the PSTN based on a least-cost routing algorithms configured in the PBX. VoIP calls are not restricted to telephones served directly by the IP network. We refer to VoIP calls to telephones served by the PSTN as "off-net" calls. Off-net calls may be routed over the IP network to a VoIP/PSTN gateway near the destination telephone.

An alternative VoIP implementation uses IP phones and does not rely on a standard PBX. Fig. 2 is a simplified diagram of an IP telephone system connected to a wide area IP network. IP phones are connected to a LAN. Voice calls can be made locally over the LAN. The IP phones include

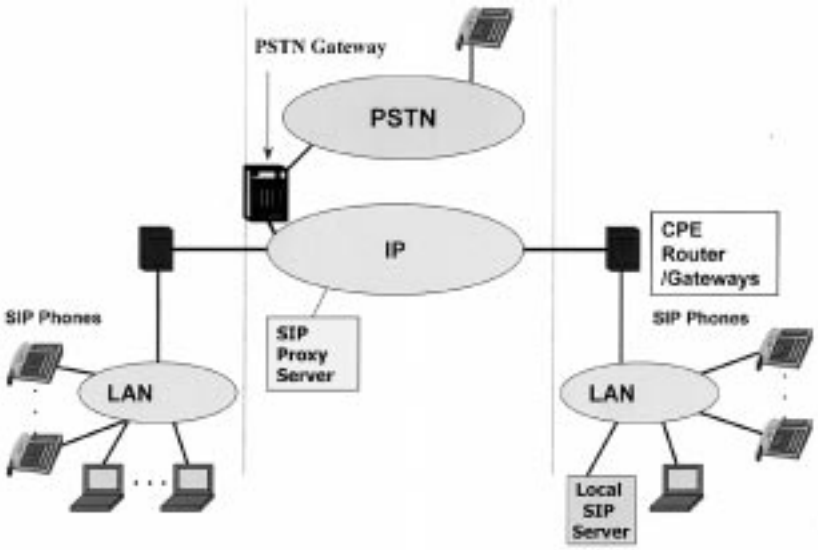

Fig. 2. VoIP from end to end.

Table 1
Characteristics of Several Voice Codecs

Codec	Algorithm	*Frame Size/ Lookahead*	Usual Rate	Comments
G.711	PCM	0.125 ms/0	64 Kb/s	Universal use
G.722		0.125 ms/1.5 ms	48, 56 or 64 Kb/s	Wideband coder
G.726	ADPCM	0.125 ms/0	32 Kb/s	High quality, low complexity
G.728	LD-CELP	0.625 ms/0	16 Kb/s	High quality in tandem; Recommended for cable
G.729(A)	CS-ACELP	10 ms/5 ms	8 Kb/s	Widespread use
G.729e	Hybrid CELP	10 ms/5 ms	11.8 Kb/s	High quality/complexity; Recommended for cable
G.723.1(6.3)	MPC-MLQ	30 ms/7.5 ms	6.3 Kb/s	Video conferencing origin
G.723.1(5.3)	ACELP	30 ms/7.5 ms	5.3 Kb/s	Video conferencing origin
IS-127	RCELP	20 ms/5ms	Var. 4.2 Kb/s avg.	
AMR	ACELP	20 ms	Var. 4.75-12.2 Kb	Compatible w. No. Amer. & Japanese digital cellular, WCDMA (not CDMA2000); Nokia IPR

codecs that digitize and encode (as well as decode) the speech. The IP phones also packetize and depacketize the encoded speech. Calls between different sites can be made over the wide area IP network. Proxy servers perform IP phone registration and coordinate call signaling, especially between sites. Connections to the PSTN can be made through VoIP gateways.

II. Voice Quality

Many factors determine voice quality, including the choice of codec, echo control, packet loss, delay, delay variation (jitter), and the design of the network. Packet loss causes voice clipping and skips. Some codec algorithms can correct for some lost voice packets. Typically, only a single packet can be lost during a short period for the codec correction algorithms to be effective. If the end-to-end delay becomes too long, the conversation begins to sound like two parties talking on a Citizens Band radio. A buffer in the receiving device always compensates for jitter (delay variation). If the delay variation exceeds the size of the jitter buffer, there will be buffer overruns at the receiving end, with the same effect as packet loss anywhere else in the transmission path.

For many years, the PSTN operated strictly with the ITU standard G.711. However, in a packet communications network, as well as in wireless mobile networks, other codecs will also be used. Telephones or gateways involved in setting up a call will be able to negotiate which codec to use from among a small working set of codecs that they support.

Codecs: There are many codecs available for digitizing speech. Table 1 gives some of the characteristics of a few standard codecs.[1]

[1]Note that the G.xxx codecs are defined by the ITU. IS-xxx codecs are defined by the TIA.

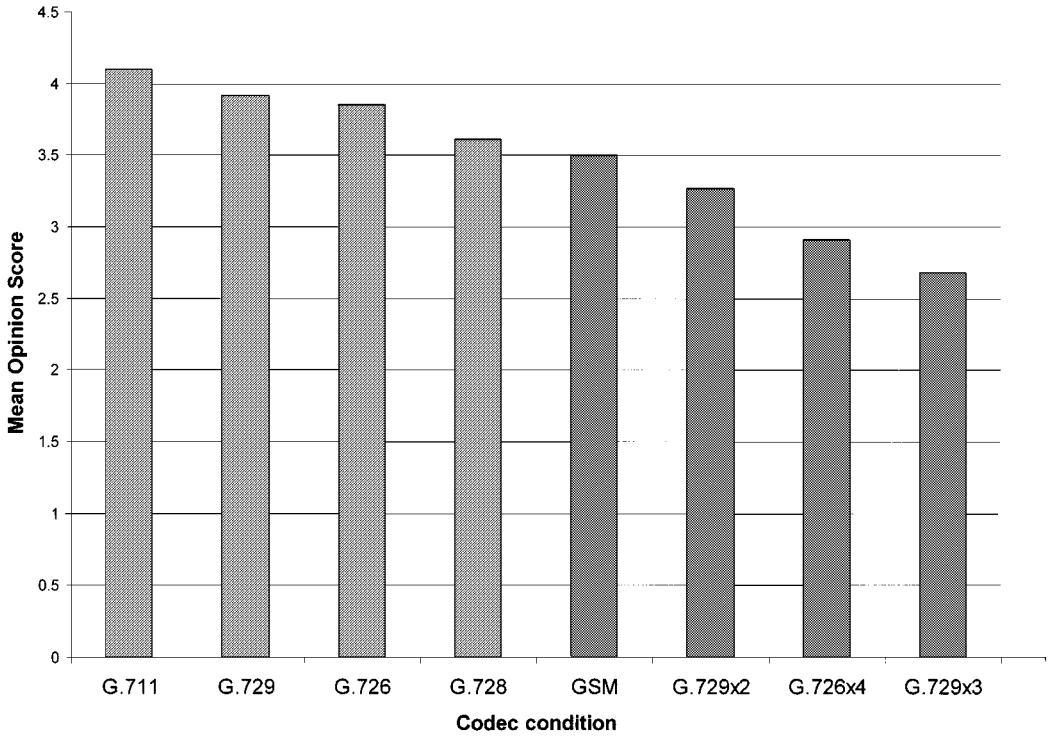

Fig. 3. Effect of codec concatenation on an MOS.

The quality of a voice call through a codec is often measured by subjective testing under controlled conditions using a large number of listeners to determine an MOS. Several characteristics can be measured by varying the test conditions. Important characteristics include the effect of environmental noise, the effect of channel degradation (such as packet loss), and the effect of tandem encoding/decoding when interworking with other wireless and terrestrial transport networks. The latter characteristic is especially important since VoIP networks will have to interwork with switched circuit networks and wireless networks using different codecs for many years. The general order of the fixed-rate codecs listed in the table, from best to worst performance in tandem, is G.711, G.726, G.729e, G.728, G.729, G.723.1. Quantitative results are given in [1]. Since voice quality suffers when placing low-bit-rate codecs in tandem in the transmission path, the network design should strive to avoid tandem codecs whenever and wherever possible.

Concatenation and Transcoding: The best packet network design codes the speech once near the speaker and decodes it once near the listener. Concatenation of low-bit-rate speech codecs, as well as the transcoding of speech in the middle of the transmission path, degrades speech quality. Fig. 3 shows the MOSs of several codecs with and without concatenation. (These results are from [1]. An MOS of 5 is excellent, 4 is good, 3 is fair, 2 is poor, and 1 is very bad. Note that G.729 $\times$ 2 means that speech coded with G.729 was decoded and then recoded with G.729 before reaching the final decoder. G.729 $\times$ 3 means that three G.729 codecs were concatenated in the speech path between the speaker and listener.) Fig. 4 shows the MOSs resulting from the interworking of different codecs, possibly in a transcoding situation.

III. Transport

Typical Internet applications use TCP/IP, whereas VoIP uses RTP/UDP/IP. Although IP is a connectionless best effort network communications protocol, TCP is a reliable transport protocol that uses acknowledgments and retransmission to ensure packet receipt. Used together, TCP/IP is a reliable connection-oriented network communications protocol suite. TCP has a rate adjustment feature that increases the transmission rate when the network is uncongested, but quickly reduces the transmission rate when the originating host does not receive positive acknowledgments from the destination host. TCP/IP is not suitable for real-time communications, such as speech transmission, because the acknowledgment/retransmission feature would lead to excessive delays. UDP provides unreliable connectionless delivery service using IP to transport messages between end points in an internet. RTP, used in conjunction with UDP, provides end-to-end network transport functions for applications transmitting real-time data, such as audio and video, over unicast and multicast network services.[2] RTP does not reserve resources and does not guarantee quality of service. A companion protocol RTCP does allow monitoring of a link, but most VoIP applications offer a continuous stream of RTP/UDP/IP packets without regard to packet loss or delay in reaching the receiver.

Although transmission may be inexpensive on major routes, in some parts of the world as well as in many private networks, transmission facilities are expensive enough to

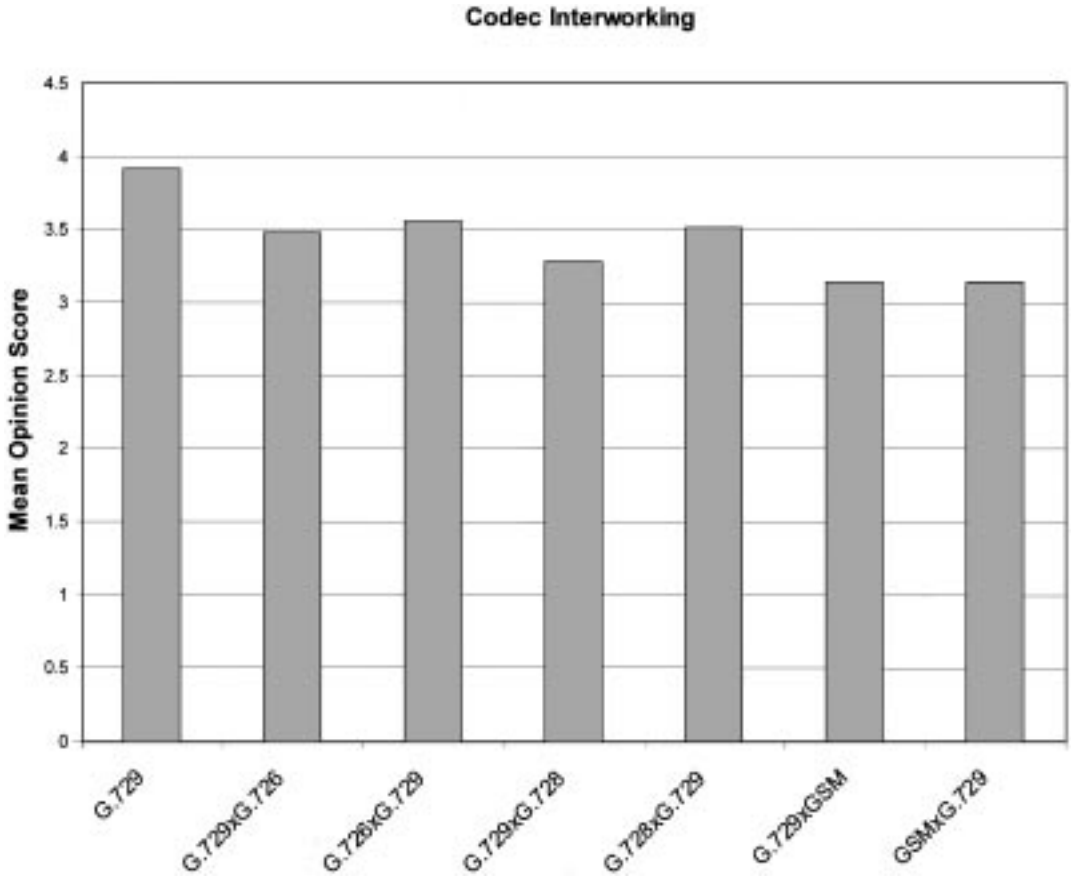

Fig. 4. Effects of transcoding.

merit an effort to use bandwidth efficiently. This effort starts with the use of speech compression codecs. Use of low bandwidth leads to a long packetization delay and the most complex codecs. An engineering tradeoff must be made to achieve an acceptable packetization delay, an acceptable level of codec complexity, and an acceptable call transmission capacity requirement. Another technique for increasing bandwidth efficiency is voice activity detection and silence suppression. Voice quality can be maintained while using silence suppression if the receiving codec inserts a carefully designed comfort noise during each silence period. For example, Annex B of ITU-T Recommendation G.729 defines a robust voice activity detector that measures the changes over time of the background noise and sends, at a low rate, enough information to the receiver to generate comfort noise that has the perceptual characteristics of the background noise at the sending telephone [3].

