A Look at Session Initiation Protocol

By Ken Camp

As a follow-up to recent online conversations, this article begins an exploration of common VoIP protocols. This first installment of an ongoing exploration of the technical workings of VoIP protocols will explore Session Initiation Protocol (SIP). As this is the first paper in the series, it will begin with an overview of how the open standards widely used in the Internet are developed.

Standards used in the worldwide Public Switched Telephone Network (PSTN) are products of the International Telecommunications Union-Telephony (ITU-T) sector, formerly known as the CCITT. This group operates under the auspices of the United Nations (UN), which is important to note because this body acts in many ways as an administrative unit focused more on international interoperability than other areas. Although this group has responsibility for telephony standards, they have not historically been known for being quick or nimble at responding to technology needs. ISDN standards took 10 years to get through this group. There are many different international political agendas that come into play when dealing with the UN, and change takes time.

Internet standards are a much different story. TCP/IP standards used throughout the Internet are primarily the work of the Internet Engineering Task Force (IETF). This group is also global—primarily an organization of volunteers. Any interested party can join the IETF and participate in standards development, and many do every year. The IETF working groups are made up of technology specialists from colleges and universities, hardware and software vendors, telecommunications providers and competitive local exchange carriers (CLECs), Internet service providers (ISPs) and Internet telephony service providers (ITSPs), government organizations, and a variety of other interested parties. Because the organization is voluntary and created by technologists focused on progress and efficiency, the structure of developing IETF standards is much different than that for the PSTN.

Internet standards are developed through the Request for Comments (RFC) process, which is quite efficient. In many cases, interested individuals or organizations collaborate to jointly present a draft for a new open standard to improve the network or enhance a capability. A proposal for a new standard that is jointly presented by team collaboration can carry broad support at introduction. Vendors will often work closely together to “grease the wheels” and move standards proposals forward. Thus, in the real world of standards development, change can often occur very quickly in Internet standards, as they move through the draft process.

In telecommunications, between the PSTN and the Internet, there are two different networks, with technical standards developed and approved by completely separate standards bodies. Although this idea might seem straightforward, it often isn’t—whether it’s the PSTN or the Internet, people often draw the network as a cloud. Although this is done for the sake of simplicity, the cloud concept breaks down when complex problems related to open standards and vendor interoperability are concerns. When you look at VoIP technologies, interoperability with the incumbent voice network, the PSTN, is absolutely necessary for global success and widespread deployment of IP telephony solutions.

Figure 1 shows three iterations of the convergence between the PSTN and the Internet. At the top is the beginning stage. The PSTN and Internet began as distinctly separate networks, each with a specific purpose. There was no connection or interoperability required between the two. This model represents the relationship between the two networks throughout the 1970s and most of the 1980s. They operated independently and many people thought they would never meet or join.

