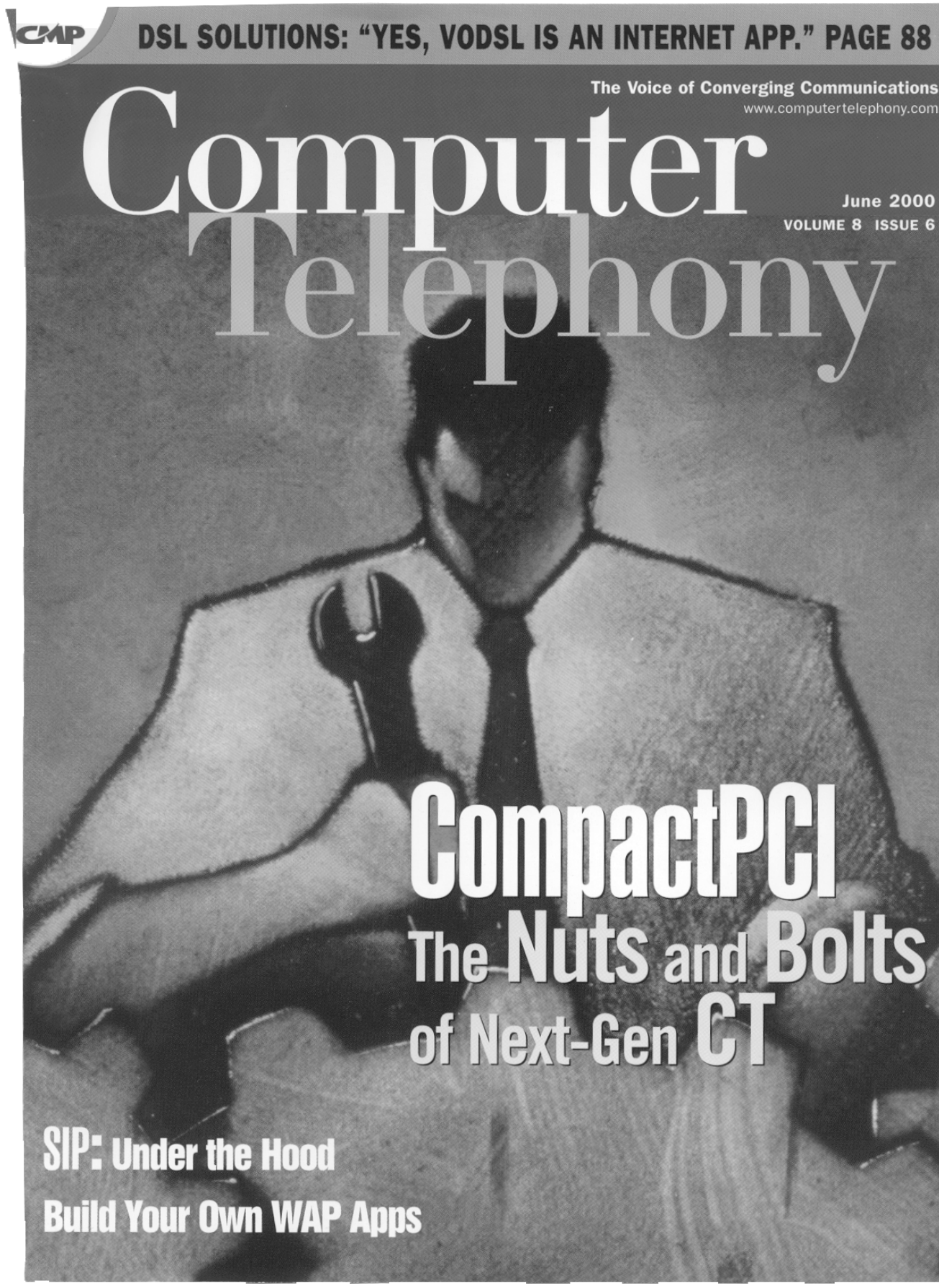


Telcom Ventures Exhibit No. 2021
Samsung Elec. v. Telcom Ventures LLC
IPR2025-00974, U.S. Patent No. 10,674,432

The Session Initiation Protocol (SIP):

A Key Component for Internet Telephony



Last month, we got our first look inside the SIP standard for signaling communications services on the Internet and emerging SIP products. This month, we've gone to principal sources for a more thorough primer.

by Jonathan D. Rosenberg
jdrosen@dynamicsoft.com
and Richard Shockey
rshockey@ix.netcom.com

Since its approval in early 1999 as an official standard, the Session Initiation Protocol (SIP) has gained tremendous market acceptance for signaling communications services on the Internet. What lies behind this success? What problems loom? How does SIP fit in with other solution components? We examine these and other issues in detail.

HISTORY

SIP has its origins in late 1996 as a component of the "Mbone" set of utilities and protocols. The Mbone, or multicast backbone, was an experimental multicast network overlayed on top of the public Internet. It was used for distribution of multimedia content, including talks and seminars, broadcasts of space shuttle launches, and IETF meetings. One of its essential components was a mechanism for inviting users to listen in on an ongoing or future multimedia session on the Internet. Basically — a session initiation protocol. Thus SIP was born.

As an Mbone tool (and as a product of the IETF), SIP was designed with certain assumptions in mind. First was scalability: Since users could reside anywhere on the Internet, the protocol needed to work wide-area from day one. Users could be invited to lots of sessions, so the protocol needed to scale in both directions. A second assumption was component reuse: Rather than inventing new protocol tools, those already developed within the IETF would be used. That included things like MIME, URLs, and SDP (already used for other protocols, such as SAP). This resulted in a protocol that integrated well with other IP applications (such as web and e-mail).

Interoperability was another key goal, although not one specific to SIP. Interoperability is at the heart of IETF's process and operation, as a forum attend-

Telcom Ventures Exhibit No. 2021
Samsung Elec. v. Telcom Ventures LLC
IPR2025-00974, U.S. Patent No. 10,674,432
Page 2 of 14

ed by implementers and operational experts who actually build and deploy the technologies they design. To these practical-minded standardizers, the KISS (Keep It Simple Stupid) principle was the best way to help ensure correctness and interoperability.

Despite its historical strengths, SIP saw relatively slow progress throughout 1996 and 1997. That's about when interest in Internet telephony began to take off. People began to see SIP as a technology that would also work for VoIP, not just Mbone sessions. The result was an intensified effort towards completing the specification in late 1998, and completion by the end of the year. It received official approval as an RFC (Request for Comments, the official term for an IETF standard) in February and issuance of an RFC number, 2543, in March.

From there, industry acceptance of SIP grew exponentially. Its scalability, extensibility, and — most important — flexibility appealed to service providers and vendors who had needs that a vertically integrated protocol, such as H.323, could not address. Among service providers MCI (particularly MCI's Henry Sinnreich, regarded as the "Pope" of SIP) led the evangelical charge. Throughout 1999 and into 2000, it saw adoption by most major vendors, and announcements of networks by service providers. Interoperability bake-offs were held throughout 1999, attendance doubling at each successive event. Tremendous success was achieved in interoperability among vendors. Other standards bodies began to look at SIP as well, including ITU and ETSI TIPHON, IMTC, Softswitch Consortium, and JAIN.

Looking forward, 2000 will be a year in which real SIP networks are deployed, SIP vendors step forward to announce real products, and applications and services begin to appear.

PINT and SPIRITS – Instant Messaging

The use of SIP by other protocols is continuing at a frantic pace, and designers of many other application environments are looking at the modularity, scalability, and transparency of SIP for their own uses.