Coding and packetization result in delays greater than users typically experience in terrestrial switched circuit networks. As we have seen, standard speech codecs are available for output coding rates in the approximate range of 64 to 5 kb/s. Generally, the lower the output rate, the more complex the codec. Packet design involves a tradeoff between payload efficiency (payload/total packet size) and packetization delay (the time required to fill the packet). For IPv4, the RTP/UDP/IP header is 40 bytes. A payload of 40 bytes would mean 50% payload efficiency. At 64 kb/s, it only takes 5 ms to accumulate 40 bytes, but at 8 kb/s it takes 40 ms to accumulate 40 bytes. A packetization delay of 40 ms is significant, and many VoIP systems use 20-ms packets despite the low payload efficiency when using low-bit-rate codecs. For continuous speech, the call transmission capacity requirement BW (in kb/s) is related to the header size H (in bits), the codec output rate R (in kb/s) and the payload sample size S (in milliseconds) as

$$BW = R + \frac{H}{S}.$$

Fig. 5 shows a plot of BW versus R and S assuming $H = 320$ b.

There are several header compression algorithms that will improve payload efficiency [4]–[6]. The 40-byte RTP/UDP/IP header can be compressed to 2–7 bytes. A typical compressed header is four bytes, including a two-byte checksum. In an IP network, header compression must be done on a link-by-link basis, because the header must be restored before a router can choose an outgoing interface. Therefore, this technique is most suitable for low-speed access links. Fig. 6 shows a plot of BW versus R and S assuming $H = 32$ b.

The lowest *BW* requirements lead to a long packetization delay and the most complex codecs. An engineering tradeoff must be made to achieve an acceptable packetization delay, an acceptable codec complexity, and an acceptable call bandwidth requirement. The following sections discuss quality and bandwidth efficiency in more detail.

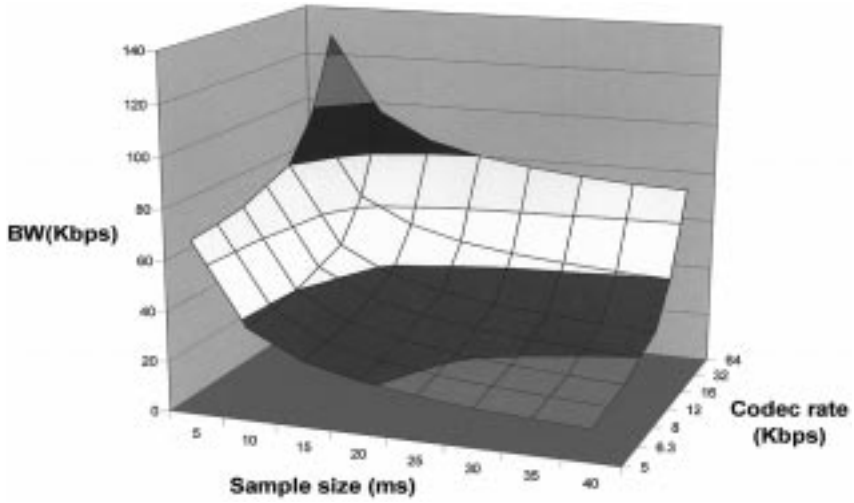

Fig. 5. The varying bands, from top to bottom, represent the following VoIP bandwidth requirements (40-byte headers): 120–140, 100–120, 80–100, 60–80, 40–60, 20–40, and 0–20.

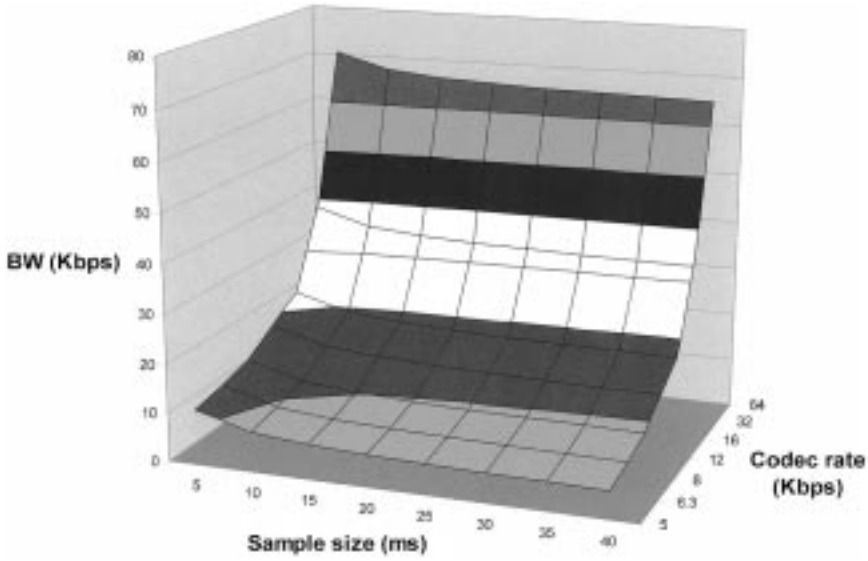

Fig. 6. From top to bottom, varying bands represent the following VoIP bandwidth requirements (4-byte headers): 70–80, 60–70, 50–60, 40–50, 30–40, 20–30, 10–20, 0–10.

A. Delay

Transmission time includes delay due to codec processing as well as propagation delay. ITU-T Recommendation G.114 [8] recommends the following one-way transmission time limits for connections with adequately controlled echo (complying with G.131 [7]):

- 0 to 150 ms: acceptable for most user applications;
- 150 to 400 ms: acceptable for international connections;
- $>$ 400 ms: unacceptable for general network planning purposes; however, it is recognized that in some exceptional cases this limit will be exceeded.

ITU-T Recommendation G.114 Annex B describes the results of subjective tests to evaluate the effects of pure delay on speech quality. A test completed in 1989 showed the percent of users rating the call as poor or worse (POW) for overall quality started increasing above 10% only for delays greater than 500 ms, but POW for interruptability was above 10% for delays of 400 ms. One of the tests, completed in 1990, "was designed to obtain subjective reactions, in context of interruptability and quality, to echo-free telephone circuits in which various amounts of delay were introduced. The results indicated that long delays did not greatly reduce mean opinion scores over the range of delay tested, viz. 1 to 1000 ms of one-way delay… However, observations during the test and subject interviews after the test showed the subjects experienced some real difficulties in communicating at the longer delays, although subjects did not always associate the difficulty with the delay "[8].

A Japanese study in 1991 measured the effect of delay using six different tasks involving more or less interruptions in the dialogue. The delay detectability threshold was defined as the delay detected by 50% of a task's subjects. As the interactivity required by the tasks decreased, the delay detectability threshold increased from 45 to 370 ms of one-way

Table 2
Delay Budget for VoIP Using G.729 Codec

Delay Source (G.729)	On-net Budget (ms)
Device Sample Capture	0.1
Encoding Delay (Algorithmic Delay + Processing Delay)	17.5
Packetization/ Depacketization Delay	20
Move to Output Queue/Queue Delay	0.5
Access (up) Link Transmission Delay	10
Backbone Network Transmission Delay	Dnw
Access (down) Link Transmission Delay	10
Input Queue to Application	0.5
Jitter Buffer	60
Decoder Processing Delay	2
Device Playout Delay	0.5
Total	121.1 + Dnw

delay. As the one-way delay increased from 100 to 350 ms, the MOS connection quality decreased from 3.74 ($\pm$0.52) to 3.48 ($\pm$0.48), and the connection acceptability decreased from 80% to 73% [8].

Delay variation, sometimes called jitter, is also important. The receiving gateway or telephone must compensate for delay variation with a jitter buffer, which imposes a delay on early packets and passes late packets with less delay so that the decoded voice streams out of the receiver at a steady rate. Any packets that arrive later than the length of the jitter buffer are discarded. Since we want low packet loss, the jitter buffer delay is the maximum delay variation that we expect. This jitter buffer delay must be included in the total end-to-end delay that the listener experiences during a conversation using packet telephony.

B. Delay Budget

Packetized voice has larger end-to-end delays than a TDM system, making the above delay objectives challenging. A sample on-net delay budget for the G.729 (8 kb/s) codec is shown in Table 2.

This budget is not precise. The allocated jitter buffer delay of 60 ms is only an estimate; the actual delay could be larger or smaller.[2] Since the sample budget does not include any specific delays for header compression and decompression, we may consider that, if those functions are employed, the associated processing delay is lumped into the access link delay.

This delay budget allows us to stay within the G.114 guidelines, leaving 29 ms for the one-way backbone network delay (Dnw) in a national network. This is achievable in small countries. Network delays in the Asia Pacific region, as well as between North America and Asia, may be higher than 100 ms. According to G.114, these delays are acceptable for international links. However, the end-to-end delays for VoIP calls are considerably larger than for PSTN calls.

[2]In the absence of Network QoS, the jitter buffer delay could be larger. With QoS and an adaptive jitter buffer, the delay could adapt down to a lower value during a long conversation.

IV. Network QoS

There are various approaches to providing QoS in IP networks. Before discussing the QoS options, one must consider whether QoS is really necessary. Some Internet engineers assert that the way to provide good IP network performance is through provisioning, rather than through complicated QoS protocols. If no link in an IP network is ever more than 30% occupied, even in peak traffic conditions, then the packets should flow through without any queue delays, and elaborate protocols to give priority to one class of packet are not necessary. The design engineer should consider the capacity of the router components to forward small voice packets as well as the bandwidth of the inter-router links in determining the occupancy of the network. If the occupancy is low, then performance should be good. Essentially, the debate is over whether excess network capacity (including link bandwidth and routers) is less expensive than QoS implementation.

The development of QoS features has continued because of the perception of some network engineers that real-time traffic (as well as other applications) may sometimes require priority treatment to achieve good performance. In some parts of the world, bandwidth is at least an order of magnitude more expensive than it is in the United States. In some cases, access links may be expensive and broadband access difficult to obtain, so that QoS may be desirable on the access links even if the core network is lightly loaded. Wireless access links are especially expensive, so QoS is important for wireless mobile IP phone calls.

QoS can be achieved by managing router queues and by routing traffic around congested parts of the network. Two key QoS concepts are the IntServ [9] and DiffServ. The IntServ concept is to reserve resources for each flow through the network. RSVP [10] was originally designed to be the reservation protocol. When an application requests a specific QoS for its data stream, RSVP can be used to deliver the request to each router along the path and to maintain router state to provide the requested service. RSVP transmits two types of Flow Specs conforming to IntServ rules. The traffic specification (Tspec) describes the flow, and the service request specification (Rspec) describes the

service requested under the assumption that the flow adheres to the Tspec. Current implementations of IntServ allow a choice of Guaranteed Service or Controlled-Load Service.

Guaranteed Service [11] involves traffic policing by a leaky token bucket model to control average traffic. Peak traffic is limited by a peak rate parameter p and an interval T so that no more than $p*T$ bytes are transmitted in any interval T. The packet size is restricted to be in the range $[m, M]$, so that smaller packets are considered to be of size m and packets larger than M are in violation of the contract. A bandwidth requirement is stated, and enough bandwidth is reserved on each hop to satisfy all the requirements of the flow. (The bandwidth requirement may not be the same on each hop [12].) If each node and hop can accept the service request, the flow should be lossless because the queue size reserved for the flow can be set to the length parameter of the token bucket. This service is designed for interactive real-time applications. To use it effectively, one needs a strict and realistic end-to-end delay budget in addition to bandwidth requirements of the flow.

Controlled-Load Service uses the same Tspec as Guaranteed Service. However, an Rspec is not defined. Flows using this service should experience the same performance as they would in a lightly loaded "best-effort" network. Controlled-Load Service would be appropriate for call admission control and would prevent the delays and packet losses that make real-time traffic suffer when the network is congested.

There are several reasons for not using IntServ with RSVP for IP telephony. Although IntServ with RSVP would work on a private network for small amounts of traffic, the large number of voice calls that IP telephony service providers carry on their networks would stress an IntServ RSVP system. First, the bandwidth required for voice itself is small, and the RSVP control traffic would be a significant part of the overall traffic. Second, RSVP router code was not designed to support many thousands of simultaneous connections per router.