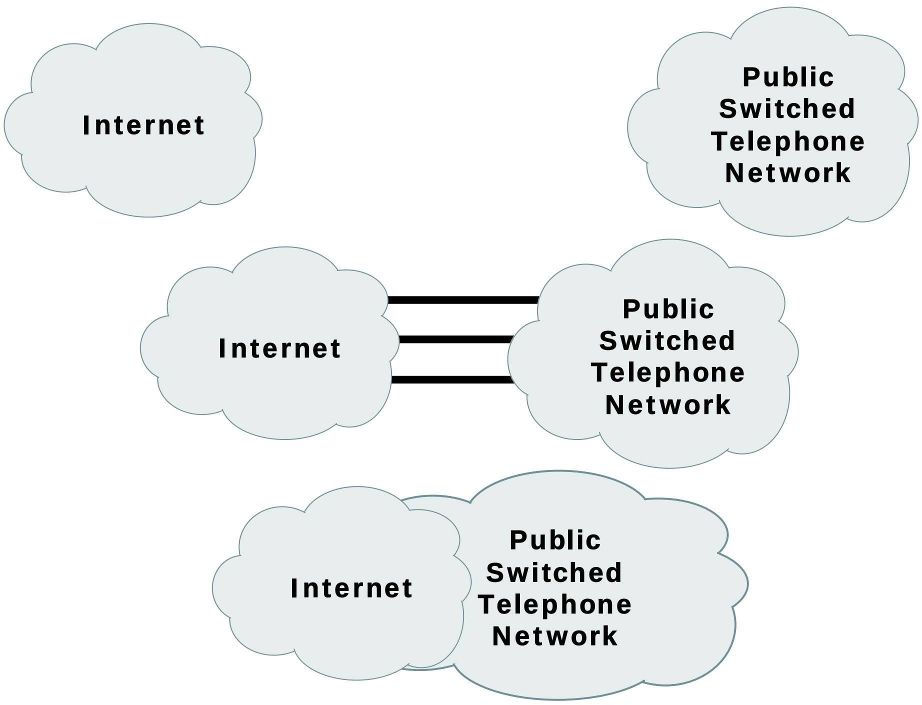


Figure 1: Network clouds converge—the Internet meets the PSTN.

In the center are connections between the two networks—separate but connected. Although there are only three lines shown, there are thousands of links. When users began dialing into the Internet using modems, the two networks began to interweave in a limited fashion. As VoIP technologies began to evolve, it became obvious that an *Internet phone call* would have to maintain some form of compatibility and interoperability with the PSTN in order to provide ubiquitous service. Connectivity alone cannot support telephony. Interoperability is a crucial factor. This conclusion presented the IETF with an often-unspoken mandate to work in cooperation with the ITU-T and assure that Internet standards supported and remained compatible with PSTN standards. The goal for many has been seamless interoperability between the two networks. The phrase *transparent to the end user* has never been used more than in discussions of interoperability between these two environments. This collaborative effort toward a ubiquitous, interoperable technology was the dominant philosophy of many, but not all, developers of the early IP telephony standards.

Today, convergence is talked of as if it's behind us, but the efforts continue. We view convergence as the migration of voice, data, and video services to a single consolidated network infrastructure. Most often, we envision this infrastructure to be a TCP/IP network, like the Internet. In the past, people said that the Internet was growing so quickly it would absorb the PSTN. Others claimed that telephony would migrate to the Internet and that one day the old legacy PSTN would simply be turned off.

After several years of evolving VoIP technologies, we see that both networks are vital, growing, and critical, although the moderate pace of improvement in VoIP technologies is creating the potential for far deeper penetration of a single infrastructure. One common view in the past was that the two networks would become linked only at high capacity access points, passing traffic through gateways to each other. Today, they are far more inextricably connected than anyone imagined, with thousands and thousands of gateways. The two networks haven't become one, but they are so tightly coupled delivering business services that they are beginning to represent a single cloud as shown in the bottom of Figure 1. This cloud is nothing more than raw service capacity, with services being defined through agreements between users (as service level agreements—SLAs) and providers (as peering agreements). The network of tomorrow is partly here today. It's a high-performance cloud of capacity that provides whatever service the end user requests of it.

> For a more detailed review of this concept and the evolution of the networked world, see Steve Shepard's book *Telecommunications Convergence* (McGraw-Hill).

Session Initiated Protocol

The original work on SIP was performed by the IETF as one of several efforts. The Multiparty Multimedia Session Control (MMUSIC) working group took much of the lead in very early efforts. Since 1999, the IETF-SIP working group has led this work. Their specific charter states that SIP is a text-based protocol, similar to HTTP and SMTP, for initiating interactive communication sessions between users. Such sessions include voice, video, chat, interactive games, and virtual reality. This group has worked to bring SIP from proposals to drafts and standards in addition to specifying and developing proposed extensions that have arisen from very aggressive protocol and feature requirements. The SIP working group will concentrate on the specification of SIP and its extensions, and will not explore the use of SIP for specific environments or applications.