A strong case in point: call enabling e-commerce web pages with one-click call options to a call center. SIP is perfect for this environment. Since the SIP user agent is very lightweight, and is being imbedded in many phone-type devices, it is very easy to imbed a SIP Java client in a Java servlet on the web server, which can launch requests to create phone calls on the telephone network. This is the focus of the PINT (PSTN and Internet Interworking) protocol, based on SIP.

Another SIP application: Internet call waiting. A CO switch detects that you are online when a call comes in for your phone. Understanding this line condition, the switch gateways the call over IP using SIP and signals you, while you are still online, that you have a call coming in. You are prompted at your desktop to either (1) accept the call, (2) reject the call, or (3) send the call to voicemail. This application fits into the broader context of IP-PSTN cross signaling environments. Its standardization, based on SIP, is taking place in the SPIRITS (Service in the PSTN/IN Invoking Internet Service) working group in IETF.

Perhaps one of the most exciting new applications of SIP is as a protocol platform for instant messaging. A replacement for AOL Messenger? Absolutely. The mechanism by which it routes call invitations, and reaches the user wherever he is, is fundamentally about using presence (i.e., the communications state of a person) to enable session initiation. Turning this presence information around and delivering it to "buddies" is a natural extension of SIP. It also means that a single infrastructure can be used for voice, video, presence, and instant messaging – a very appealing goal. – *R.S., J.R.*

WHAT DOES IT DO?

As the name implies, the session initiation protocol (SIP) is about initiation of interactive communications sessions between users. SIP also handles termination and modifications of sessions as well. SIP actually doesn't define what a "session" is; this is described by content carried in SIP messages. Most of SIP is about the initiation part, since this is really the most difficult aspect. "Initiating a session" requires determining where the user to be contacted is actually residing at a particular moment. A user might have a PC at work, a PC at home, and an IP desk phone in the lab. A call for that user might need to ring all phones at once. Furthermore, the user might be mobile; one day at work, and the next day visiting a university. This dynamic location information needs to be taken into account in order to find the user.

Once the user to be called has been located, SIP can perform its second main function — delivering a description of the session that the user is being invited to. As mentioned, SIP itself does not know about the details of the session. What SIP does do is convey information about the protocol used to describe the session. SIP does this through the use of multipurpose internet mail extensions (MIME), widely used in web and e-mail services to describe content (HTML, audio, video, etc.). The most common protocol used to describe sessions is the session description protocol (SDP), described in RFC2327. SIP can also be used to negotiate a common format for describing sessions, so that other things besides SDP can be used.

Once the user has been located and the session description delivered, SIP is used to convey the response to the session initiation (accept, reject, etc.). If accepted, the session is now active. SIP can be used to modify the session as well. Doing so is easy — the originator simply re-initiates the session, sending the same message as the original, but with a new session description. For this reason, modification of sessions (which includes

Telcom Ventures Exhibit No. 2021
Samsung Elec. v. Telcom Ventures LLC
IPR2025-00974, U.S. Patent No. 10,674,432
Page 3 of 14

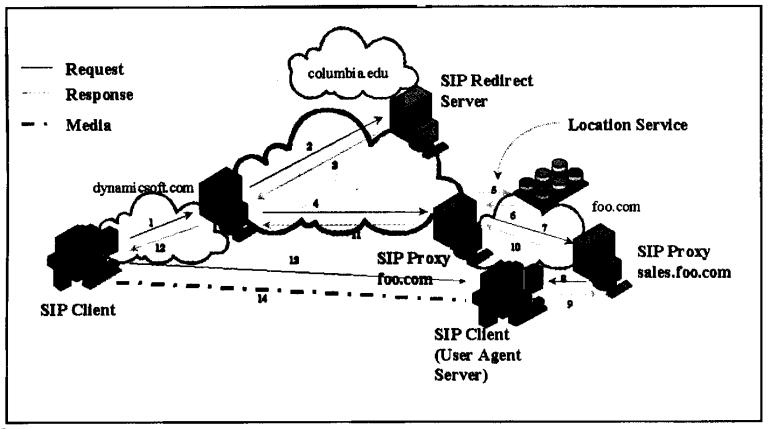


Figure 1: Session initiation in SIP.

things like adding and removing audio streams, adding video, changing codecs, hold and mute) are easily supported with SIP, so long as the session description protocol can support them (SDP supports all of the above).

Finally, SIP can be used to terminate the session (i.e., hang up).

HOW DOES IT WORK?

SIP is based on the request-response paradigm. To initiate a session, the caller (known as the User Agent Client, or UAC) sends a request (called an INVITE), addressed to the person the caller wants to talk to. In SIP, addresses are URLs. SIP defines a URL format that is very similar to the popular mailto URL. If the user's e-mail address is jdrosen@dynamicsoft.com, their SIP URL would be sip:jdrosen@dynamicsoft.com. This message is not sent directly to the called party, but rather to an entity known as a **proxy server**. The proxy server is responsible for routing and delivering messages to the called party. The called party then sends a response, accepting or rejecting the invitation, which is forwarded back through the same set of proxies, in reverse order.

A proxy can receive a single INVITE request, and send out more than one INVITE request to different addresses. This feature, aptly called "forking," allows a session initiation attempt to reach multiple locations, in the hopes of finding the desired user at one of them. A close analogy is the home phone line service, where all phones in the home ring at once.

Consider the scenario in Figure 1. In our example, the caller (jdrosen@dynamicsoft.com) wishes to place a call to joe@columbia.edu. Jdrosen sends his SIP INVITE message to the proxy for dynamicsoft.com (Step 1). This proxy then forwards the request out to Columbia, where it reaches the Columbia.edu server (Step 2). This server is actually not a proxy, but a similar device called a **redirect** server. Instead of forwarding calls, a redirect server asks the requestor to contact the next server directly. The Columbia.edu server looks up Joe in its database, and determines that today, Joe is on sabbatical to foo.com. It therefore sends a special response, called a redirect, to the dynamicsoft.com proxy, instructing it to instead try joe@foo.com (Step 3).

The dynamicsoft proxy then acts on this response, which means it directly tries to contact joe@foo.com. So, its sends the INVITE to the foo.com server (step 4). This server consults its database (Step 5), and learns (Step 6) that Joe is actually in sales. So, it constructs a new URL, joe@sales.foo.com, and sends the INVITE to the sales.foo.com proxy (Step 7).

The proxy for the sales department then needs to forward the INVITE to the PC where Jo is currently sitting. How does it know which PC Joe is at? SIP defines another request, called REGISTER, which is used to inform a proxy of an address binding. In this case, when Joe turned on his SIP client on his PC, the client would register the binding sip:joe@sales.engineering.com to sip:joe@mypc.sales.foo.com. This would allow the proxy to know that Joe is actually at mypc, a specific host on the network. The bindings registered through SIP are periodically refreshed, so that if the PC crashes, the binding is eventually removed.

The sales.foo.com proxy consults this registration database, and forwards the INVITE to joe@mypc.sales.foo.com (Step 8). This INVITE then reaches Joe at his PC. Joe can then respond to it (thus the request-response model). SIP provides many responses, and these include acceptance, rejection, redirection, busy, and so on. The response is forwarded back through the proxies to the original caller (Steps 9,10,11,12). An acknowledgement is sent (another type of request, called ACK) in Step 13, and the session is established. Media can then flow (Step 14).