It should be noted, however, that RSVP is a signaling protocol, and it has been proposed for use in contexts other than IntServ. For example, RSVP-TE is a constraint-based routing protocol for establishing LSPs with associated bandwidth and specified paths in an MPLS network [13]. RSVP has also been proposed as the call admission control mechanism for VoIP in differentiated services networks.

A. Differentiated Services

Since IntServ with RSVP does not scale well to support many thousands of simultaneous connections, the IETF has developed a simpler framework and architecture to support DiffServ [14]. The architecture achieves scalability by aggregating traffic into classifications that are conveyed by means of IP-layer packet marking using the DS field in IPv4 or IPv6 headers. Sophisticated classification, marking, policing, and shaping operations need only be implemented at network boundaries. Service provisioning policies allocate network resources to traffic streams by marking and conditioning packets as they enter a differentiated services-capable network, in which the packets receive a particular PHB based on the value of the DS field.

The primary goal of differentiated services is to allow different levels of service to be provided for traffic streams on a common network infrastructure. A variety of resource management techniques may be used to achieve this, but the end result will be that some packets will receive different (e.g., better) service than others. This will, for example, allow service providers to offer a real-time service giving priority to the use of bandwidth and router queues, up to the configured amount of capacity allocated to real-time traffic.

Despite the term "differentiated services," the IETF DiffServ working group undertook to define standards that have more generality than specific services. The reason is that if the IETF were to define new standard services, everyone would have to agree on what constitutes a useful service and every router would have to implement the mechanisms to support it. To deploy that new service, you would have to upgrade the entire Internet. Since a router has only a few functions, it makes more sense to standardize forwarding behavior ("send this packet first" or "drop this packet last"). So the DiffServ working group first defined PHBs, which could be combined with rules to create services.[3]

An important requirement is scalability, since the IETF intended differentiated services to be deployed in very large networks. To achieve scalability, the DiffServ architecture prescribes treatment for aggregated traffic rather than microflows and forces much of the complexity out of the core of the network into the edge devices, which process lower volumes of traffic and lesser numbers of flows.

The DiffServ architecture is based on a simple model where packets entering a network are classified and possibly conditioned at the boundaries of the network, and then assigned to different behavior aggregates. Each behavior is identified by a single DS codepoint. Within the core of the network, packets are forwarded according to the PHB associated with the DS codepoint.

One candidate PHB for voice service is EF. The objective of the EF PHB is to build a low-loss, low-latency, low-jitter, assured bandwidth, end-to-end service through DS domains. Such a service would appear to endpoints like a point-to-point connection or "virtual leased line." Since router queues cause traffic to experience loss, jitter, and excessive latency, EF PHB tries to ensure that all EF traffic experiences either no or very small queues. Since queues arise when the short-term traffic arrival rate exceeds the departure rate at some node, this ensures that, at every node, the aggregate EF traffic maximum arrival rate is less than the EF minimum departure rate [15]–[17]. The original idea was to ensure low delay and no packet loss. Subsequent analysis has shown that, under the no loss hypothesis, evaluating the worst-case arrival patterns on each node leads to poor delay bounds after just a few hops. Using a worst-case analysis to determine admission criteria would lead to unacceptably low utilization.

[3]Recently, the IETF DiffServ Working Group has started considering per domain behaviors, but as of this writing the work is still in progress.

However, simulations and early EF trials show that good performance can be achieved with reasonable efficiency [18].

The appeal of DiffServ is that it is relatively simple (compared to IntServ), yet provides applications like VoIP some improvement in performance compared to "best-effort" IP networks. However, DiffServ relies on ample network capacity for EF traffic and makes use of standard routing protocols that make no attempt to use the network efficiently. Confronted with network congestion, EF would drop packets at the edge instead of queuing or rerouting them. DiffServ has no topology-aware admission control mechanism. The IETF DiffServ Working Group has not recommended a mechanism for rejecting additional VoIP calls if accepting them would degrade the quality of calls in progress.[4]

B. MPLS-Based QoS

For several decades, traffic engineering and automated rerouting of telephone traffic have increased the efficiency and reliability of the PSTN. Frame relay and ATM also offer source (or "explicit") routing capabilities that enable traffic engineering. However, IP networks have relied on destination-based routing protocols that send all the packets over the shortest path, without regard to the utilization of the links comprising that path. In some cases, links can be congested by traffic that could be carried on other paths comprised of underutilized links. It is possible to design an IP network to run on top of a frame relay or ATM ("Layer 2") network, providing some traffic engineering features, but this approach adds cost and operational complexity.

MPLS offers IP networks the capability to provide traffic engineering as well as a differentiated services approach to voice quality. MPLS separates routing from forwarding, using label swapping as the forwarding mechanism. The physical manifestation of MPLS is the LSR. LSRs perform the routing function in advance by creating LSPs connecting edge routers. The edge router (an LSR) attaches short (four-byte) labels to packets. Each LSR along the LSP swaps the label and passes it along to the next LSR. The last LSR on the LSP removes the label and treats the packet as a normal IP packet.

MPLS LSPs can be established using LDP [19], RSVP-TE [20], or CR-LDP [21]. When using LDP, LSPs have no associated bandwidth. However, when using RSVP-TE or CR-LDP, each LSP can be assigned a bandwidth, and the path can be designated for traffic engineering purposes. MPLS traffic engineering (MPLS-TE) combines extensions to OSPF or IS-IS, to distribute link resource constraints, with the label distribution protocols RSVP-TE or CR-LDP. Resource and policy attributes are configured on every link and define the capabilities of the network in terms of bandwidth, a Resource Class Affinity string, and a traffic engineering link metric. When performing the constraint-based path computation, the originating LSR compares the link attributes received via OSPF or IS-IS to those configured on the LSP.

[4]Indeed, the working group co-chairs probably did not believe that admission control was within their charter.

Differentiated services can be combined with MPLS to map DiffServ Behavior Aggregates onto LSPs [22]. QoS policies can be designated for particular paths. More specifically, the EXP field of the MPLS label can be set so that each label switch/router in the path knows to give the voice packets highest priority, up to the configured maximum bandwidth for voice on a particular link. When the high-priority bandwidth is not needed for voice, it can be used for lower priority classes of traffic.

DiffServ and MPLS DiffServ are implemented independently of the routing computation. MPLS-TE computes routes for aggregates across all classes and performs admission control over the entire LSP bandwidth. MPLS-TE and MPLS DiffServ can be used at the same time. Alternatively, DiffServ can be combined with traffic engineering to establish separate tunnels for different classes. DS-TE makes MPLS-TE aware of DiffServ, so that one can establish separate LSPs for different classes, taking into account the bandwidth available to each class. So, for example, a separate LSP could be established for voice, and that LSP could be given higher priority than other LSPs, but the amount of voice traffic on a link could be limited to a certain percentage of the total link bandwidth. This capability is currently being standardized by the IETF Traffic Engineering Working Group [23], [24].

Voice DS-TE tunnels can be based on a delay metric or a bandwidth metric. Combining DS-TE with DiffServ over MPLS allows QoS for VoIP with the capability of fast reroute if a link or node failure occurs. DiffServ can guarantee that a specified amount of voice bandwidth is available on each link in a network. DS-TE routing and admission control can create a guaranteed bandwidth tunnel that has the required bandwidth in the highest priority queue on every link. Service conditioning at the edge can ensure that the aggregate VoIP traffic directed onto the guaranteed bandwidth tunnel is less than the capacity of the tunnel. This allows a tight SLA with admission control without overprovisioning the network.

A VoIP network designer can choose DiffServ, MPLS-TE plus DiffServ, or DS-TE according to the economics of the situation. If VoIP is to be a small portion of the total traffic, DiffServ or MPLS-TE plus DiffServ may be sufficient. DS-TE promises more efficient use of an IP network carrying a large proportion of VoIP traffic, with perhaps more operational complexity.

V. Call Signaling

There are several VoIP call signaling protocols. We shall discuss and compare the characteristics of the H.323 protocol suite, SIP, MGCP, and Megaco/H.248. H.323 and SIP are peer-to-peer control-signaling protocols, while MGCP and Megaco are master–slave control-signaling protocols. MGCP is based on the PSTN model of telephony. H.323 and Megaco are designed to accommodate video conferencing as well as basic telephony, but they are still based on a connection-oriented paradigm, despite their use for packet communications systems. H.323 gateways have more call control function than the media gateways using MGCP, which

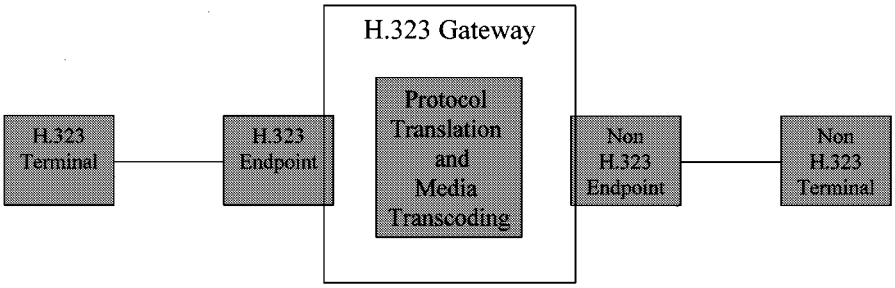

Fig. 7. H.323 gateway.

assumes that more of the intelligence resides in a separate media gateway controller. SIP was designed from scratch for IP networks, and accommodates intelligent terminals engaged in not only voice sessions, but other applications as well.

A. H.323

The ITU-T Recommendation H.323 protocol suite has evolved out of a video telephony standard [25]. When early (ca. 1996) IP telephony pioneers developed proprietary products, there was an industry call to develop a VoIP call control standard quickly so that users and service providers would be able to have a choice of vendors and products that would interoperate. The Voice-over-IP Activity Group of the International Multimedia Telecommunications Consortium (IMTC) recommended H.323, which had been developed for multimedia communications over packet data networks. These packet networks might include LANs or WANs. The IMTC held the view that VoIP was a special case of IP Video Telephony. Although not all VoIP pioneers agreed that video telephony would quickly become popular, the H.323 protocol suite became the early leading standard for VoIP implementations. Versions 2–4 include modifications to make H.323 more amenable to VoIP needs.

H.323 entities may be integrated into personal computers or routers or implemented in stand-alone devices. For VoIP, the important H.323 entities are terminals, gateways, and gatekeepers. An H.323 gateway provides protocol translation and media transcoding between an H.323 endpoint and a non-H.323 endpoint (see Fig. 7). For example, a VoIP gateway provides translation of transmission formats and signaling procedures between a telephone switched circuit network (SCN) and a packet network. In addition, the VoIP gateway may perform speech transcoding and compression, and it is usually capable of generating and detecting DTMF signals.

The H.323 VoIP terminal elements include the following.

- A System Control Unit provides signaling for proper operation of the H.323 terminal that provides for call control using H.225.0 and H.245 (as described below).
- H.225.0 layer formats the transmitted audio and control streams into messages, retrieves the audio streams from messages that have been received from the network interface, and performs logical framing, sequence numbering, error detection and error correction as appropriate.
- An audio codec transcodes and may also compress speech.

H.323 Gatekeeper Characteristics: H.323 gatekeepers perform admission control and address translation functions. Several gatekeepers may communicate with each other to coordinate their control services. Networks with VoIP gateways should (but are not required to) have gatekeepers to translate incoming E.164 addresses into Transport Addresses (e.g., IP address and port number). The gatekeeper is logically separate from the other H.323 entities, but physically it may coexist with a terminal, gateway, or an H.323 proxy. When present in a VoIP network, the gatekeeper provides the following functions.

- Address translation—the gatekeeper translates alias addresses (e.g., E.164 telephone numbers) to Transport Addresses, using a translation table that is updated using Registration messages and other means.
- Admissions control—the gatekeeper authorizes network access using H.225 messages. Admissions criteria may include call authorization, bandwidth, or other policies.
- Bandwidth control—the gatekeeper controls how much bandwidth a terminal may use
- Zone management—a terminal may register with only one gatekeeper at a time. The gatekeeper provides the above functions for terminals and gateways that have registered with it.
- Participation in call control signaling is optional.
- Directory services are optional.

Registration, Admissions, and Status Channel: The RAS channel carries messages used in gatekeeper endpoint registration processes that associate an endpoint's alias (e.g., E.164 telephone number) with its TCP/IP address and port number to be used for call signaling. The RAS channel is also used for transmission of admission, bandwidth change, status, and disengage messages between an endpoint and its gatekeeper. H.225.0 recommends time outs and retry counts for RAS messages, since they are transmitted on an unreliable UDP channel.