Throughout its work, the group will strive to maintain the basic model and architecture defined by SIP:

- Services and features are provided end-to-end whenever possible
- Extensions and new features must be generally applicable rather than applicable to only a specific set of session types
- Simplicity is key
- Reuse of existing IP protocols and architectures, and integrating with other IP applications, is crucial

SIP provides protocols and mechanisms that allow both end systems and proxy servers to provide services including:

- Call forwarding under a variety of scenarios (no answer, busy, and so on)
- Calling party and called party number identification using any naming scheme
- Personal mobility allowing a single address that is location and terminal independent
- Capabilities negotiation between terminals
- Call transfer
- Instant messaging
- Event notification
- Control of networked devices

There are also extensions to SIP that provide for fully meshed conferences and connections to multipoint control units (MCUs).

SIP uses an addressing structure very much like email addressing. Given that users might log on from any location and receive an IP address dynamically, there must be some way to resolve some common convention to the active and current IP address. As people are familiar and comfortable with email addresses, this structure seems most appropriate and remains a popular choice.

SIP is a text-based protocol like HTTP or SMTP, so the addresses, which are SIP Uniform Resource Locaters (URLs), can be imbedded in email messages or Web pages. Also, as a text protocol, the addresses are network neutral. Thus, the URL might point to an email-like address, using SIP, an H.323 address, or it might point to a PSTN telephone number. The ITU-T E.164 standard defines the telephone numbering structure.

SIP provides a comprehensive set of building blocks that can be extended to allow for E911 or Advanced Intelligent Network Services. Because it can support forking to multiple destinations, SIP can support call forwarding, Automatic Call Distribution (ACD) techniques for call centers, and redirecting a call to multiple alternative locations.

SIP operates independently of the network layer and requires only unreliable datagram or packet delivery. It provides its own reliability mechanism. Although in the IP environment of the Internet, SIP is used over UDP or TCP, it could run over IPX, Frame Relay, ATM AAL5, or X.25 without modification. Generally, UDP is used to avoid the overhead associated with TCP.

The protocol model for SIP as originally described in RFC 2543 is shown in Figure 2. It provides four functional components:

- User Agents either initiate call requests or are the destination of those requests. A User Agent might be IP telephony software running in a computer or an IP telephone.
- The Registrar keeps track of users within the network or domain. User Agents register with the registrar as members of the network
- The Proxy Server is an application layer routing process that directs SIP requests and replies within the network.
- The Redirect Server receives requests for users (UAs) and provides the location of other SIP User Agents or servers where the called party can be reached.

Within the SIP Server, the Registrar, Proxy Server, and Redirect Server can be implemented in the same software package.

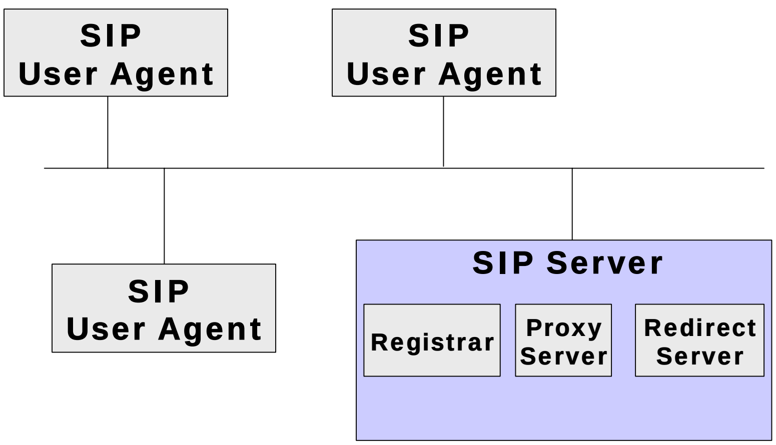


Figure 2: The SIP model.

In a SIP session, a user initiates a call, which prompts the User Agent to transmit a SIP message. These messages will then traverse one or more SIP Servers. Once the destination User Agent information is obtained, actual message transfer takes place directly between the User Agents. If one end of the call is located in the PSTN, some gateway between the IP-based SIP network and PSTN is required to provide all the necessary protocol conversions between networks.