SIP is patterned after HTTP in many ways. HTTP is also request-response. SIP borrows much of the syntax and semantics from HTTP. The textual message formatting, usage of headers, MIME support, and many headers are identical. An http expert looking at a SIP message would have difficulty distinguishing them.

MAIN ADVANTAGES

Services: Internet telephony began on the premise that it was cheaper than normal phone calling. Users were willing to tolerate degraded quality or reduced function for lower cost. However, the cost differentials are rapidly disappearing. To continue to exist, Internet telephony must find another reason to be. The answer is services.

Some of the most exciting applications have already found killer status on the Internet, though not (yet) in the form of multimedia services. Now think of integrating multimedia communications, such as

Telcom Ventures Exhibit No. 2021
Samsung Elec. v. Telcom Ventures LLC
IPR2025-00974, U.S. Patent No. 10,674,432
Page 4 of 14

SIP: A Carrier's Perspective

We've recently devoted a lot of attention to the rise of the session initiation protocol (SIP) for IP telephony signaling, and, we think, with good reason. The innovation that we've already seen emerge from SIP initiatives is only a taste of what we believe to be the protocol's potential. Nevertheless, what makes the most sense technologically is not always what wins in the marketplace. We were therefore pleasantly surprised to discover major telecom carriers to be as committed as the vendor community to the development and adoption of SIP.

MCI WorldCom, under the direction of Henry Sinnreich, Distinguished Member of Engineering, is now carrying out one of the most sophisticated implementations of SIP to date. MCI has already developed its own SIP redirect/proxy/location servers, and is working with other vendors in areas like softswitches and IP telephones.

"We chose SIP because it is completely integrated with other Internet and web protocols," notes Sinnreich. "With SIP, everything that has been learned from web services can now be applied to telephony."

SIP's technical elegance alone would not win carrier adoption, of course. "The idea is simple," says Sinnreich. "You invest money in new network infrastructure only if it can bring new revenue. And new revenue comes from services that don't exist today. On the web, we've already seen an explosion of new multimedia services. So if you can apply a set of integrated web protocols that would include telephony, the potential for more new services is almost obvious."

While "new services" is already a mantra in discussions of Internet telecom, some of the real "newness" that convergence promises has been obscured.

In the worst cases, the "next-gen" network starts to look almost identical to the "legacy" network. Sinnreich blames this dilution (and we agree) on carriers' and equipment vendors' desire to protect their traditional sources of revenue. This turf war, he says, is reflected at the protocol level.

"At this point, you basically have three camps. One is H.323, which has basically been pushed by big players in the computer industry who want to enter the communications business. My personal opinion is that H.323 has effectively already failed in the U.S., and is now starting to fail overseas, mainly because it is based on an ISDN architecture. The second camp is SIP, which is the pure Internet play. The third, which I consider to be the most dangerous and potentially negative, is those in the telecom industry who are trying to preserve circuit-switched control and signaling systems, using the Internet as just a cheaper wire. The call agent in the case of MGCP or MEGACO, for example, is essentially there to preserve the signaling and control architecture of SS7 and the Intelligent Network. The software, of course, is all proprietary, so that the vendor who implements the call agents and softswitches in the carrier's network now has control over that carrier's services."

TURF PROTECTION

Confusion surrounds the relation of protocols like the Media Gateway Control Protocol (MGCP) and MEGACO (its likely successor) to SIP, as they are complementary in certain ways and mutually exclusive in others. Sinnreich explains, however, "MGCP and MEGACO are great protocols for internally controlling an IP telephony gateway. But when they are used as control protocols for delivering services across the network, the carrier essentially puts his fate in the hands of the softswitch or call agent vendor. By limiting the user's access to a proprietary group of application servers, the MGCP or MEGACO call agent basically reinvents unequal access [of the PSTN], only now it's located at the control level, whereas it used to be at the transport level."

While some try to avoid taking such strong positions on VoIP protocols by claiming "protocol agnosticism," Sinnreich sees this notion as untenable. "The Internet and the web are nothing but a collection of protocols, which have been so well designed that now everybody can communicate with everybody else in an inexpensive and effective way. Executives sometimes ask me, 'Why all this talk about protocols, why should I be interested in protocols?' I have to explain that soon, the core of our business will be based on these protocols."

THE NEW SERVICE PARADIGM

Unlike MCI, startup carrier Level 3 entered the market unencumbered by an existing circuit-switched infrastructure. Level 3 also does not provide services directly to end users, but offers its network infrastructure to other service providers. Almost from day one, the carrier was committed to SIP. Matt Johnson, product marketing manager for Level 3's Dynamic Bandwidth Applications and Softswitch-Enabled Services Group, says that by pushing more intelligence out to the network endpoints, SIP can "generally simplify the systems required in the network."

Level 3 also takes a very strategic position on the softswitch's role in network infrastructure. According to Johnson, "The softswitch should behave basically as a web server behaves, providing a kind of 'dumb intelligence.'" In Level 3's network, softswitches are platforms that provide open interfaces to services, but do not themselves contain the service intelligence. SIP, because it facilitates point-to-point connections, is crucial to this model. Yet Level 3 sees SIP not just as a signaling protocol, but as an enabling technology for a fundamentally different way of delivering telecom services. Level 3's entire business model, in fact, is based on the idea of disaggregated services. "We're focused on carrying out an enabling strategy," Johnson explains. "We won't ever touch the end user directly with our network, but we will enable ASPs or ITSPs to provide innovative applications to the end user. In effect, we're turning the revenue model of traditional networks inside out, by giving customers access to the services they want from any number of different providers."

Telcom Ventures Exhibit No. 2021
Samsung Elec. v. Telcom Ventures LLC
IPR2025-00974, U.S. Patent No. 10,674,432
Page 5 of 14

voice, with web, e-mail, buddy lists, instant messaging, and online games. Whole new sets of features, services, and applications become conceivable.

SIP is ideally suited here. Its use of URLs, its support for MIME and carriage of arbitrary content (SIP can carry images, MP3s, even Java applets), and its usage of e-mail routing mechanisms, means that it can integrate well with these other applications. For example, it is just as easy to redirect a user to another phone as it is to redirect a user to a web page.

Scalability: SIP uses the Internet model for scalability — fast and simple in the core, smarter with less volume in the periphery. To accomplish this, SIP defines several types of proxy servers. "Call-stateful" proxies generally live at the edge of the network. These proxies track call state, and can provide rich sets of services based on this knowledge. Closer to the core, "transaction-stateful" (also known as just "stateful") proxies track requests and responses, but have no knowledge of session or call state. Once a session invitation is accepted, the proxy forgets about it. When the session termination arrives, the proxy forwards it without needing to know about the session.

Finally, "stateless" proxies exist in the core. These proxies receive requests, like INVITE, forward them, and immediately forget. The SIP protocol provides facilities to ensure that the response can be correctly routed back to the caller. Stateless proxies are very fast, but can provide few services. Call-stateful proxies are not as fast, but they live at the periphery, where call volumes are lower.

Extensibility: History has taught Internet engineers that protocols get extended and used in ways they never intended (e-mail and web are both excellent examples of this). So, they've learned to design in support for extensibility from the outset. SIP has numerous mechanisms to support extensions. It does not require everyone to implement the extensions. Facilities are provided that allow two parties to determine the common set of capabilities, so that a session initiation can always be completed, no matter what.

Flexibility: SIP is not a complete system for Internet telephony. It does not dictate architecture, usage patterns, or deployment scenario. It does not mandate how many servers there are, how they are connected, or where they reside. This leaves operators tremendous flexibility in how the protocol is used and deployed. One way to think of it is that SIP is a LEGO block; operators can piece together a complete solution by obtaining other LEGO blocks, and putting them together in the way that they see fit.