Call Signaling Channel: The Call Signaling Channel carries H.225.0 call control messages using TCP, making it a reliable channel. H.323 endpoints and gatekeepers use Q.931 messages (with TCP) for call signaling. In networks with no gatekeeper, endpoints send call signaling messages directly to the called endpoint using the Call Signaling Transport Addresses. If the network has a gatekeeper, the calling end-

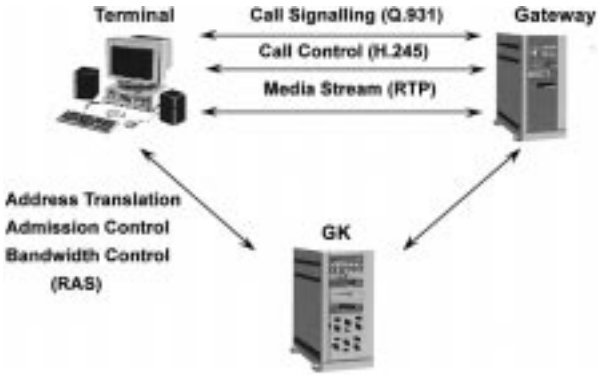

Fig. 8. Direct endpoint call signaling.

point sends the initial admission message to the gatekeeper using the gatekeeper's RAS Channel Transport Address. In the initial exchange of admissions messages, the gatekeeper tells the originating endpoint whether to send the call signaling messages directly to the other endpoint or to route them through the gatekeeper

Call signaling may be routed in two ways. Fig. 8 shows direct endpoint call signaling, which sends call signaling messages directly between the endpoints or gateways

Figs. 9 and 10 show gatekeeper routed call signaling, which routes call-signaling messages from one endpoint through the gatekeeper to the other endpoint.

In direct endpoint call signaling, the gatekeeper participates in call admission but has little direct knowledge of connections. Due to its limited involvement, a single gatekeeper can process a large number of calls, but the gatekeeper has a limited ability to perform service management functions. The gatekeeper cannot determine call completion rates, and, if it is to perform call detail recording, it must depend on the endpoints for call duration information.

The gatekeeper routed call signaling method results in more load on the gatekeeper, since it must process the Q.931 messages. The gatekeeper may close the call signaling channel after call setup is completed. However, if the gatekeeper remains involved in the call, e.g., to produce call records or to support supplementary services, it will keep the channel open for the duration of the call.[5]

H.245 Control Function: The H.245 Control Channel carries end-to-end H.245 control messages governing operation of the H.323 entities (H.323 host, H.323 gateway or H.323 gatekeeper). The key function of the H.245 Control Channel is capabilities exchange. Other H.245 functions include opening and closing of logical channels, flow control messages, mode preference requests, and general commands

[5]Both H.225 and H.245 use TCP to establish a reliable transport connection between endpoints, gateways, and gatekeepers. In the case of gatekeeper-routed call signaling, the TCP connections are kept up for the duration of the call. Although normally reliable, the failure of a TCP connection could result in mid-call termination even though the TCP connection was not in use at the time. For example, suppose gatekeeper routed call signaling is used, and the TCP connection from gateway to gatekeeper is broken due to a timeout or a failure to exchange keepalive messages during a link failure or rerouting. Calls may be dropped even though the RTP voice media streams may have been unaffected by the network event that caused the TCP connection to the gatekeeper to fail.

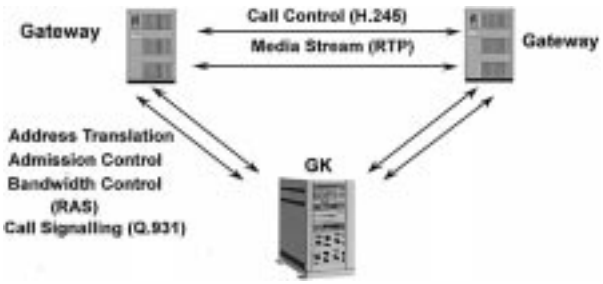

Fig. 9. Gatekeeper routed call signaling (Q.931).

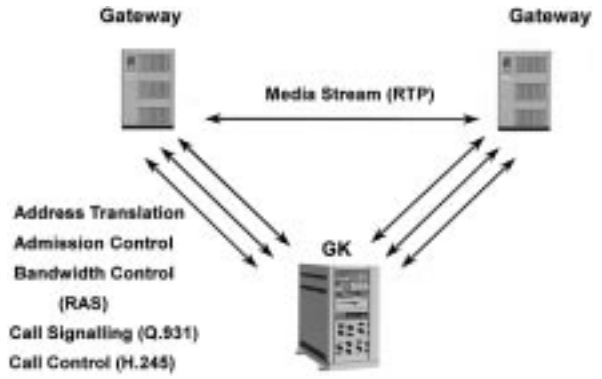

Fig. 10. Gatekeeper routed call signaling (Q.931/H.245).

and indications. The endpoint establishes an H.245 Control Channel for each call in which the endpoint participates. This logical H.323 Control Channel is open for the entire duration of the call. To conform to Recommendation H.245, H.323 endpoints must support the syntax, semantics, and procedures of the following protocol entities:

- master/slave determination;
- capability exchange;
- logical channel signaling;
- bidirectional logical channel signaling;
- close logical channel signaling;
- mode request;
- round-trip delay determination;
- maintenance loop signaling.

As an example of how H.245 is used, let us discuss how it accommodates simple telephony signaling.

DTMF Relay and Hook-Flash Relay: Short DTMF tones transmitted by low-bit-rate codecs (e.g., G.729 and G.723.1) may be distorted to the extent that the user may have trouble accessing automated DTMF-based systems such as voice mail, menu-based ACD systems, automated banking systems, etc. H.323v2 offers a remedy by sending the DTMF tones "out of band" instead of being compressed the same as speech. This is called DTMF relay. If DTMF relay is enabled, an H.323 gateway detects DTMF signals, cancels the DTMF from the voice stream before it is sent over RTP, and sends an H.245 User Input Indication providing the value of the DTMF digit (0–9, A-D, * or #) and an estimate of the duration of the tone to the remote endpoint. The gateway will only send DTMF signals using H.245 if the H.245 capability exchange procedure results in the knowledge that the remote endpoint is capable of

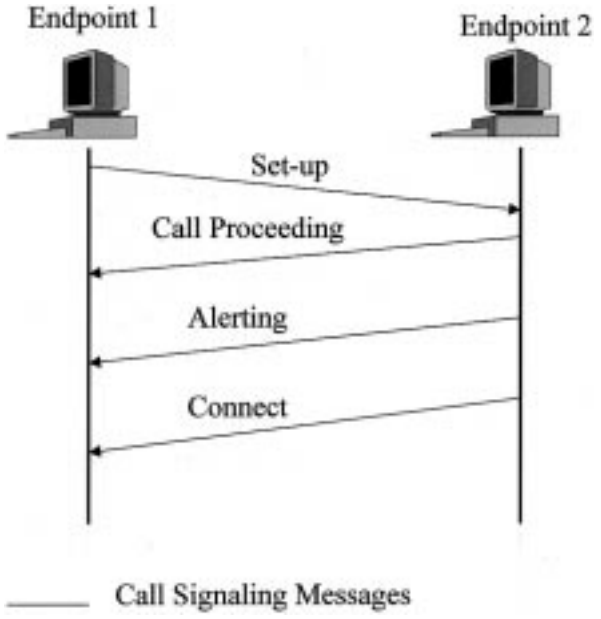

Fig. 11. Basic call setup with no gatekeeper.

receiving DTMF signals in the user input indication. The H.245 standard specifies two indications for conveying DTMF input in the user input indication: the *alphanumeric* indication and the *signal* indication. H.323v2 adds support for both these methods. The *signal* indication includes the digit duration and optional RTP information such as a time stamp that may be used by a receiver for synchronizing the DTMF signal with the RTP stream.

H.323v2 also supports the relay of hookflash indications by using H.245 user input messages from gateway telephony interfaces to gateway packet interfaces. When a gateway receives a hookflash indication in a signal user input indication and the telephony interface is FXO,[6] the Gateway generates a hookflash on the FXO interface.

Since hookflash duration varies among analog telephone vendors, gateways must be configured to compensate for this variance and avoid hookflash bounce. The receiving Gateway should use the configured default hookflash duration on its telephony interface. If a "duration" is specified in the hookflash indication received by H.245, Recommendation H.245 requires that it be ignored.

Call Setup: Fig. 11 diagrams basic call setup signaling for the case where neither endpoint is registered with a gatekeeper. The calling endpoint (endpoint 1) sends the setup (1) message to the well-known call signaling channel TSAP identifier (TCP port #1720) of endpoint 2. Endpoint 2 responds with call proceeding (2), alerting (3), and finally the connect (4) message containing an H.245 control channel transport address for use in H.245 signaling.

Fig. 12 diagrams a basic setup with gatekeeper routed call signaling. First, the originating gateway sends an admission request (ARQ) to the gatekeeper, which responds with an admission confirmation (ACF). Then setup proceeds as indicated.

[6]An FXO interface is used to connect to a PSTN central office and is the interface offered on a standard telephone.

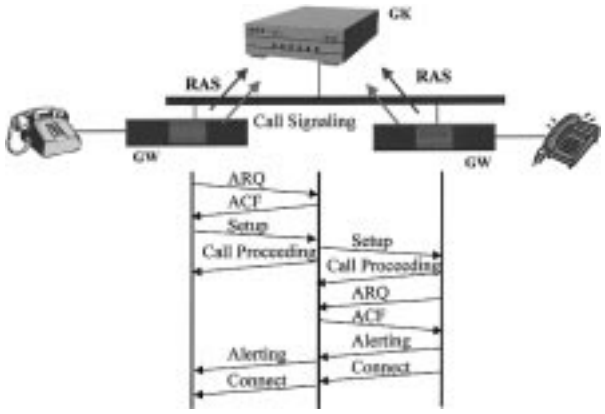

Fig. 12. Basic call setup with gatekeeper routed call signaling.

Fig. 13 diagrams call setup where both endpoints are registered with separate gatekeepers, and both use gatekeeper routed call signaling. Note that these diagrams do not show explicitly the establishment of TCP connections between the endpoints and the gatekeepers. The first part of the call setup is similar to the single gatekeeper case shown in Fig. 12. When the call setup message reaches endpoint 2, it initiates an ARQ(6) /ACF(7) exchange with gatekeeper 2. Assuming the call is acceptable, gatekeeper 2 sends its own call signaling address in a ARJ(7) reject message (instead of ACF) with a cause code commanding the endpoint to route the call signaling to it. The rest of the diagram is self-explanatory.

As one can see from Fig. 13, call signaling can involve many messages passing back and forth among the H.323 entities. To reduce the call setup time for straightforward calls such as VoIP, H.323v2 introduced an alternate call setup procedure called "Fast Connect." H.323 endpoints may use either Fast Connect or H.245 procedures to establish media channels in a call.

Fast Connect Procedure: Fast Connect shortens basic point-to-point call setup time by reducing the number of messages exchanged. After one round-trip message exchange, endpoints can start a conversation. This is accomplished by including a fastStart element in the SETUP message. The fastStart element describes a sequence, in preference order, of media channels that the calling endpoint proposes to use, including all of the parameters necessary to open and begin transferring media on the channels immediately. The called endpoint can agree to use the Fast Connect procedure by sending a Q.931 message containing a fastStart element selecting from amongst the OpenLogicalChannel proposals that the calling endpoint offered. Channels accepted in this way are considered opened as if the usual H.245 procedures had been followed. The called endpoint may begin transmitting media immediately after sending a Q.931 message with the fastStart acceptance of the call, and the calling endpoint may begin transmitting media as soon as it receives that message.