Session Description Protocol

The MMUSIC working group of the IETF also provided RFC 2327—the Session Description Protocol. SDP is intended for describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation. SDP is used within both SIP and Megaco implementations. SDP is not intended to support negotiation of session content or media encoding but to act as a general-purpose tool. It also supports multicast media and can be used for broadcast environments such as Internet radio or television.

> ✎ This article doesn't explore SDP in depth, but it's beneficial to have a high-level grasp of how the information describing multimedia sessions is transmitted between user systems.

SDP provides Session Announcements as the mechanism used to convey session description information between devices or nodes and proactively delivered to users. These announcements might also be delivered via email or the Web, allowing for automatic launching of the appropriate application on the called parties workstation. SDP includes:

- The session's name and purpose
- Time the session is active
- The type of media used in the session; this might be voice, video, data, and so on
- The format of the media (MPEG video, H,261 video, and so on)
- The transport protocol used (UDP, TCP, IP, and so on)
- Information necessary to receive the media (TCP/IP ports, addresses, and formats)

The actual syntax for the port and addressing information vary depending on the transport protocol in use. Following is an example of an SDP description:

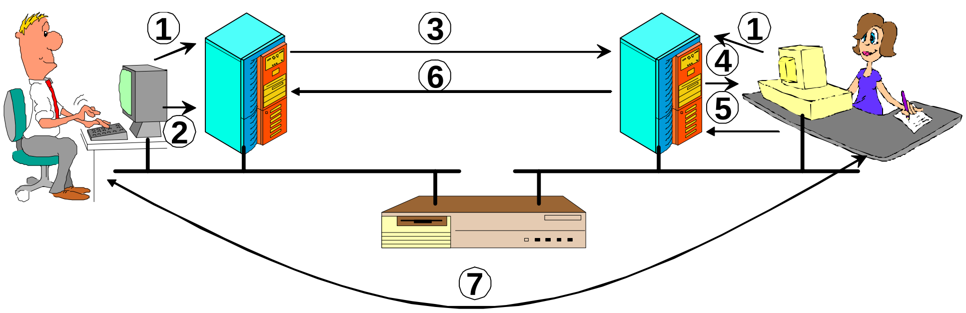
```
v=0
o=kcamp 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP Seminar
i=A Seminar on VoIP
u=http://www.realtime-voip.com/seminar/voip.52.ps
e=ken@ipadventures.com (Ken Camp
c=IN IP4 63.215.128.129/127
t=7944393265 8746931596
a=recvonly
m=audio 49360 RTP/AVP 0
m=video 51782 RTP/AVP 31
m=application 32416 udp wb
a=orient:portrait
```

In general terms, SDP is used to convey enough information for a user to join the session or call. This information might not include encryption keys in the VPN environment. Those might be handled by another set of protocols such as IPSec.

Call Setup Using SIP

Whether it involves voice or data communications, we often use the same couple, Bob and Alice, to explain how a communication occurs, and we'll use them in Figure 3. Let's step through a call using SIP. We'll assume Bob and Alice are on different LANs and that the two networks are connected by a router. Each network has a SIP Server on the local LAN in this simulation.

1. When Bob and Alice turn on their computers, the User Agent software (part of the VoIP client), automatically registers them each with their local SIP Server.
2. Bob initiates the telephone call, and the User Agent on his computer transmits an *invitation* to the SIP Server on his local network. This *invitation* contains the session description information.
3. Because Alice registered with the SIP Server on her own local network, the SIP Server on Bob's network doesn't know how to reach her. Bob's SIP Server has to forward the *invitation* to every SIP Server it knows how to reach—Alice's SIP Server, in this case.
4. As Alice is on the same LAN and already registered with her SIP Server at startup, it knows how to reach her and forwards the *invitation* to her.
5. Alice also wants to talk to Bob, so she answers that call, which returns and acknowledgement (ACK) over the same path the *invitation* followed. Alice's session description information is included in the acknowledgement.
6. Now that both ends have exchanged session description information, they have the IP address and port information to directly contact the other party on the call and can now transmit RTP encapsulated media directly. The SIP Servers are no longer needed on the session.
7. The conversation takes place.