ENUM and TRIP

Although the standard method of connecting with SIP servers is to use an e-mail-like address such as sip:yourname@yourdomain.com, that will not be helpful if all you have is a telephone number. Though many people will use SIP directly from their PC, we still have billions of telephones in use that only have 12-key entry capability. We tend to think of phone calls as using phone numbers, so why should SIP (or any other form of Internet telephony) use a different form of identification? This is particularly important since the PSTN is not going to go away any time soon. Fortunately, SIP can easily carry phone numbers, using the new telephone URL (tel:5551212, for example).

The question is — if all you have is a telephone number, how do you find a SIP resource (or any other resource for that matter) on the Internet that is associated with that telephone number? When the SIP address is an e-mail-like identifier, the resource is easily found through a DNS query, since the e-mail-like identifier contains a domain name. The problem is harder for phone numbers.

Addressing this problem is the task that the IETF ENUM WG (Telephone Number Resolution Working Group) has set for itself. This work is in an advanced stage of development, so we can speak with some confidence on the shape of the ultimate output, and its implications for SIP and Internet telephony in general.

Telephone numbers are well understood. Their definition has been the work of the ITU (International Telecommunications Union). The formatting and structure of telephone numbers is defined in recommendation E.164. These numbers can be no larger than 15 digits and are globally unique.

The ENUM plan is to enter telephone numbers into the Internet DNS [domain name system] so that any application, including SIP, can discover resources available to that globally unique phone number.

The technique would work something like this. Take a phone number [1-212-691-8215] and translate it into a format that the DNS system can understand, such as 5.1.2.8.1.9.6.2.1.2.1.e164.foo. (Don't worry. As a SIP user you will never have to figure out how to reverse type in phone numbers. Your SIP phone or SIP proxy will do this for you automatically.)

Each and every digit to the left of the domain is a zone in DNS terms, and authority for zones can be delegated at each digit. This permits both individuals and enterprises the ultimate right to decide what Internet services are available for this number.

If, perhaps, you have plugged a SIP phone into your network, you simply dial a phone number as you always have. The SIP phone or proxy server would do the number domain translation and through classic DNS resolution discover a DNS Resource Record that essentially says 1-212-691-8215 can be reached by SIP by contacting sip:main.number@computertelephony.com. Your SIP UA would then follow standard SIP procedures, and call the user manning the main number at the *Computer Telephony* offices.

Interestingly, the ENUM mechanism gives even wider flexibility than this. Instead of containing just SIP URLs, the DNS entries can contain e-mail addresses for VPIM-based universal messaging or LDAP-based white page resources for advanced caller identification. The DNS entries could even contain H.323 addressing information. In fact, any URL can be placed inside, allowing clients to be contacted using a variety of communications mechanisms.

ENUM depends, of course, on a particular phone number having some kind of resource on the Internet associated with it. However, the vast majority of phone num-

Telcom Ventures Exhibit No. 2021
Samsung Elec. v. Telcom Ventures LLC
IPR2025-00974, U.S. Patent No. 10,674,432
Page 6 of 14

bers correspond to simple PSTN phones. These numbers will have no corresponding entries in the DNS. To contact them, a SIP phone must route a call to a telephone gateway, which connects to the PSTN. Normally, a service provider does not own all the gateways that can be used to terminate a call from one of its users. It enters into peering relationships, through clearinghouses or settlement organizations, to have access to a wider set of gateway services. That being the case, how does a service provider know which of its peers has gateways that can terminate a call to a particular number? This problem calls for some kind of routing protocol that allows service providers to exchange routes.

That is the idea behind the TRIP (telephony routing over IP) Protocol, being developed by the IPTEL working group in the IETF. Based on the well-known border gateway protocol(BGP), TRIP servers (often collocated with SIP proxies) maintain and exchange information on what gateways are available to establish calls to ranges of telephone numbers. Like BGP, TRIP provides support for aggregation, scalability, and reliability. Gateway failures are quickly detected, and alternates are used instead. TRIP permits multiple service providers to route calls through each others' gateways, thus optimizing the capital cost of each provider in a particular service region, and avoiding unnecessary duplication or over-provisioning of gateways.

SIP SERVICE PROVISIONING

Beyond SIP's ability to create new, innovative services, the protocol has the potential for making service creation available to the "masses," in much the same way as web content. These services can be created by service providers, enterprise administrators, and IT departments, or even directly by the end users. By opening up innovation to the public at large, all sorts of new services and features can be developed, creating entire new markets. We have seen the same thing in the web space — the accessibility of the web has fostered a huge market for e-commerce, since anyone with an idea and a few dollars can put together new content. In overcoming the highly centralized and carrier-controlled model of telephony, and in putting tools for service creation in so many hands, SIP has here the potential to deliver what the PSTN Intelligent Network only promised.

What kind of services or applications could be enabled by SIP? Besides the traditional call-forwarding, follow-me, and do-not-disturb, SIP has the potential for enabling a whole new class of services that integrate multimedia with web, e-mail, instant messaging, and "presence" (meant here as, "are you currently online?"). The value that the Internet brings to Internet telephony is the suite of existing applications that can be merged with voice and video communications. As an example, at the end of a call, a user can transfer the other party to a web page instead of another phone. This transfer would end the call, and cause the other party's web browser to jump to the new page. In essence, the value of VoIP and SIP comes not from integration at the network layer (i.e., run your voice services on top of your data network), but at the *services* layer (i.e., combine your voice services with your data services).

HOW TO PROGRAM IT?

Developing services, of course, requires APIs. What kind of APIs are used to program services delivered by SIP? There has been significant activity in this area, resulting in numerous new interfaces, each with its own distinct set of strengths and weaknesses.

The first API that surfaced is the call processing language (CPL). CPL is not actually an API, but rather an XML-based scripting language for describing call services. It is not a complete programming language, either. It has primitives for making decisions based on call properties, such as time-of-day, caller, called party, and priority, and then taking actions, such as forwarding calls, rejecting calls, redirecting calls, and sending e-mail. CPL is engineered for end-user service creation.

A server can easily parse and validate a CPL, guarding against malicious behavior. The running time and resource requirements of a CPL can also be computed automatically from the CPL. An interpreter for CPL is very lightweight, allowing CPL services to execute very quickly. For these reasons, it is possible for an end user to write a CPL (typically with some kind of GUI tool), upload it to the network, and have it instantly verified and instantiated in real time.

At the opposite end of the spectrum in SIP is CGI (the common gateway interface). Many web designers are familiar with HTTP CGI; it's an interface that allows people to generate dynamic web content using Perl, Tcl, or any other programming language of choice. Since HTTP and SIP are so similar, it was recognized that an almost identical interface could be used for SIP. The result is SIP CGI, which is roughly 90% equivalent to HTTP CGI.

Like HTTP CGI, SIP CGI passes message parameters through environment variables to a script that runs in a separate process. The process sends instructions back to the server through its standard output file descriptor. The benefit of SIP CGI is that it makes development of SIP services work much like the creation of dynamic web content. In fact, for SIP services that contain substantial web components, development will closely mirror web-only services. The importance of leveraging web tools for voice service creation is that a much larger class of developers becomes available.

CGI has substantially more flexibility than CPL (CGI doesn't even mandate a particular programming language), but is much more risky to execute. Furthermore, because of its usage of separate processes, SIP CGI doesn't scale as well as CPL.