Security: The H.235 standard addresses security issues, including authentication, integrity, privacy, and nonrepudiation. The authentication function makes sure that the endpoints participating in the conference are really who they say

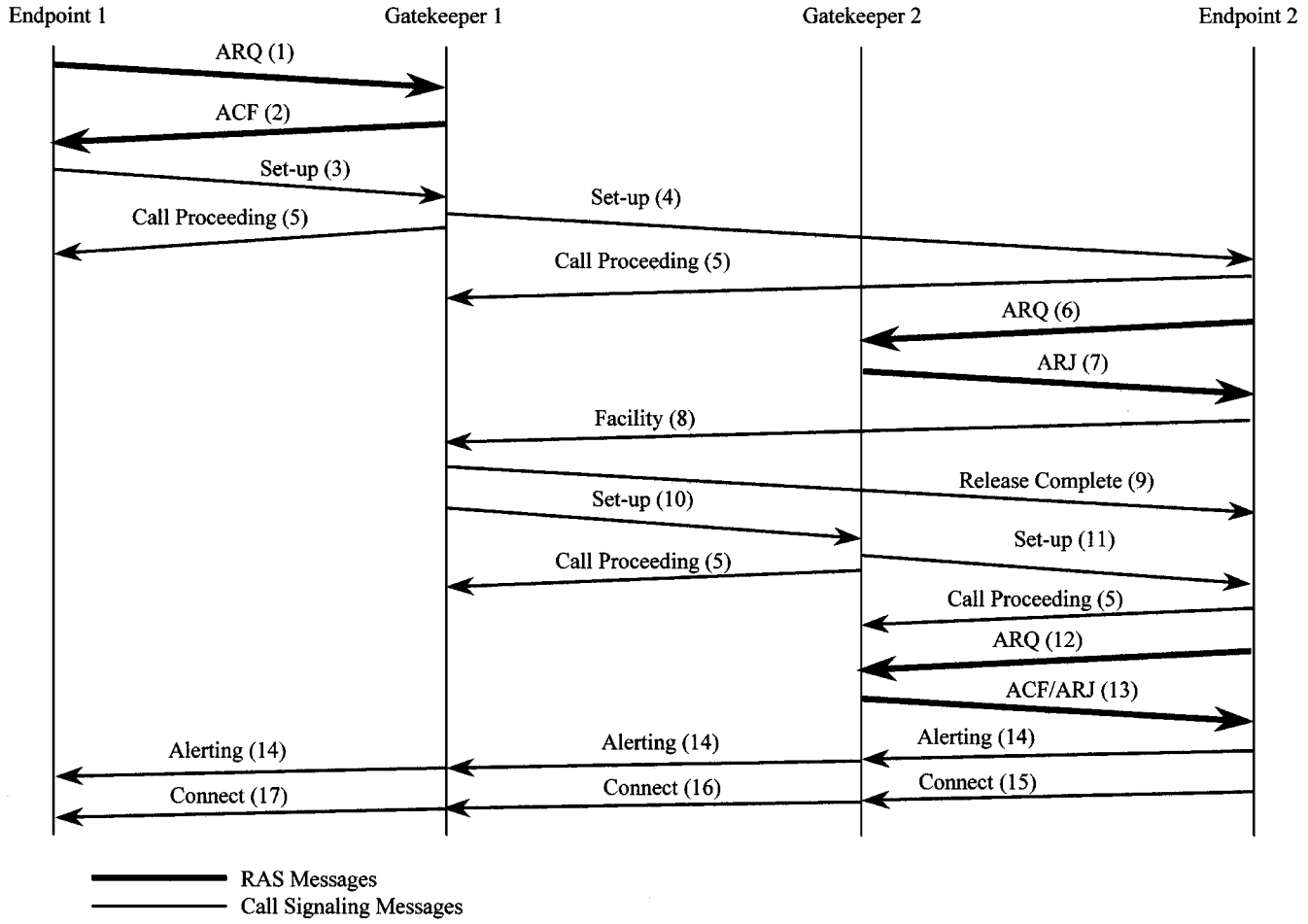

Fig. 13. Gatekeeper routed call signaling involving two gatekeepers.

they are. The integrity function provides a means to validate that the data within a packet is indeed an unchanged representation of the data. Privacy is provided by encryption and decryption mechanisms that hide the data from eavesdroppers so that if it is intercepted it cannot be heard. Nonrepudiation is a means of protection against someone falsely denying that they participated in a conference. H.323v2 specifies hooks for each of these security features. H.235 specifies the proper usage of these hooks.

The RAS channel used for gateway-to-gatekeeper signaling is not a secure channel. To ensure secure communication, H.235 allows gateways to include an authentication key in their RAS messages. The gatekeeper can use this authentication key (password with hashing) to authenticate the source of the messages. Some VoIP equipment now supports this H.235 feature in response to service provider requirements.

B. SIP

SIP [26] is a control (or signaling) protocol similar to HTTP. It is a protocol that can set up and tear down any type of session. SIP call control uses SDP [27] to describe the details of the call (i.e., audio, video, a shared application, codec type, size of packets, etc.). SIP uses a URI[7] to identify a *logical* destination, not an IP address. The address could be a nickname, an e-mail address (e.g., sip:schulzrinne@cs.columbia.edu), or a telephone number. In addition to setting up a phone call, SIP can notify users of *events*, such as "I am online," "a person entered the room," or "e-mail has arrived." SIP can also be used to send instant text messages.

SIP allows the easy addition of new services by third parties. Microsoft has included a SIP stack in Windows XP, its latest desktop operating system, and it has a definite schedule for rolling out a new .NET server API that is the successor to the Windows 2000 server. Since SIP will support intelligent devices that need little application support from the network as well as unintelligent devices that need a lot of support from the network, we have an opportunity analogous to the transition from shared computers to personal computers. In the 1960s and 1970s, we used dumb terminals to access applications on a mainframe computer shared by many hundreds

[7]A URI is a pointer to a resource that generates different responses at different times, depending on the input. A URI does not depend on the location of the resource. A URI usually consists of three parts: the protocol for communicating with the server (e.g., SIP), the name of the server (e.g., *www.nice.com*), and the name of the resource. A URL is a common form of URI; the reader need not worry about the difference.

Table 3
SIP Service Options

PSTN-like services	Create new services
Caller ID	Web/Voice integration
PBX-like features	Programmable services
Call forwarding	Multi-destination routing
Call transfer	Presence
AIN-like features	Instant messaging
Freephone	Multimedia
Find me/follow me	Event notification
Conference calls	Caller and called party preferences
	Unified messaging

of users. Starting in the 1980s, we began to use sophisticated applications on a PC, but we were also able to use the PC as a communications terminal to gain access to applications and databases on shared computers (servers) in the network. SIP hosts with various degrees of sophistication will perform some functions locally while allowing us to access applications in the network. SIP is different from H.323 in this regard. Whereas the H.323 model requires application interaction through call control, SIP users can interact directly with applications.

SIP can be used to create new services in addition to replicating traditional telephone services. Presence and instant messaging is an example of a new type of service that can use SIP. There are several popular instant-messaging systems that allow users to create buddy lists and convey status to other member of the buddy list. Status messages can show that one is talking on the phone, or in an important meeting, out to lunch, or available to talk. The members of the buddy list can use these "presence" status messages to choose an appropriate time to make a phone call, rather than interrupting at an inopportune time. Several leading suppliers of instant messaging software have committed to converting their systems to the use of SIP.

Table 3 describes some of the types of services that can be offered using SIP.

Using a client–server model, SIP defines logical entities that may be implemented separately or together in the same product. Clients send SIP requests, whereas servers accept SIP requests, execute the requested methods, and respond.

The SIP specification defines six request methods:

- REGISTER allows either the user or a third party to register contact information with a SIP server.
- INVITE initiates the call signaling sequence.
- ACK and CANCEL support session setup.
- BYE terminates a session.
- OPTIONS queries a server about its capabilities.

The SIP protocol is structured into four layers and has six categories of responses. [24]

Some of the important SIP functional entities are listed below.

- User agent performs the functions of both a user agent client, which initiates a SIP request, and a user agent server, which contacts the user when a SIP request is received and returns a response on behalf of the user.

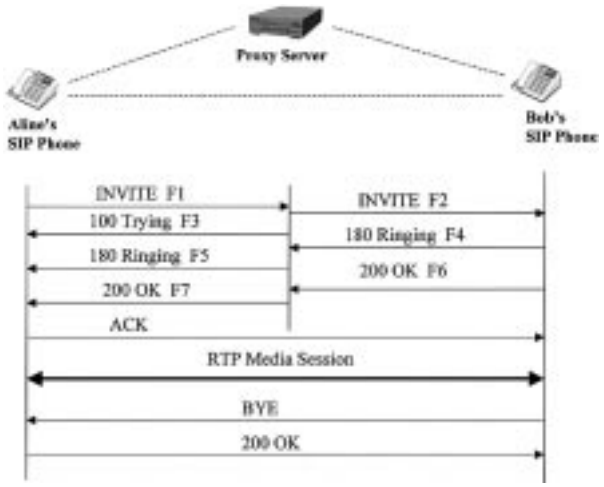

Fig. 14. SIP session setup with one proxy server.

- SIP proxy acts as both a SIP client and a SIP server in making SIP requests on behalf of other SIP clients. A SIP proxy server may be either stateful or stateless. A proxy server must be stateful to support TCP, or to support a variety of services. However, a stateless proxy server scales better (supports higher call volumes).
- Registrar is a SIP server that receives, authenticates and accepts REGISTER requests from SIP clients. It may be collocated with a SIP proxy server.
- Location server stores user information in a database and helps determine where (to what IP address) to send a request. It may also be collocated with a SIP proxy server
- Redirect server is stateless. It responds to a SIP request with an address where the request originator can contact the desired entity directly. It does not accept calls or initiate its own requests.

We will use simple examples to explain basic SIP operations. The first example uses a single proxy, as would be likely for SIP-based IP telephony within a single enterprise building or campus.

Aline calls Bob to ask a question about SIP. Aline and Bob work in the same corporate campus of buildings served by the same SIP proxy server. Since Aline and Bob do not call each other regularly, Aline's SIP phone does not have the IP address of Bob's SIP phone. Therefore, the SIP signaling goes through the SIP proxy server. Aline dials Bob's private number (555–6666). Her SIP phone converts this private number into a related SIP URI (sip:555–6666@nice.com) and sends an INVITE to the SIP proxy server. Fig. 14 shows the SIP message exchange for this example.

SIP uses a request/response transaction model similar to HTTP. Each transaction starts with a request (in simple text) that invokes a server function ("method") and ends with a response. In our example, Aline's SIP phone starts the transaction by sending an INVITE request to Bob's SIP URI (sip:555–6666@nice.com). The INVITE request contains header fields that provide information used in processing the message, such as a call identifier, the destination address,

the originator's address, and the requested session type. Here is Aline's INVITE (message F1 in Fig. 14):

INVITE sip:bob@nice.com SIP/3.0
Via: SIP/3.0/UDP 192.2.4.4:5060
To: Bob < sip:555-6666@nice.com >
From: Aline < sip:555-1234@nice.com >; tag=203 941 885
Call-ID: b95c5d87f7721@192.2.4.4
Cseq: 26 563 897 INVITE
Contact: < sip:555-1234@192.2.4.4 >
Content-Type: application/sdp
Contact-Length: 142
(Aline's SDP not shown)

The first line gives the method name (INVITE). We will describe the header fields in the following lines of the example INVITE message, which contains a minimum required set:

Via contains the IP address (192.2.4.4), port number (5060), and transport protocol (UDP) that Aline wants Bob to use in his response.

To contains a display name (Bob) and a SIP URI (sip:555–6666@nice.com) toward which this request was sent.

From contains a display name (Aline) and a SIP URI (sip:555–1234@nice.com) that identify the request originator.

Call-ID contains a globally unique identifier for this call.

These three lines (*To*, *From*, and *Call-ID*) define a peer-to-peer SIP relationship between Aline's SIP phone and Bob's SIP phone that is sometimes referred to as a "dialog."

The command sequence (*Cseq*) contains an integer and a method name. Aline's SIP phone increments the *Cseq* number for each new request.

Contact contains Aline's username and IP address in the form of a SIP URI. While the *Via* header tells Bob's SIP phone where to send a response, the *Contact* header tells both the proxy server and Bob's SIP phone where to send future requests for this dialog.

Content-type describes the message body.

Content-length gives the length (in octets) of the message body.

The body of the SIP message contains a description of the session, such as media type, codec type, packet size, etc., in a format prescribed (usually) by SDP. The way the SIP message carries a SDP message is analogous to the way an HTTP message carries a web page.

Since Aline's SIP phone does not know Bob's IP address, the INVITE message goes first to the SIP proxy server. When it receives the INVITE request, the proxy server sends a 100 Trying response back to Aline's SIP phone, indicating that the proxy is trying to route the INVITE to Bob's SIP phone. In general, SIP responses have a numerical three- digit code followed by a descriptive phrase. This response (Message F3 in Fig. 14) contains the same to, from, call-ID and Cseq header values as the INVITE message, and Aline's SIP phone can correlate this response with what it sent. The proxy server adds another Via header with its own IP address to the INVITE and forwards it (Message F2 in Fig. 14) to Bob's SIP phone.

When Bob's SIP phone receives the INVITE, it alerts (rings) Bob, so that he can decide whether to answer. Since Aline's name is in the To header, Bob's SIP phone could display Aline's name. Bob's SIP phone sends a 180 Ringing response through the proxy server back to Aline's SIP phone. The proxy uses the Via header to determine where to send the response, and it removes its own address from the top. When Aline's SIP phone receives the 180 ringing response, it indicates ringing by displaying a message on the SIP phone display or by an audible ringback tone.

When Bob pushes the speakerphone button, his SIP phone sends a 200 OK response to indicate that he has answered the call. The 200 OK message body contains the SDP media description of the type of session that Bob's SIP phone can establish on this call. Thus there is a two-way exchange of SDP messages, negotiating the capabilities to be used for the call. Aline's SIP phone sends ACK directly to Bob's SIP phone (it does not pass through the stateless proxy server), and Aline can talk to Bob through an RTP media session. Note that the actual voice packets are routed directly from one SIP phone to another, and their headers have no information about the SIP messages or proxy servers that set up the RTP media session.

In this example, Bob is unable to answer Aline's question, but suggests that she call Henry in Dallas. Henry is an SIP expert, but he is with a different company, global.com. Bob has Henry's email address, but not his telephone number. When Bob says goodbye and presses the button, his SIP phone sends a BYE directly to Aline's SIP phone. Aline's SIP phone responds with a 200 OK, which terminates the call, including the RTP media session.

Now Aline calls Henry. Using the laptop computer connected to her SIP phone, Aline types Henry's email address and clicks on the button to establish a SIP phone call. Aline's SIP phone sends an INVITE addressed to Henry's SIP URI, which is based on his email address (henry@global.com). Since the Nice.com proxy server does not know how to route the call to Henry, it uses domain name service (DNS) to find the global.com SIP server.

