SIP can provide
realtime communications
as a network service.

MARTIN J. STEINMANN, PINGTEL

Unified Communications with SIP

Communications systems based on the SIP (Session Initiation Protocol) standard have come a long way over the past several years. SIP is now largely complete and covers even advanced telephony and multimedia features and feature interactions. Interoperability between solutions from different vendors is repeatedly demonstrated at events such as the SIPit (interoperability test) meetings organized by the SIP Forum, and several manufacturers have proven that proprietary extensions to the standard are no longer driven by technical needs but rather by commercial considerations.

Even in light of all this excellent news, most implementations still fall short in one key area: native SIP call control and SIP-based feature interaction required for multivendor interoperability. SIP first unfolds its full potential if it is used for more than just transport “channels” that interconnect otherwise proprietary IP PBX implementations. The simple fact that SIP “goes in” and “comes out” of a PBX system does not mean that this system has much to do with SIP at all. Native SIP call control and SIP transport “channels” are two very different

and oft-confused architectural approaches to building SIP communications systems.

Thanks to many enterprise users who increasingly insist on standards-based, open, and therefore interoperable systems, the industry is embracing a new model. Realtime communications, including telephony, is starting to look like yet another IT application—an application that runs on standard hardware, uses standard operating systems and middleware, and follows PC-like economics. It is an application designed as an open system that accommodates a wide variety of endpoints from many different vendors and integrates into an existing IT infrastructure with Web services, corporate directories, and IT best practices.

THE UNIFIED COMMUNICATIONS CORE

Ease of use, manageability, resiliency, and scale start with the right architecture. Communications systems implementing native SIP call control are distributed systems that offer a unified communications core with end-to-end SIP message routing. Features are implemented as autono-

Unified Communications with SIP

mous services that interact with phones, gateways, and other features using SIP.

Let's look at a specific example: the MoH (music on hold) feature is defined in the SIP standard as a music server that sits on the network; it is effectively a SIP user agent that plays music. Putting a call on hold transfers that call from the user's phone to the music server. Retrieving the call signals the music server to transfer the call back. Several such music servers can be part of a system, providing redundancy, accommodating different user groups or locations, supplying scalability, or simply offering a different choice of music.

This unified communications core is not confined to a single IP PBX box implementing what is commonly referred to as a B2BUA (back-to-back user agent), but instead offers realtime communications capabilities as a network service. After all, a SIP PBX is not meant to be a collection of UAs terminating point-to-point SIP "channels," but rather a SIP proxy able to route SIP signaling flows to their proper destinations.

Feature servers connect to this unified communications core and provide services, such as call park, conference bridge, call center, IM, MoH, and many others. Each service is assigned the necessary resources in the form of server hardware and network connectivity, and resiliency is provided on a per-service basis with the core unified communications infrastructure being fully redundant and fault tolerant.

This architecture is not just a technical issue that matters only to engineers who build SIP systems; it also matters to the end user. The rest of this article outlines the connection between the technical architecture of a SIP system and the effect this can have on voice quality, flexibility, scalability, reliability, and the direct impact on both capital expenditure and operating cost. Where possible, the open source SIPxchange project on <http://www.sipfoundry.org> is used as an example of how such a unified communications core can be implemented.

HOW VOICE QUALITY IS INFLUENCED IN YOUR SYSTEM

Two primary factors affect the quality of a conversation: the voice codec used; and the quality of the network con-

nection between the two phones measured in terms of overall delay, jitter, and available bandwidth.

As SIP-based communications systems are designed to provide better voice quality than comparable TDM (time-division multiplexing) systems, the SIP standard implies a strict separation of media and signaling traversing the unified communications core. Signaling is handled by SIP proxies, where media flows peer-to-peer along the most direct route between endpoints. This is very critical because voice quality is directly affected by overall delay, jitter, and available bandwidth.

In a proxy-based SIP architecture, endpoints negotiate the best available codec among themselves as the call is set up. No transcoding is normally required, unnecessary systems are not traversed, and voice travels on the most direct route possible. This is important for communications on a LAN and is absolutely critical in a multisite deployment where one SIP server allows users from different locations to obtain service.