Somewhere in the middle are SIP Servlets. HTTP Servlets are in wide use for developing dynamic web content. Servlets are very similar to the CGI concept. However, instead of using a separate process, messages are passed to a class that runs within a JVM (Java Virtual Machine) inside of the server. As a result, Servlets are re-

Telcom Ventures Exhibit No. 2021
Samsung Elec. v. Telcom Ventures LLC
IPR2025-00974, U.S. Patent No. 10,674,432
Page 7 of 14

stricted to Java, but suffer less overhead than SIP CGI. Use of a JVM for executing servlets means that the Java "sandbox" concept can be applied to protect the server from the script. Like SIP CGI, SIP Servlets closely mirror the operation of HTTP Servlets; they simply enhance the interface to support the wider array of functions a proxy can execute, as compared to an HTTP origin server.

Besides these new APIs, traditional APIs (such as JAIN, Parlay, JTAPI, and TAPI) can also be used to develop SIP services. However, the telephony focus of these APIs restricts them from performing services that take advantage of SIPs unique capabilities, such as integration with web and e-mail.

QUALITY OF SERVICE

Perhaps the most vexing problem in voice-over-IP, in general, has been the issue of quality of service. The delay in conversations that many VoIP users encounter is caused by the jitter and latency of packet delivery within the Internet itself. It's useful to review some of the basic principles of the Internet to understand what can be done about the problem, what the IETF's response has been, and how it impacts SIP.

Currently, the Internet offers a single service, traditionally referred to as "best effort." In other words, all packets are created equal. There is no difference to the Internet whether a packet is e-mail, FTP, or the download of a web page. If the Internet gets very busy, packets get dropped or delayed.

Unfortunately, the human ear is extremely sensitive to latency in the delivery of sound. The human ear can detect delays of 200 milliseconds or greater in voice conversations.

SIP itself does not get involved in reservation of network resources or admission control. This is because SIP messages may not even run over the same networks that the voice packets traverse. The complete independence of the SIP path and the voice path enables ASPs to provide voice services without providing network connectivity. This is an extremely important advantage of the SIP architecture. Given this, SIP relies on other protocols and techniques in order to provide quality of service.

Most users have dealt with QoS issues by either adding bandwidth to their networks, or by applying complex and expensive framing techniques, such as ATM, to IP traffic. This may be sensible for intra-enterprise VoIP configurations, since the network can be administered directly. However, when Internet traffic must exit a domain or a particular carrier boundary, all bets are off; other methods must be used.

To create QoS on the Internet, you must create different classes of service for packets. The IETF has taken two approaches: The first is Integrated Services (RFC2211 and RFC2212), also known as INTSERV. The second is Differentiated Services (RFC2475), or DIFFSERV. Describing these two techniques is another article in itself, but they can be summarized.

INTSERV essentially creates an end-to-end private lane for packet voice traffic that is opened and monitored by each router along the path and the endpoints. No packets are sent out unless the entire route signals its ability to meet and guarantee the service requirements for the call. The protocol used to reserve the resources in the network, and get confirmation of those resources, is known as RSVP, or the resource reservation protocol (RFC2205).

DIFFSERV creates classes of service, and controls the admission of that traffic onto the Internet, by filtering packets at the edge of the network. Here, there are no explicit requests for resources from the network. The advantage to the DIFFSERV approach is that it does not require the maintenance of network state by all elements, and thus scales better than RSVP.

Clearly, the INTSERV approach offers the highest level of quality for sensitive applications, such as voice. For SIP, this means the transparent integration of two forms of signaling: first, the signaling to set up the call (using SIP) and — once the media addresses and codecs are agreed upon — the second, for setting up QoS using RSVP.

This separation of session establishment and QoS reservation introduces an interesting side effect: One may succeed (namely, the call setup), while the other (resource reservation) can fail. The result is that the phone may ring and be answered, even though the network cannot support the call. To handle this problem, a coupling mechanism has been developed for SIP based on work done initially within the PacketCable Forum Distributed Call Signaling (DCS) group. This coupling allows the SIP INVITE (specifically, the SDP), to contain indicators that tell the called user not to "ring the phone" until sufficient resources have been reserved (using RSVP or some other mechanism). Once the reservations have succeeded, the caller sends a new request, tentatively dubbed "PRECONDITIONS_MET," to the called user, indicating that resources are available, and the phone should ring. Of course, if the QoS reservation fails, the call can optionally proceed with best effort.

This means that SIP systems can make use of comprehensive end-to-end QoS models for Internet telephony. Since SIP itself does not specify those mechanisms, new and more comprehensive QoS services that are discovered can be used without affecting SIP.

RELATED PROTOCOLS COMPARED:

Admittedly, SIP is hardly the only protocol in the VoIP space. Two others are closely related to SIP — H.323 and MGCP. (We group IPDC, SGCP, Megaco, and H.248 together with MGCP for purposes of this discussion).

H.323

The ITU developed H.323. Version 1 was standardized in 1996. Its focus was multimedia communications services for LANs without QoS. H.323 v.1 was not targeted for IP specifically, but rather for any type of packet LAN. It had numerous shortcomings for Internet telephony, some of which were addressed in Version 2, released in 1998. Version 3 has been recently completed, and Version 4 is now under development.

Telcom Ventures Exhibit No. 2021
Samsung Elec. v. Telcom Ventures LLC
IPR2025-00974, U.S. Patent No. 10,674,432
Page 8 of 14

H.323 is a complete, vertically integrated suite of protocols and an architecture for delivering multimedia conferencing applications. It includes signaling, registration, admission control, security, interworking requirements with H.320, H.321, and other ITU conferencing systems, inter-domain data exchange, transport, and codecs. H.323 defines several entities, including terminals (end systems, like PCs), gateways, multipoint conferencing units, and something called a gatekeeper. A gatekeeper is similar to a SIP proxy, in that it plays the role of a signaling relay.

There are numerous differences between SIP and H.323. The first is scope; H.323 specifies a complete, vertically integrated system. Not much room is left for flexibility or different architectures. SIP, on the other hand, is a single component. It works with RTP, for example, but does not mandate it. SIP systems can be composed into a variety of architectures, and numerous protocols and additional systems can be plugged in at the discretion of the service provider. SIP can be considered a building block, whereas H.323 is a specific system.

The flip side of this determinism is that H.323 does numerous things that SIP, purposefully, does not address. For example, one of the H.323 protocols, H.245, contains powerful mechanisms for conference control for distributed multiparty conferences. This conference control allows a chairman to grant or deny speaking privileges to other conference participants. This kind of control is possible within a SIP-established conference, but it is not addressed by SIP itself, and there are currently no standalone standard protocols that can do this.

H.323 has its origins as a LAN protocol; numerous enhancements (such as FastStart) were added to address usage as a wide-area protocol. SIP, in contrast, was designed from day one as a wide-area protocol. SIP's support for fast, stateless proxies in the core, and call stateful proxies in the periphery, adds significant scalability here. Furthermore, its ability to pass data to clients, and have it reflected back, means that state can be pushed to the periphery (several headers, such as Via and Record-Route, provide that capability). These are absent in H.323.

Many of the other differences stem, in general, from the different histories and design philosophies of the protocols' parent organizations. ITU's H.323 borrows its call-signaling component from existing work done in ITU, namely the Q.931 protocol, used for user-to-network signaling in ISDN.