Actually, what the Nice.com server needs is a list of next hops that can be used to reach the global.com server. The *next hop* is defined by the combination of IP address, port and transport protocol. The SIP specification gives an algorithm for determining an ordered list of next hops.

Aline's INVITE (message F1 in Fig. 15) looks similar to the one she sent to Bob:

INVITE sip:henry@global.com SIP/3/0
Via: SIP/3.0/UDP 192.2.4.4:5060
To: Henry < sip:henry@global.com >
From: Aline < sip:aline@nice.com >; tag=9 817 514 140
Call-ID:z73a3b65d55609@192.2.4.4
Cseq: 704 452 INVITE
Contact: < sip:aline@192.2.4.4 >
Content-Type: application/sdp, etc.

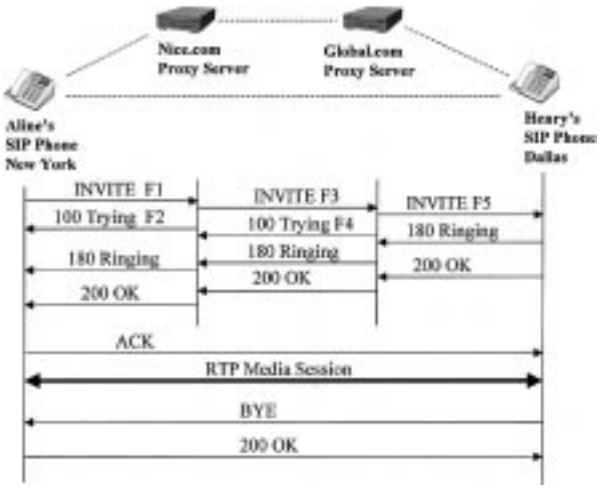

Fig. 15. SIP call setup with two proxy servers.

Note that, in this INVITE message, the SIP URI's are based on email addresses instead of telephone numbers. The flow of messages is similar to the setup of the call to Bob, except that the SIP messages now pass through the global.com proxy server as well as the nice.com proxy server, as shown in Fig. 15.

SIP allows proxy servers to make complex decisions about where to send the INVITE. In the example, Henry could have been traveling and had his calls forwarded to a company office in Washington, DC. A proxy server can send an INVITE to several locations at the same time, so the call could be routed simultaneously to Henry's voicemail server in Dallas and his guest office in Washington. If Henry answers the call in Washington, the session with the voicemail server can be terminated.

The INVITE request could contain information to be used by the destination proxy server to determine the set of destinations to ring. For instance, destination sets may be constructed based on time of day, the interface on which the request has arrived, failure of previous requests, or current level of utilization of a call distributor. Aline might program her SIP phone to request a follow-me service only to business locations. On the other hand, Henry might program his SIP server to forward calls to his mobile phone, but only a privileged access list (family and boss?) would have calls forwarded to his home.

SIP facilitates mobility, because the same person can use different terminals with the same address and same services. SIP promises to be used by many programmers to develop new services. Many of these new services may be offered on the public Internet. There are, however, some complications to using an open peer-to-peer signaling and control protocol like SIP. One of them is security.

SIP Security Issues: Like the Internet, SIP has promise, but SIP in a shared network raises some security concerns that should be addressed before it is widely adopted. SIP solutions encounter security issues in preserving confidentiality and integrity of SIP requests, preventing replay attacks or message spoofing, ensuring the privacy of the participants in a session, and preventing denial of service (DOS) attacks.

SIP messages may contain sensitive sender information, including who communicates with whom and for how long, and perhaps their email address or from what IP address they participate in calls. Both individuals and corporations may wish that this kind of information be kept private.

The first solution that comes to mind is encryption. SIP encryption uses the known port 5061 instead of 5060. Encrypting the entire SIP request or response on the links between SIP entities can prevent packet sniffers and other eavesdroppers from discovering who is calling whom. However, SIP requests and responses cannot be entirely encrypted end to end because message fields such as the *Request-URI, Route and Via* fields need to be visible to proxies so that SIP requests can be routed properly. Furthermore, SIP encryption is defined to include the SDP payload. Network entities will be unable to determine the codec used, the packet size, or the amount of bandwidth required for the RTP stream. Indeed, when SIP encryption is used, network entities may not even be able to determine whether the call is voice only or includes video. SIP requests and responses can be protected by transport or network layer security mechanisms. IPSec is a network layer security protocol that is most suited to virtual private network (VPN) architectures.

For SIP to function securely, proxy servers must be part of the SIP network trust relationship. TLS [28] is a transport layer protocol[8] like TCP and UDP, and any of them can be specified as the transport protocol in the *Via* header field or a SIP-URI. TLS is suitable for architectures in which the hosts are joined by a chain of trust. (Aline trusts the nice.com proxy server, which in turn trusts the global.com proxy server, which Henry trusts.) If such a SIP network trust relationship were not established, there is a possibility that rogue proxy servers might modify the signaling (e.g., adding *Via* headers)

In the SIP call setup example, Aline used Henry's email address to call Henry. Since an email address is often guessable from a person's name and organizational affiliation, the concept of an unlisted "phone number" has to be implemented differently, perhaps through a user location service (in the proxy server) that has access lists, so that each user can restrict what kind of location and availability information is given to certain classes of callers.

Caller identity is also an issue. Consider ways to manipulate the user location service to get access to someone. The *From* header field usually identifies the requestor, but in many cases the end user controls this information, and the end user may not be who he claims to be. To prevent this kind of fraud, SIP provides a cryptographic authentication mechanism. More specifically, SIP authentication uses a stateless challenge-based mechanism. A proxy server or user agent may challenge the initiator of any request to provide assurance of identity.

DOS is an insidious security problem involving the malicious routing of large volumes of traffic at a particular network interface. Typically, one or a few users launch a dis-

[8]TLS, an IETF protocol based on SSL 3.0, provides an encrypted connection between an authenticated client and server.

tributed DOS attack by commandeering multiple network hosts to overload a target host. Such a flood of messages directed at a SIP proxy server could overload the proxy server resources and prevent authentic SIP messages from reaching their destinations.

When a SIP proxy server is operating on a computer that is routable from the public Internet, it should be a part of an administrative domain with secure routing policies, including the blocking of source-routed traffic, especially filtering ping traffic. However, we can expect attackers to become more sophisticated.

An attacker could falsify the *Via* header in a request, identifying a target proxy server as the originator of the message, and then send the bogus message request to a large number of SIP network elements. The SIP user agents or proxies would generate response traffic aimed at the target, thereby creating a denial of service attack.

Similarly, an attacker could falsify *Route* headers that identify the target and then send the request to forking proxies that would amplify messages sent to the target.

If REGISTER requests are not authenticated and authorized properly, the registrar and associated proxy servers could be commandeered and used in a DOS attack by registering a large number of contacts with the same target host.

There are several ways to protect a host from being commandeered for a DOS attack. The user agent could invoke the authentication/challenge process for each call. The SIP proxy server could limit the number of near simultaneous call requests going to a single host. The user agent could disallow requests that do not use a persistent security association established using TLS or IPSec to the proxy server. This solution is also appropriate for two proxy servers that trust one another.

Administrative domains that participate in security associations can use TLS and/or IPSec to aggregate traffic over secure tunnels and sockets to and from "bastion hosts," which can absorb DOS attacks, ensuring that SIP hosts behind them in the administrative domain do not become overloaded with bogus messages. This solution seems suitable for service providers carrying large volumes of SIP traffic.

Note that SIP security has nothing to do with media security or the security of other protocols carried in SIP messages. Specifically, RTP media encryption is a separate topic.

C. Master/Slave Architectures

The call processing function can be separated from the VoIP gateway function. We can define a new entity, a "call agent," to control the gateways and perform call processing. The physical product implementing the call agent function need not be located near the gateway and could control many gateways. This architecture simplifies the VoIP gateway product, allowing the gateway to be located in homes and small offices at low cost.

Consider the diagram of a circuit-switched network in Fig. 16. The switches send telephone traffic directly from one to the other, but communicate call-signaling information

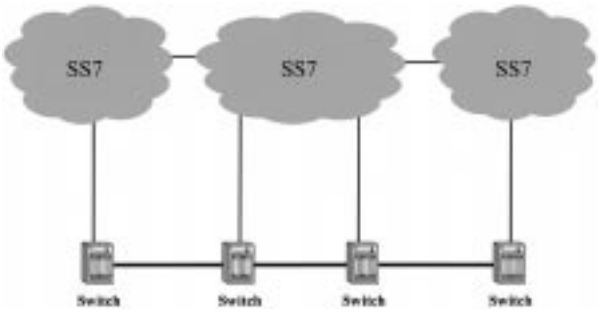

Fig. 16. Existing circuit switched networks.

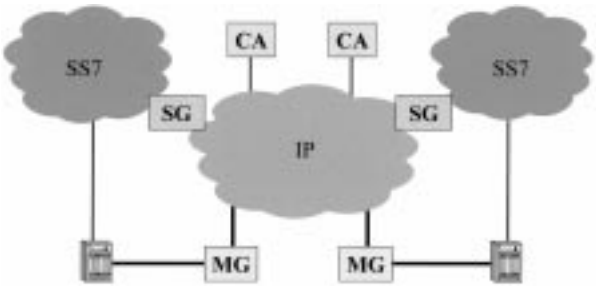

Fig. 17. Master/slave architecture involving call agents, signaling, and media gateways.

among each other using a separate packet-signaling SS7 network. Note that, although packet switched, the SS7 protocol is not related to the IP.

Some network engineers say IP telephony must replace the PSTN in such a way that the essential functions of the PSTN will continue to work throughout an extended migration period. This leads to two types of gateways. Media gateways accept voice or "media" traffic from the circuit switches and packetize the voice to be transmitted over the IP network. Signaling gateways connect the signaling (e.g., SS7) networks and IP networks, so that the call agents connected to the IP network can communicate with the circuit switches connected to the signaling networks, as diagrammed in Fig. 17.

The MG allows connections between dissimilar networks by providing media conversion and/or transcoding functions. For example, an MG may receive packets from an IP network, depacketize them, transcode them, and pass the media stream to a switched circuit network. It would reverse the order of the functions for media streams received from the switched circuit network. Although an MG may perform media adaptation, in some cases an MG may act like a switch in joining two terminations or resources of the same type. Hence, other functions that an MG could perform include a conference bridge with all packet interfaces, an interactive voice response unit, or a voice recognition system. An MG also supports resource functions including event notification, resource allocation and management, as well as system functions, such as establishing and maintaining an association with the Call Agent.

An SG function resides at the edge of the data network, relaying, translating or terminating call control signals between the packet data network and the circuit switched tele-

phony network. An SS7-IP gateway would employ the SG function. On the other hand, the MG could also employ an SG function to process traditional telephony signaling associated with trunk or line terminations at the MG, such as the D channel of an ISDN BRI line or PRI trunk.

The call agent, which is often termed the "media gateway controller," must communicate with the media gateway to control its actions. Several protocols have been developed for this type of communication, including simple gateway control protocol (SGCP) [29], IP device control (IPDC) protocol, media gateway control protocol (MGCP) [30]–[32], and Megaco/H.248 [33]. SGCP is the original ASCII string-based master-slave signaling protocol for VoIP. MGCP followed the following year, combining characteristics of SGCP and IPDC with more capabilities. Megaco is a similar protocol that the IETF has developed with still more capabilities.

Although the MGCP RFC was not a standards-track document, many vendors have implemented gateways and call agents using MGCP. It is also the basis for the network-based call signaling (NCS) protocol developed by the PacketCable group of Cable Labs. There are several available implementations of NCS 1.0.

Both SCGP and MGCP are designed as distributed system protocols that give the user the appearance of a single VoIP system. They are stateless protocols in the sense that the sequence of transactions between the MG and the call agent can be performed without any memory of previous transactions. On the other hand, MGCP does require the MGC to keep call state.

Both MGCP and Megaco support the following media gateway functions:

- Create, modify and delete connections using any combination of transit network, including frame relay, ATM, TDM, Ethernet or analog. Connections can be established for transmission of audio packets over several types of bearer networks:
  - IP networks using RTP and/or UDP;
  - ATM networks using AAL2 or another adaptation layer;
  - an internal connection, such as the TDM backplane or the interconnection bus of a gateway. This is used for connections that terminate in a gateway but are immediately rerouted over the telephone network ("hairpin" connections).
- Detect or generate events on end points or connections. For example, a gateway may detect dialed digits or generate a ringback tone on a connection. A call agent will use MGCP to send "notification requests" which include a list of "events" that the media gateways are to detect. The protocol uses the "Requested Events" list, the "Digit Map" and the "Detect Events" list in the handling of these events. When it detects an event, the media gateway takes some action, as specified by the call agent, such as reporting the event or applying another tone to the connection.
- Collect digits according to a digit map received from the call agent, and send a complete set of dialed digits to the call agent.
- Allow mid-call changes, such as call hold, playing announcements, and conferencing.
- Report call statistics.