1 Registration – performed by each station

2 Bob initiate call, sending invitation and SDP

3 SIP Server forwards invitation to all known SIP Servers

4 Alice's SIP Server delivers invitation to called party

5 Alice accepts invitation and returns SDP

6 SIP Server returns ACK to calling party

7 End-to-end telephone conversation

Figure 3: Bob and Alice make a SIP phone call.

One advantage to this process is that it's much simpler than H.323, using fewer messages for call setup. H.323 is a protocol we'll explore in another article. SIP doesn't require the overhead of TCP's three-way handshake and guaranteed delivery. The registration process with SIP Servers can provide better extensive support for mobile users. And, as SIP is a text-oriented protocol, a simple BYE command is used to terminate the session.

> This article serves as a very brief technical overview of SIP. It presents one view of how SIP works. There are numerous references on the Internet for more information about SIP. For readers wishing to explore further, Wikipedia has an excellent starting point article at http://en.wikipedia.org/wiki/Session_Initiation_Protocol. There is also an excellent set of resources maintained by Columbia University at http://www.cs.columbia.edu/sip/.

Summary

SIP is one of several protocols being used to implement VoIP solutions today. It offers several competitive advantages as an efficient protocol and is very popular in some circles. In articles to follow, we'll explore other VoIP protocols.

Appendix

The following list highlights SIP RFC references:

RFC 3261 SIP: Session Initiation Protocol - The core protocol specification; obsoletes RFC 2543.

RFC 3524 Mapping of Media Streams to Resource Reservation Flows

RFC 3515 The Session Initiation Protocol (SIP) Refer Method

RFC 3487 Requirements for Resource Priority Mechanisms for the Session Initiation Protocol (SIP)

RFC 3486 Compressing the Session Initiation Protocol (SIP)

RFC 3485 The Session Initiation Protocol (SIP) and Session Description Protocol (SDP) Static Dictionary for Signaling Compression (SigComp)

RFC 3428 Session Initiation Protocol (SIP) Extension for Instant Messaging

RFC 3420 Internet Media Type message/sipfrag

RFC 3388 Grouping of Media Lines in Session Description Protocol (SDP)

RFC 3361 Dynamic Host Configuration Protocol (DHCP-for-IPv4) Option for Session Initiation Protocol (SIP) Servers

RFC 3319 Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers

RFC 3327 Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts

RFC 3326 The Reason Header Field for the Session Initiation Protocol (SIP)

RFC 3325 Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks Defines

RFC 3324 Short Term Requirements for Network Asserted Identity

RFC 3323 A Privacy Mechanism for the Session Initiation Protocol (SIP)

RFC 3329 Security Mechanism Agreement for the Session Initiation Protocol (SIP)

RFC 3313 Private Session Initiation Protocol (SIP) Extensions for Media Authorization

RFC 3312 Integration of Resource Management and SIP Framework for preconditions

RFC 3311 The Session Initiation Protocol (SIP) UPDATE Method

RFC 3262 Reliability of Provisional Responses in the Session Initiation Protocol (SIP)

RFC 3263 Session Initiation Protocol (SIP): Locating SIP Servers

RFC 3264 An Offer/Answer Model with the Session Description Protocol (SDP)

RFC 3265 Session Initiation Protocol (SIP)-Specific Event Notification

SIP event model; defines SUBSCRIBE and NOTIFY

RFC 3087 Control of Service Context using SIP Request-URI

RFC 3050 Common Gateway Interface for SIP

RFC 2976 The SIP INFO Method

RFC 2848 The PINT Service Protocol: Extensions to SIP and SDP for IP Access to Telephone Call Service