SIP, on the other hand, borrows much of its concepts from HTTP, also developed within IETF. The result is that H.323 has much more of a telephony-centric flavor, while SIP has more of a web flavor. Intimately tied to this are SIP's facilities, which allow it to integrate with web, e-mail, and other existing IP applications. Wherever SIP has a field where a SIP URL can appear, any other URL type can be present. SIP's powerful and extensive URL definition allows for SIP URLs to be embedded in web browsers and e-mail tools. H.323 has no URL format. Since applications that integrate web, e-mail, buddy lists, IM, and other IP applications with voice are likely to be the killer app for VoIP, this functionality is critical.

SIP's allegiance to the KISS (Keep It Simple Stupid) principle has made it generally easier to implement and interoperate. To illustrate: H.245 has another set of functions that allow for very powerful multimedia capability negotiation between participants. This negotiation allows each side to convey sets of capabilities, and to describe interactions between them. ("I can do G.723.1 compression with H.263 but only without B-frames, but I can do G.729 with H.261.")



Telcom Ventures Exhibit No. 2021
Samsung Elec. v. Telcom Ventures LLC
IPR2025-00974, U.S. Patent No. 10,674,432
Page 9 of 14

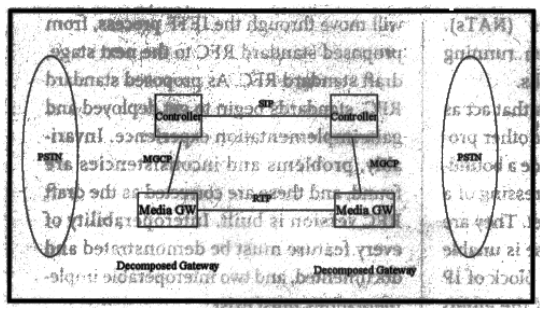


Figure 2: MGCP and SIP Used Together.

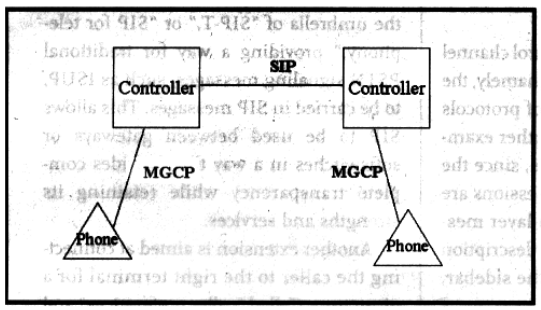


Figure 3: MGCP for phone control.

SIP (actually, SDP) negotiates much simpler capabilities — just a list of codecs, in order of preferences, for a particular media stream. Experience has shown that in practice, this is sufficient.

Another important difference is extensibility. Both protocols can be extended, but differ significantly in method. H.323 defines protocol elements called nonStandardParams, which are sprinkled throughout the protocol. Vendors can add non-standard elements, identified by a vendor ID. H.323 can also be extended through a version change. Versions must be completely backward compatible, ensuring that each version of the protocol takes up more room than its predecessor.

SIP, on the other hand, allows for standards-based extensions to perform specific functions. Implementations can implement different sets of extensions depending on their needs. SIP takes care of interoperability in this environment by providing a powerful set of tools for indicating and negotiating the set of extensions used in requests and responses.

SIP can be extended in numerous ways, including adding headers, new methods, new bodies, and parameters to existing headers. Therefore, it is less restrictive than H.323 in where new things can be placed. The history of the Internet has shown that protocols get used in ways never intended; that extensions get developed, used, and sometimes fade away. Support for this kind of evolution is critical for long-term success.

MGCP

MGCP (and its relatives) was conceived as a tool for decomposing a telephony gateway into a controlling signaling component and a controlled media component. The protocol allows the controller to instruct the controlled to send and receive media from specific addresses, generate tones, and modify configuration. The protocol also sees that the controlled entity reports back to the controller, when detecting DTMF digits and tones, for example.

MGCP performs a much different function than SIP does. In fact, a complete system cannot be built with MGCP alone. A session initiation protocol is still needed between separate controllers. This is illustrated in Figure 2.

MGCP can also control an IP phone (See Figure 3). The controller acts as a PBX or Class 5 switch, providing every possible service to the phone: receiving off-hook signal, instructing the phone (which is the controlled media gateway) to send dialtone, receiving dialed digits, and launching a SIP INVITE to connect the call. When the response comes, the controller tells the phone to send media to the appropriate location, and to receive media and play it out the speakers.

In this application, MGCP "competes" with SIP in the sense that instead of MGCP, SIP could have been placed in the phone.

There is a substantial difference in these two approaches. The MGCP approach assumes the phone is nothing more than a dumb black phone with the standard twelve buttons. The system cannot support any services beyond those traditional PSTN services. This is not surprising, since the architecture closely mirrors the way residential phone service is provided today. Services are pushed back into the network, even if they don't necessarily need to be there.

Some argue that pushing services back into the network enhances service availability. However, an MGCP phone cannot support any of the advanced and new features that the Internet can bring. Furthermore, a user cannot even make a phone call without a controller providing the service. In both SIP and H.323, users can call each other without proxies or gatekeepers. Some argue

SIP URLs

Henning Shultzerinne's Ultimate SIP resource: www.cs.columbia.edu/~hgs/sip/
Unofficial SIP WG Homepage: www.softarmor.com/sipwg/
ENUM Charter: **ww.ietf.org/html.charters/enum-charter.html**
ENUM Draft #1: search.ietf.org/internet-drafts/draft-faltstrom-e164-05.txt
ENUM Draft #2: search.ietf.org/internet-drafts/draft-vaudreuil-enum-e164dir-00.txt
DNS Zone Redirection: www.rfc-editor.org/rfc/rfc2672.txt
IPTEL Charter: www.ietf.org/html.charters/iptel-charter.html
TRIP Home page: www.bell-labs.com/mailing-lists/iptel/
PINT: www.ietf.org/html.charters/pint-charter.html
SPIRITS: www.ietf.org/html.charters/spirits-charter.html

Telcom Ventures Exhibit No. 2021
Samsung Elec. v. Telcom Ventures LLC
IPR2025-00974, U.S. Patent No. 10,674,432
Page 10 of 14

that this is strength of this architecture, not a weakness. However, the Internet is what it is — and that is a medium that allows any two users to contact each other, independent of the application. Forcing communications through some entity without providing value just means that end systems will be upgraded with software that has no such restriction (like a SIP client).

INTEROPERABILITY

Since SIP was approved as an RFC in March 1999, four interoperability bakeoffs have been held. Only ten or so implementations were tested at the first bakeoff; 45 were tested at the most recent one, in April. These implementations represent a wide cross section of the industry, including large gateway vendors such as Cisco; standalone phones such as Pingtel; proxy vendors such as dynamicsoft; PC clients such as Netspeak; service providers such as MCI; and protocol analyzers such as Agilent (formerly Hewlett Packard).

Success at these events has been astronomical. At the third bakeoff, a complex testing scenario included seven elements (three proxies and four user agents) from seven different vendors. The call was originated from a user agent, and sent to one of the proxies. The proxy provided a cryptographic challenge to the originator, who then retried the call setup automatically with security credentials. The credentials were verified, and the INVITE was forked to two proxies. One of those forked once more, causing three phones to ring simultaneously. One of them answered, causing the others to stop ringing. The scenario also involved a TCP-to-UDP conversion, and a complex application of SIP's Record Routing capability. This scenario was successfully executed several times during the event.

OK, WHAT'S THE BAD NEWS....