The digit collection mechanism allows call agents to serve large numbers of residential and small business gateways. It can be used to collect not only dialed destination telephone numbers, but also access codes, credit card numbers, etc. The requirement to collect digits according to a digit map is related to the efficiency of communications between the MG and the call agent in a distributed system. If the gateway were to send each dialed digit to the call agent separately, as soon as they were dialed, there would be an unnecessarily large number of interactions. Therefore, the gateway should store the digits in a buffer and send a complete set to the call agent. The gateway needs to know how many digits to accumulate before transmission. For example, the single digit "0" could be used to connect to the local operator, four digits "xxxx" could be a local extension number, "8xxxxxxx" could be a number from a company's private dial plan, and 9011 + up to 15 digits could be an international number. The distributed VoIP system can use MGCP to send the gateway a digit map that corresponds to the dial plan. Digit maps simply define a way for the gateway to match sequences of dialed digits against a grammar.

Aside from some differences in terminology, the Megaco protocol gives the call agent more flexibility of transport type and control over the media gateway, as well as some hooks for applications such as video conferencing. Both MGCP and Megaco provide a procedure for the call agent to send a package of properties, signals, or events, for example, to the gateway for use on the lines and trunks attached to the gateway. The package contents are not a part of either protocol, so the implementer can define or change packages without any change to the protocol. Megaco has a defined way for the call agent and the gateway to negotiate the version to be used, but MGCP does not have a version control mechanism, so one must rely on a vendor proprietary negotiation process.

In the areas of security and quality of service, Megaco is more flexible than MGCP. While MGCP supports only IPSEC, Megaco also supports an authentication header. Both protocols support authentication of the source address. While MGCP only supports UDP for signaling messages, Megaco supports UDP, TCP, ATM, and SCTP. Megaco also has better stream management and resource allocation mechanisms.

Either MGCP or Megaco (or even SGCP or IPDC) may be used for a master-slave VoIP architecture, especially when the goal is to control many low-cost IP telephony gateways. For communications among call agents, or for control of trunk groups, SIP may be more appropriate. While MGCP and Megaco have specific verbs for VoIP call control, SIP allows a single primitive to be used to provide different services. Consequently, SIP offers the promise of supporting a wide range of services beyond basic telephony, including instant messaging, presence management, and voice-enabled

web-based e-commerce, and SIP facilitates new application development by independent third parties. Some soft switch vendors use MGCP or Megaco to control gateways, but use SIP at the application layer.

VI. Telephony Routing Over IP (TRIP)

For many years to come, there will be more telephones served by the global PSTN than by IP telephony. Users of IP phones will want to call people who use traditional telephones. There are an increasing number of gateways that support VoIP on one side and are connected to the PSTN on the other. Many gateways could complete a call. How does the system find the right gateway?

Telephony routing over IP (TRIP) addresses the following problem: "given a phone number that corresponds to a terminal on a circuit switched network, determine the IP address of a gateway capable of completing a call to that phone number"[34] This is essentially an address to route translation problem.

TRIP does not help find the IP address of a personal computer that serves as an interface to a telephone. For example, a service provider might want to deliver an instant message to a PC associated with a telephone. Directory protocols are better suited to such a problem.

TRIP also does not facilitate calls from a traditional phone to a personal computer that may be used for VoIP. Since IP addresses are often assigned by DHCP or by dialup network access servers, it seems to be a good idea to assign a permanent telephone number to a VoIP terminal, even if that terminal is a computer. A PSTN switch would have to obtain a mapping from this telephone number to an IP address for the PC. This is a name-to-address translation problem that can also be solved using a directory protocol.

The problem that TRIP does address is a complex one. Given the universal connectivity of the PSTN, nearly any VoIP/PSTN gateway could potentially complete a phone call to anywhere in the world. However, there are many factors that influence the decision of which gateway to choose. The calling party may be using signaling or media protocols that are not supported by all gateways. Capacity must also be taken into account in the gateway selection process. Some gateways may support thousands of simultaneous calls, while others support very few. The gateway service provider will want to charge enough to offset costs and make a profit. The user has to pay something, and the gateway service provider has to be paid. However, the end user may be a customer of an IP Telephony service provider who does not own the gateway, but has some business relationship with the gateway service provider. The primary IP telephony service provider may have some gateways as well and is likely to have some policy about what calls are routed to its own gateways and what calls are routed to business partner gateways. Because of these complexities, there cannot be a universal gateway directory. Service providers must exchange information on the availability of gateways, subject to policy. Using this information, each service provider can create its own local database of available gateways.

The main functional component of TRIP is the LS, a logical entity that has access to the telephony routing information base (TRIB). The TRIB combines information on gateways available from within its telephony administrative domain with information on gateways available (based on policy) in other IT administrative domains.

TRIP is modeled after the IETF interdomain routing protocol BGP-4 [35], in that it is a protocol for sharing reachability information across administrative domains. As border routers use BGP-4 to distribute IP routes across IP administrative domains, so location servers can use TRIP to distribute telephone routes among telephony administrative domains. "TRIP uses BGP's interdomain transport mechanism, BGP's peer communication, BGP's finite state machine, and similar formats and attributes as BGP" [36] However, TRIP also has some link state features and uses intradomain flooding similar to OSPF. There are some other important differences between BGP and TRIP.

- TRIP is an application layer protocol, whereas BGP is a network layer protocol.
- There may be many intermediate network and IP service providers between location servers that run TRIP. BGP usually runs between routers in adjacent networks.
- TRIP peers exchange information describing routes to application layer location servers.
- TRIP uses a transport network to communicate between servers. It has nothing to do with routing table advertisements.
- There may be islands of TRIP connectivity. There may not be VoIP connectivity among the islands, but within each island, any gateway can have complete connectivity to the entire PSTN.
- Compared to IP routes, many more parameters are necessary to describe gateway routes. Hence gateway routes are relatively more complex.

To illustrate the TRIP architecture, Fig. 18 shows a diagram of the relationship of three ITADs. Each ITAD has at least one LS. ITAD1 has both end users and gateways. ITAD2 has only end users. ITAD3 has only gateways. An LS learns about the gateways in their domain through an out-of-band intradomain protocol, which is represented by the dashed lines in ITAD3. The administrative domains have agreements that allow the LSs to exchange gateway data. Using TRIP, the LS in ITAD2 can learn about the three gateways in ITAD3, as well as the two gateways in ITAD1. The end users in ITAD2 can use a non-TRIP protocol to access the LS databases. The LS in ITAD1 can learn about the gateways in ITAD3 from the LS in ITAD2; this information might be in an aggregated advertisement.

A. Example — Clearinghouse

A clearinghouse is like a route reflector. Members of the clearinghouse agree to accept each other's IP telephony traffic at their gateways. Clearinghouse members can use TRIP to exchange routes with the clearinghouse. Fig. 19

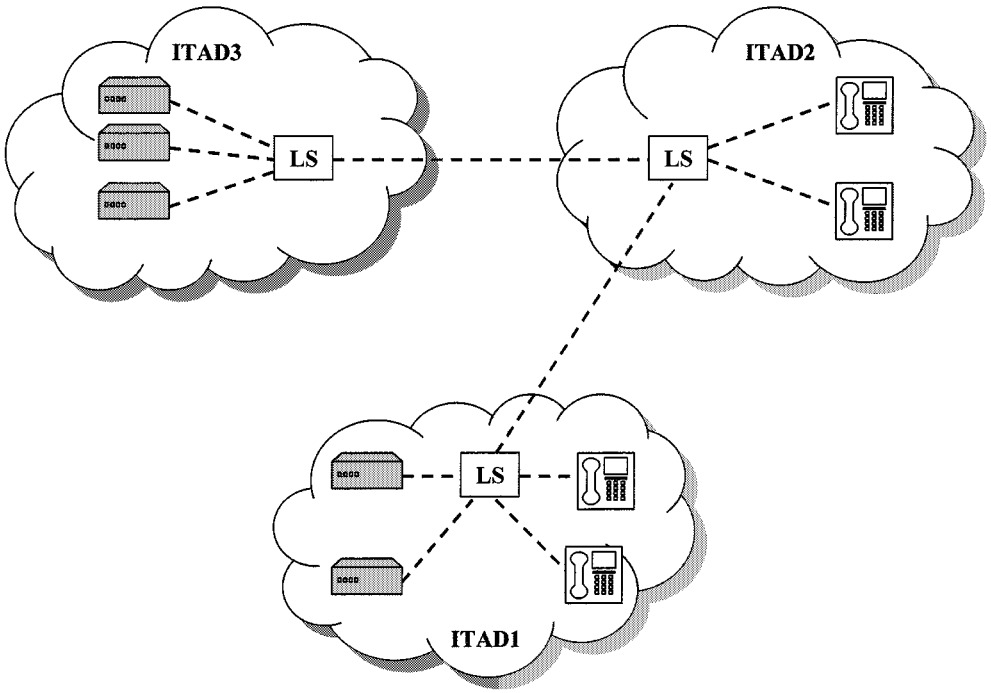

Fig. 18. TRIP architecture.

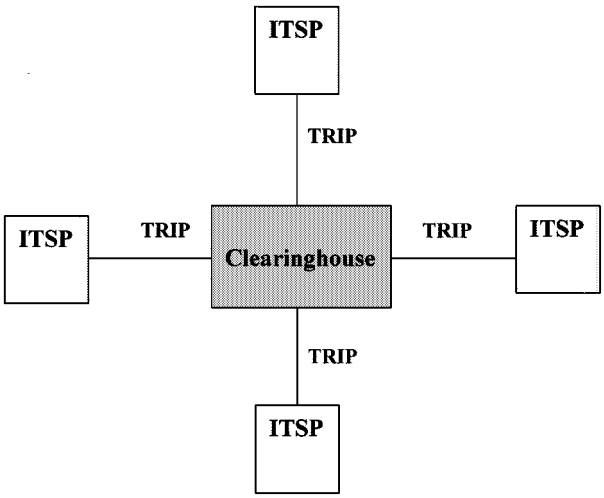

Fig. 19. IT clearinghouse using TRIP.

shows a diagram of four ITSPs using TRIP to exchange gateway routes with the clearinghouse.

VII. VoIP Issues With NAT and Firewalls

VoIP is one of many IP applications that have problems traversing NATs and firewalls. While there are solutions, they all increase the expense and operational complexity of Internet telephony.

NAT allows private networks to connect to a common network (e.g., the Internet) although they have overlapping address realms. NAT is used and tolerated as a means to ameliorate IPv4 address depletion by allowing globally registered IP addresses to be reused or shared by several hosts. NAT also protects the privacy of the internal network topology and addresses. NAT routers, placed at the border between private and public networks, convert the private addresses in each IP packet into IANA-registered public IP addresses. In addition to modifying the IP address, NAT must modify the IP checksum and the TCP checksum. The packet sender and receiver should remain unaware that NAT is taking place. Firewalls commonly support NAT.

There are both static and dynamic NAT devices and routers, but dynamic NAT is more common today. Edge devices that run dynamic NAT allow an entire private IP subnet to share a pool of public IP addresses. So long as a private host has an outgoing connection, incoming packets sent to the public NAT address can reach it. After the connection is terminated or times out, the binding expires, and the NAT returns the address to the pool for reuse.

Network Address Port Translation (NAPT), a variation of dynamic NAT, allows many hosts to share a single IP address by multiplexing streams differentiated by TCP/UDP port number. For example, suppose private hosts 10.0.0.2 and 10.0.0.3 both send packets from source port 1180. A NAPT router might translate these to a single public IP address 9.245.160.1 and two different source ports, say 5431 and 5432. The NAPT would route response traffic for port 5431 to 10.0.0.2:1180, while traffic to port 5432 would go to 10.0.0.3:1180.

Multihost residential users, teleworkers, and small businesses use NAPT devices (sometimes called SOHO routers) to allow multiple computers to share a single public IP address for outbound traffic while blocking inbound session requests. A provider of DSL or cable modem service often assigns the single IP address. A NAPT router allows several computers to share that IP addresss. Enterprises with private address realms also use NAPT.

A. Protocol Complications With NAT

VoIP is one of many applications that can be adversely affected when IP clients connect through a NAT or NAPT. The NAT device may use an application level gateway (ALG). An ALG examines and modifies application payload content to allow packets from a specific application or protocol to pass through the NAT transparently. However, few NAT devices offer ALG functions for VoIP, and some protocols are not amenable to this approach.

There are several categories of problems that VoIP applications have with NAT.

1) Many applications fail with NAT because the packets contain IP address or port information in the payload. A simple NAT only changes the IP address of the packet itself, not the IP addresses and ports in the payload. In the case of H.323, it is the call setup packets that contain the address and port information in the payload.
2) H.323 and SIP, as well as other applications such as FTP and RTSP, use bundled sessions. They exchange address and port parameters within a control session to establish data sessions. NAT cannot determine the inter-dependency of the bundled sessions and assigns unrelated addresses and port numbers to these sessions, which does not work.
3) An IP application (such as IP phone) that attempts to originate a session from an external realm will be able to locate its peer in a private realm only when it knows the externally assigned IP address ahead of time. This is a problem for a traditional dynamic NAT, which only permits sessions to be established in one direction.
4) SIP messages may carry URL's that specify signaling addresses in the "Contact," "To," and "From" fields. Once they traverse a NAT, the IP addresses and domain names in the host port portion of the URL may not be valid.

B. H.323 Characteristics

H.323 is a protocol suite that uses multiple UDP streams and dynamic ports. An H.323 call consists of many different simultaneous connections. There are two or more TCP connections for each call. For a voice conference call, there may be as many as four different UDP ports open. All connections except one are made to dynamic ports.

During call setup, a TCP connection carries H.225 signaling, including the Q.931 messages. During slow start call setup, the H.245 messages carry the terminal characteristics and requested call parameters in a TCP connection separate from the H.225 data stream. There is no well-known port associated with the H.245 channel. Instead, the H.225 channel is used to convey the H.245 port information. The firewall needs to monitor the H.225 channel for the H.245 port, because it is not possible to implement a sufficiently stringent static rule that allows an H.245 connection while blocking other undesired TCP connections.

During FastStart call setup, the H.245 message is imbedded in the H.225 message along with the Q.931 message. To work properly, an ALG has to modify the addresses inside these messages. Q.931 and H.245 messages are encoded in ASN.1 in the packet payload, and they are variable in length. Of course, these difficulties have not prevented vendors from developing NAT-enabled firewalls with ALG functions that allow H.323 to pass through. However, small inexpensive NATs and firewalls do not have H.323 ALGs.

C. NAT/Firewall Problems With RTP

Media transport for all IP multimedia applications, including VoIP, uses RTP in conjunction with UDP. There are no fixed ports associated with RTP, and it is impossible to define static rules that can allow RTP media through a firewall without also allowing undesirable packets to pass through. Furthermore, RTP and RTCP ports are paired, with RTP receiving an even port number, and RTCP receiving the next higher odd port number. NAPT typically assigns new port numbers at random, breaking the pair relationship of RTP and RTCP port numbers. Also, for multimedia sessions, the NAT functions scramble the source and destination addresses used for packets and without special processing by the NAT, these will not correspond with the values used in the control connections. Thus, the multimedia devices may not associate the RTP sessions with the correct call.

D. NAT/Firewall Traversal

We have observed some problems that session-oriented protocols such as VoIP experience with NATs and firewalls. There are four types of solutions.

The first solution is a proxy placed at the border between two domains (e.g., between a private IP address space and a public address space). The proxy would terminate sessions with both hosts, or with both client and server, and relay application signaling messages as well RTP media streams transparently between the two hosts. Only designated protocols, such as SIP or H.323, would pass through the proxy. All other traffic would have to traverse the NAT and/or firewall to communicate between the two domains.

The second solution is an ALG embedded in the NAT or firewall. The ALG does not terminate sessions, but rather examines and modifies application payload content to allow VoIP traffic traverse the NAT/firewall. The ALG is the most common commercial solution now, but ALG-enabled firewalls tend to be somewhat expensive. Placing several ALG's within the same firewall increases its complexity and may degrade performance. Futhermore, any changes in the VoIP protocol used will require a new ALG from the firewall vendor for all the previously installed firewalls that VoIP has to traverse. The upgrade also tends to be expensive.

A third approach is to remove the application logic from the NAT/firewall. A new type of firewall dynamically opens "pinholes" to let a VoIP call through it, without exposing the private network by allowing penetration by a wide range of IP addresses. A firewall control proxy (FCP), placed in the signaling path between private and public domains, monitors the call setup signals (such as H.323 and SIP) and commands

the firewall to allow RTP streams destined to the appropriate IP addresses to pass through. For protocols such as SIP and H.323, moving stateful inspection and manipulation of signaling packets out of NAT/firewalls should improve scalability and performance while reducing development costs.

The IETF is exploring this third approach in the Middlebox Communications (Midcom) Working Group. The MidCom group is trying to agree on a control protocol that would enable another device (an FCP, basically) to control middle boxes such as NATs and firewalls. By providing a generalized standard interface communications interface for the middle boxes, the working group hopes to improve performance, lower software development and maintenance costs, and easier deployment of new applications. [37]

These two types of solutions, ALGs and FCP/MidCom, require changes to NAT and firewall design. A fourth type of solution seeks a means to "traverse" the NAT and/or firewall without changing its design, and without requiring it to perform additional processing. The challenge of this type of solution is to allow VoIP signaling and media streams to traverse the NAT and/or firewall without compromising security.

Two Internet drafts [38], [39] have suggested ways to allow VoIP and other multimedia traffic to traverse NATs and firewalls. Although the methods are different, they both employ external proxy servers with persistent connections to the VoIP/multimedia devices. Two essential elements of these traversal methods are as follows.

1) The user behind the NAT must send the first packet to establish the NAT binding.
2) Media sent to user A must be to the source port from which A's media came.

To that end, devices in the private address realms communicate with the proxy servers in the public address realm via "probe packets" or "cookies." The proxy servers associate the origination address/port pair with the "token" or "cookie."

VIII. Summary and Conclusion

Providing reliable, high-quality voice communications over a network designed for data communications is a complex engineering challenge. Factors involved in designing a high-quality VoIP system include the choice of codec and call signaling protocol. There are engineering tradeoffs between delay and efficiency of bandwidth utilization. Packetized voice has larger end-to-end delays than a TDM system. One reason is that an IP network typically has higher delay variation than a TDM system. Since any packets that arrive later than the length of the jitter buffer are discarded, the jitter buffer delay must be set to the maximum delay variation that we expect, in order to achieve low packet loss probability. The jitter buffer delay becomes a major component of the end-to-end delay budget, to which must be added the encoding delay and packetization delay. VoIP performance can be improved by network QoS techniques (such as differentiated services) that are not widely available in the public Internet today, but may be deployed by specialized commercial IP networks.

We have compared several VoIP signaling protocols. H.323 and SIP use a peer-to-peer control-signaling paradigm, while MGCP and Megaco use a master-slave control-signaling paradigm. H.323 had the early lead among VoIP services, but SIP is becoming more popular. Either MGCP or Megaco is appropriate for the control of many low-cost IP telephony residential gateways. For communications among call agents, or for control of trunk groups, SIP may be more appropriate. SIP also offers the promise of supporting a wide range of services beyond basic telephony, including instant messaging, presence management and voice-enabled web-based e-commerce.

We have reviewed the motivation and characteristics of TRIP, a location server protocol for the inter-domain advertising of PSTN destinations reachable from participating gateways, and the attributes of those gateways. We also reviewed the challenges that VoIP signaling protocols and media packet streams have in coping with network address translation and firewalls.

While posing complex engineering challenges, VoIP remains a topic of extensive product development and intense standards activity. We can expect more VoIP solutions and more protocol developments in the near future, as well as an increasing volume of telephone traffic using this technology.

References

[1] M. Perkins, K. Evans, D. Pascal, and L. Thorpe, "Characterizing the subjective performance of the ITU-T 8 kb/s speech coding algorithm – ITU-T G.729," *IEEE Commun. Mag.*, vol. 35, pp. 74–81, Sept. 1997.

[2] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," IETF RFC 1889, 1996.

[3] A. Benyassine, E. Schlomot, H. Y. Su, D. Massaloux, C. Lamblin, and J. P. Petit, "ITU-T G.729 annex B: A silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications," *IEEE Commun. Mag.*, vol. 35, pp. 64–73, Sept. 1997.

[4] M. Degermark, B. Nordgren, and S. Pink, "IP Header Compression," IETF RFC 2507, 1999.

[5] S. Casner and V. Jacobson, "Compressing IP/UDP/RTP headers for low-speed serial links," IETF RFC 2508, 1999.

[6] M. Engan, S. Casner, and C. Bormann, "IP header compression over PPP," IETF RFC 2509, 1999.

[7] "Stability and Echo," CCITT Recommendation G.131 , 1988.

[8] "One-way transmission time," ITU-T Recommendation G.114 , 1996.

[9] R. Braden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: An overview," IETF RFC 1633, 1994.

[10] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol (RSVP) version 1 functional specification," IETF RRC 2205, 1997.

[11] S. Shenker, C. Partridge, and R. Guerin, "Specification of guaranteed quality of service," IETF RFC 2212, 1997.

[12] P. White and J. Crowcroft, "The integrated services in the internet: State of the art," *Proc. IEEE*, vol. 85, pp. 1934–1946, Dec. 1997.

[13] D. Awduche, A. Hannan, and X. Xiao, "Applicability statement for extensions to RSVP for LSP tunnels," IETF RFC 3210, 2001.

[14] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wong, and W. Weiss, "An architecture for differentiated services," IETF RFC 2475, 1998.

[15] V. Jacobson, K. Nichols, and K. Poduri, "An expedited forwarding PHB," IETF RFC 2598, 1999.

[16] B. Davie and A. Charney *et al.*, "An expedited forwarding PHB," IETF RFC 3246, 2002, to be published.

[17] A. Charny *et al.*, "Supplemental information for the new definition of the EF PHB (expedited forwarding per hop behavior)," IETF RFC 3247, 2002.
[18] M. Listanti, F. Ricciato, and S. Salsanso. Delivering statistical QoS guarantees using expedited forwarding PHB in a Differentiated Services network. [Online]. Available: http://www1.tlc.polito.it/cour-mayeur/intserv/int8.pdf
[19] L. Andersson, P. Doolan, N. Feldman, A. Fredette, and B. Thomas, "LDP Specification," IETF RFC 3036, 2001.
[20] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP tunnels," IETF RFC 3209, 2001.
[21] B. Jamoussi *et al.*, "Constraint-based LSP setup using LDP," IETF RFC 3212, 2002.
[22] F. Le Faucheur *et al.*, "MPLS support of differentiated services," IETF RFC 3270, 2002.
[23] ——, "Requirements for support of Diff-Serv-Aware MPLS traffic engineering," IETF Internet Draft, work in progress.
[24] ——, "Protocol extensions for support of Diff-Serv-Aware MPLS traffic engineering," IETF Internet Draft, work in progress.
[25] "ITU-T Recommendation H.323: Packet-based multimedia communications systems," International Telecommunication Union , 1997.
[26] J. Rosenberg, H. Schulzrinne, Camarillo, Johnston, Peterson, Sparks, Handley, and Schooler, "SIP: Session initiation protocol v.2.0," IETF RFC 3261, 2002.
[27] M. Handley and V. Jacobson, "SDP: Session description protocol," IETF RFC 2327, 1998.
[28] T. Dierks and C. Allen, "The TLS Protocol, Version 1.0," IETF RFC 2246, 1998.
[29] M. Arango and C. Huitema, "Simple gateway control protocol (SGCP) Version 1.0,", 1998.
[30] M. Arango, A. Dugan, I. Elliott, C. Huitema, and S. Pickett, "Media gateway control protocol (MGCP) Version 1.0," IETF RFC 2705, 1999.
[31] N. Greene, M. Ramalho, and B. Rosen, "Media gateway control protocol architecture and requirements," IETF RFC 2805, 2000.
[32] M. Arango *et al.*, "Media gateway control protocol (MGCP) Version 1.0bis," draft-andreasen-mgcp-rfc2705bis-02.txt, work in progress.
[33] F. Cuervo, N. Greene, A. Rayhan, C. Huitema, B. Rosen, and J. Segers, "Megaco Protocol Version 1.0," IETF RFC 3015, 2000.
[34] J. Rosenberg and H. Schulzrinne, "A Framework for Telephony Routing over IP," IETF RFC 2871, 2000.
[35] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," IETF RFC 1771, 1995.
[36] J. Rosenberg, H. Salama, and M. Squire, "Telephony Routing over IP (TRIP)," IETF RFC 3219, 2002.
[37] P. Srisuresh, J. Kuthan, J. Rosenberg, A. Molitor, and A. Rayhan, "Middlebox Communication Architecture and Framework," draft-ietf-midcom-framework-07, work in progress.
[38] J. Rosenberg and H. Schulzrinne, "SIP traversal through residential and enterprise NAT's and firewalls," Internet Engineering Task Force, Internet Draft, work in progress.
[39] S. Davies, S. Read, and P. Cordell, "Traversal of non-protocol aware firewalls and NATS," Internet Engineering Task Force, Internet Draft, work in progress.

Bur Goode (Senior Member, IEEE) received the B.S. degree in physics and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA. He received the master's degree from the Sloan School of Management, Massachusetts Institute of Technology, Cambridge.

He is currently with AT&T Labs, Weston, CT, developing network architecture and technology for global services. He was formerly with IBM and affiliated companies, including Satellite Business Systems, where he was the architect of the SBS TDMA Demand Assigned System.

Dr. Goode was Guest Editor of the Special Issue on the Global Information Infrastructure for the PROCEEDINGS OF THE IEEE in 1997, as well as Guest Coeditor of the Special Issue on Intelligent Networks for IEEE COMMUNICATIONS MAGAZINE in 1992.