If SIP sounds too good to be true ... well, you know the saying. Emerging issues in the Internet could ruin the promise of SIP (as well as H.323) over the long term. The problem is the increasing shortage of IP v4 numbers and the growing use of network address translators (NATs). There are similar issues when running SIP and H.323 through firewalls.

NATs break many protocols that act as establishment mechanisms for other protocols, such as SIP. NATs provide a boundary between the private IP addressing of a network and the public Internet. They are most often used if an enterprise is unable to secure access to a sufficient block of IP numbers from their ISP, or if the enterprise wants the presumed luxury of being able to switch ISPs without having to renumber their network.

SIP, fundamentally, is a control channel for establishing other sessions (namely, the media sessions). These kinds of protocols (of which FTP and H.323 are other examples) cause problems for NATs, since the addresses for the established sessions are in the body of the application layer messages, as we see in the session description protocol examples shown in the sidebar, "SIP Call Flow Examples."

When used with SDP, SIP messages carry the IP addresses and ports to be used for the media sessions. There may be multiple media sessions within a particular SIP call. Since SDP carries IP addresses and not host names, the external caller user agent will send media to an IP address that is not globally routable. It is only a valid IP address within the private network.

A nearly identical problem exists for firewalls. When a user inside the firewall sends media to an address outside the firewall, it will be dropped by the firewall unless a rule is established to allow it to pass. Since the media is sent on dynamic ports to dynamic addresses, these rules must be dynamically installed through application-aware devices, such as proxies.

There is no easy solution to the problem of NATs. Anyone considering VoIP deployments should be aware of the limitations they present. A "Birds of a Feather" (BoF) session was held at the 47th IETF in Adelaide, Australia, called FOGLAMPS, to investigate solutions to this problem.

FUTURE DIRECTIONS

Where does SIP go from here? SIP itself will move through the IETF process, from proposed standard RFC to the next stage, draft standard RFC. As proposed standard RFC, standards begin to get deployed and gain implementation experience. Invariably, problems and inconsistencies are found, and these are corrected as the draft RFC version is built. Interoperability of every feature must be demonstrated and documented, and two interoperable implementations must exist.

Numerous extensions are also under development. Several of these fall under the umbrella of "SIP-T," or "SIP for telephony," providing a way for traditional PSTN signaling messages, such as ISUP, to be carried in SIP messages. This allows SIP to be used between gateways or softswitches in a way that provides complete transparency while retaining its strengths and services.

Another extension is aimed at connecting the caller to the right terminal for a given user. Called "caller preferences" and "callee capabilities," the extension incorporates presence information (i.e., buddy lists) into SIP messages. A caller can request a call for joe@company.com to be routed to Joe's mobile phone, or to a PC client that supports video.

Much work is also in progress on supporting infrastructure. This includes an SNMP Management Information Base (MIB) for management of SIP servers, gateways, and clients, DHCP option codes for SIP to support autoconfiguration, and QoS mechanisms that permit SIP sessions to take place only if sufficient bandwidth exists to support them. ✲

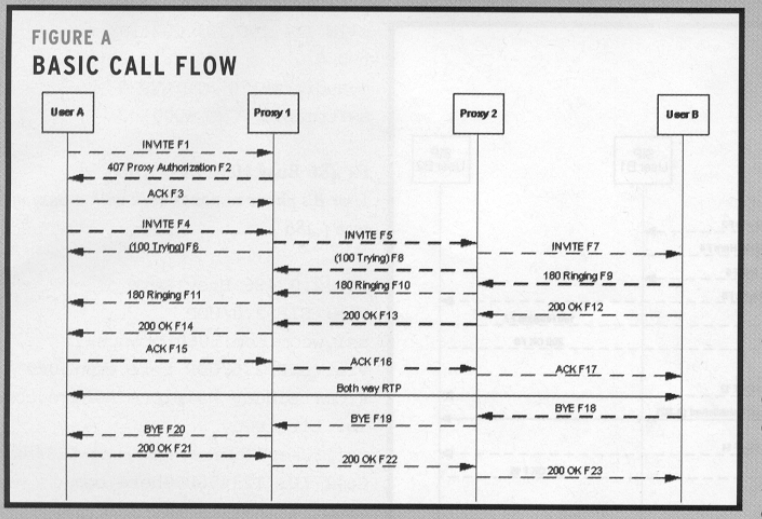
Jonathan D. Rosenberg is the Chief Scientist of dynamicsoft (www.dynamicsoft.com). He is the co-author of SIP (RFC2543), the chair of the IETF IPTEL work group, and the co-chair of the IETF SIP work group.

Richard Shockey is the principal of Shockey Consulting LLC. He is the co-chair of the IETF ENUM work group and chair of the IETF BOF work on QUALDOCS. He is also senior technical industry liaison at NeuStar, Inc.(www.neustar.com).

Telcom Ventures Exhibit No. 2021
Samsung Elec. v. Telcom Ventures LLC
IPR2025-00974, U.S. Patent No. 10,674,432
Page 11 of 14

SIP Call Flow Examples

FIGURE A
BASIC CALL FLOW



BASIC CALL FLOW

In Figure A, Caller A completes a call to User B using two proxies: Proxy 1 and Proxy 2. The initial INVITE (F1) does not contain the Authorization credentials that Proxy 1 requires, so an Authorization response is sent containing the challenge information.A new INVITE (F4) is then sent containing the correct credentials and the call proceeds. The call terminates when User B disconnects by initiating a BYE message.

F1 INVITE A -> Proxy 1

The call begins, as always, with an INVITE message that contains information on caller and called party as well as the session description request (2nd part).

```
INVITE sip:UserB@ss1.wcom.com SIP/2.0
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 12345600@here.com
CSeq: 1 INVITE
Contact: BigGuy <sip:UserA@here.com>
Content-Type: application/sdp
Content-Length: 147

v=0
o=UserA 2890844526 2890844526 IN IP4 here.com
s=Session SDP
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

F2 407 Proxy Authorization Required Proxy 1 -> User A

SIP always works in a request-response mode and in this example Proxy 1 challenges Caller A for authentication

```
SIP/2.0 407 Proxy Authorization
Required
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here
.com>
To: LittleGuy <sip:UserB@there
.com>
Call-ID: 12345600@here.com
CSeq: 1 INVITE
Proxy-Authenticate: Digest
realm="MCI WorldCom SIP",
domain="wcom.com",
nonce="wf84f1ceczx41ae6cbe5aea9c8e88d359",
opaque="", stale="FALSE", algorithm="MD5"
Content-Length: 0
```

As we move further down the call flow, the actual voice call begins , using Realtime Transport Protocol (RTP) to move the voice stream.

F17 ACK Proxy 2 -> B

```
ACK sip: UserB@there.com SIP/2.0
Via: SIP/2.0/UDP ss2.wcom.com:5060
Via: SIP/2.0/UDP ss1.wcom.com:5060
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>;tag=314159
Call-ID: 12345601@here.com
CSeq: 1 ACK
Content-Length: 0
```

Calls are then terminated with a BYE request to the caller.

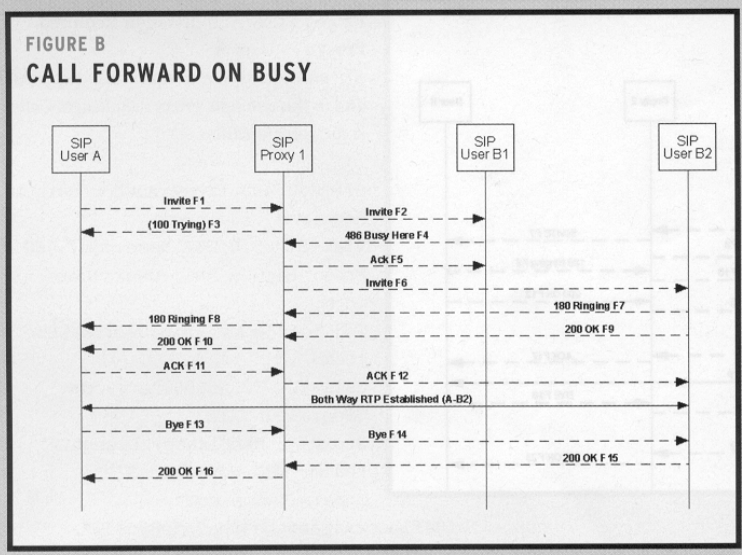
F18 BYE User B -> Proxy 2

```
BYE sip: UserA@ss2.wcom.com SIP/2.0
Via: SIP/2.0/UDP there.com:5060
Route:
<sip:UserA@ss1.wcom.com>,<sip:UserA@here.com>
From: LittleGuy
<sip:UserB@there.com>;tag=314159
To: BigGuy <sip:UserA@here.com>
Call-ID: 12345601@here.com
CSeq: 1 BYE
Content-Length: 0
```


Telcom Ventures Exhibit No. 2021
Samsung Elec. v. Telcom Ventures LLC
IPR2025-00974, U.S. Patent No. 10,674,432
Page 12 of 14

SIP Call Flow Examples

FIGURE B
CALL FORWARD ON BUSY



CALL FORWARD ON BUSY

In this scenario User B wants calls forwarded to another destination if the original line is busy. It is assumed that the proxy knows where to forward the call.

F1 INVITE A -> Proxy

```
INVITE sip:UserB@ss1.wcom.com SIP/2.0
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 12345600@here.com
CSeq: 1 INVITE
Contact: BigGuy <sip:UserA@here.com>
Proxy-Authorization: DIGEST username="UserA",
realm="MCI WorldCom
SIP", nonce="9137d175c20a0d6eadd7be1c863302ae",
opaque="",
uri="sip:ss1.wcom.com",
response="cf25aad811c806bde46a369220158cec"
Content-Type: application/sdp
Content-Length: ...

v=0
o=UserA 2890844526 2890844526 IN IP4
client.here.com
s=Session SDP
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

F4 486 Busy Here B1 -> Proxy

User B's phone responds back with a busy message (486)

```
SIP/2.0 486 Busy Here
Via: SIP/2.0/UDP
ss1.wcom.com:5060;branch=1
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy
<sip:UserB@there.com>;tag=123456
Call-ID: 12345600@here.com
CSeq: 1 INVITE
Content-Length: 0
```

F6 INVITE Proxy -> B2

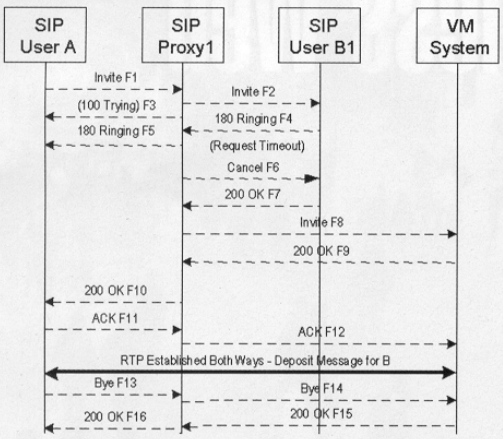
The call is then forwarded to a new location at "THERE.COM." Since the call is going to a new location, a new INVITE and session description is sent.

```
INVITE sip:UserB2@ there.com SIP/2.0
Via: SIP/2.0/UDP ss1.wcom.com:5060;branch=2
Via: SIP/2.0/UDP here.com:5060
Record-Route: <sip:UserB@ss1.wcom.com>
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 12345600@here.com
CSeq: 1 INVITE
Contact: BigGuy <sip:UserA@here.com>
Content-Type: application/sdp
Content-Length: ...

v=0
o=UserA 2890844526 2890844526 IN IP4
client.here.com
s=Session SDP
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```


Telcom Ventures Exhibit No. 2021
Samsung Elec. v. Telcom Ventures LLC
IPR2025-00974, U.S. Patent No. 10,674,432
Page 13 of 14

FIGURE C
FORWARDED ON NO ANSWER



FORWARDED ON NO ANSWER

Here User A attempts to call User B, who does not answer. The call is forwarded to User B's mailbox, and the voicemail system plays a message for a ring-no-answer. The call flow then moves from User A to the voicemail system.

F1

```
INVITE sip:UserB@wcom.com SIP/2.0
Via: SIP/2.0/UDP here.com:5060
From: TheBigGuy <sip:UserA@here.com>
To: TheLittleGuy <sip:UserB@wcom.com>
Call-Id: 12345600@here.com
CSeq: 1 INVITE
Contact: TheBigGuy <sip:UserA@here.com>
Proxy-Authorization:Digest username="UserA",
realm="MCI WorldCom SIP",
nonce="ea9c8e88df84f1cec4341ae6cbe5a359",
opaque="",
uri="sip:UserB@wcom.com", response=<appropriately
calculated hash goes here>
Content-Type: application/sdp
Content-Length: <appropriate value>

v=0
o=UserA 2890844526 2890844526 IN IP4
client.here.com
s=Session SDP
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Here B1 rings for, let's say, nine seconds. (This duration is a configurable parameter in the Proxy Server.) The Proxy sends a Cancel option and proceeds down its internal list of options, eventually selecting a voicemail URL for "forward no answer."

```
SIP/2.0 180 Ringing
F5 Via: SIP/2.0/UDP here.com:5060From:
TheBigGuy <sip:UserA@here.com>
To: TheLittleGuy
<sip:UserB@wcom.com>;tag=3145678
Call-Id: 12345600@here.com
CSeq: 1 INVITE
Content-Length: 0
```

Here the voicemail system is ready to set up a RTP session, and record the message by using the 200 OK signal. The 'contact' field indicates where the file is being stored.

```
F9  SIP/2.0 200 OK
Via: SIP/2.0/UDP wcom.com:5060; branch=2
Via: SIP/2.0/UDP here.com:5060
Record-Route: <sip:UserB@wcom.com>
From: TheBigGuy <sip:UserA@here.com>
To: TheLittleGuy
<sip:UserB@wcom.com>;tag=123456
Call-Id: 12345600@here.com
CSeq: 1 INVITE
Contact: TheLittleGuyVoiceMail <sip:UserB-dep-
fna@vm.wcom.com>
Content-Type: application/sdp
Content-Length: <appropriate value>

v=0
o=UserB 2890844527 2890844527 IN IP4
vm.wcom.com
s=Session SDP
c=IN IP4 110.111.112.114
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Finally User A hangs up on the voicemail system ...but the VM system could have disconnected the call on its own.

```
F13 BYE sip:UserB@wcom.com SIP/2.0
Via: SIP/2.0/UDP here.com:5060
Route:<sip: UserB-dep-fna@vm.wcom.com>
From: TheBigGuy <sip:UserA@here.com>
To: TheLittleGuy
<sip:UserB@wcom.com>;tag=123456
Call-Id: 12345600@here.com
CSeq: 2 BYE
Content-Length: 0
```


Telcom Ventures Exhibit No. 2021
Samsung Elec. v. Telcom Ventures LLC
IPR2025-00974, U.S. Patent No. 10,674,432
Page 14 of 14