Consider the following: you travel with your colleague to a different city and, using your softphone, call her in her hotel room. Do you want the media for this call to go back to your company's office, through the PBX, and back out to your colleague, or would you prefer that audio to flow directly between the two hotel rooms?

In addition to P2P (peer-to-peer) routing of media, a system should easily accommodate and support a variety of codecs. There are many different voice codecs, from wide-band audio (HD voice), to the normal PSTN (public switched telephone network) quality codec, to highly compressed voice that saves bandwidth. As devices communicate peer-to-peer and negotiate codecs between them, new codecs become available to the system as they are introduced by the end devices. No change or upgrade is required to the IP PBX system, and no limitations are imposed on the number of calls the system can accommodate simultaneously for any given codec since media streams do not have to go in and out of the IP PBX server (this would overload the Ethernet interface of the server). SIP also ensures that no single point of failure is introduced, as even if the IP PBX controller goes down, calls that are already set up will continue uninterrupted.

RESILIENCE IS A MAJOR CONCERN

Because you rely on your communications system for many needs, you should never have to worry about the system failing. Back in the old days when PBX systems were based on proprietary hardware, they were engineered to be resilient. Redundancy was built into the hardware. Whereas these one-box solutions were easy to

buy and deploy, they also tended to be costly.

Knowing servers can go down, IT now has a philosophy that ensures open systems are built to be tolerant of such failures. If a router in the network goes down, the network routes around it; if an e-mail server goes down, messages are queued for later delivery; if a Web server goes down, another one takes its place. SIP-based communications systems work in exactly the same way. Redundancy in SIP is provided by allowing more than one SIP proxy server to be on the network routing calls. If one goes down, the DNS will route requests to another server automatically and seamlessly to the user as it would with requests destined for a Web server.

No expensive and hard-to-manage cluster technology is required, nor do you need high-availability hardware. In fact, resilient SIP systems can be built with the cheapest of hardware and even using geographically distributed set-ups. The open source SIPxchange system illustrates this. A master-slave configuration between two servers can be set up easily using the set-up wizard. Phones will register with either server based on a load-sharing policy defined in the DNS server. Call processing is shared between the two servers using the same load-sharing policy. State in the registrar server is shared between the two machines in realtime, so that if one machine fails, the other takes over without losing any calls. Both servers are centrally managed and appear as one large system with a common dialplan to the user and administrator.

REALIZING A UNIFIED DIALPLAN

For many companies the ultimate goal is to have a unified infrastructure that includes a unified dialplan between headquarters and possibly many branch offices. Local emergency calls still have to be possible even if the IP connection to headquarters is unavailable. Limiting the number of PSTN trunk lines at branch-office locations to the bare minimum required for emergency calling saves additional cost through pooling of resources and even allows the enterprise to use a SIP trunking provider (an Internet telephony service provider) instead of the traditional T1 PSTN connectivity.

Although the concept of a fully redundant system can be extended from two redundant servers to a mesh of interconnected SIP proxy servers providing call control as a true network service, this is most often not a cost-effective solution when it comes to connecting many branch offices. The reason lies in the need to synchronize registration state information in realtime between the systems. A centralized and fully redundant SIP server is more cost effective. It provides call routing for all the offices, similar

Peer-to-peer Routing of Media and NAT Traversal

Routing of SIP messages and media across network boundaries has long been a major challenge for the adoption of SIP (Session Initiation Protocol). Enterprise deployment as discussed here differs from the most generic case of allowing communications to take place between random endpoints on the Internet in that the enterprise environment has to be secure at all times.

Protocols standardized by the IETF (Internet Engineering Task Force) for the purpose of NAT (network address translators) traversal, such as STUN (simple traversal of UDP through NATs) and ICE (interactive connectivity establishment), do not encompass specific requirements for the secure traversal of enterprise boundaries (for more information, see Sparks on page 22). Traversing enterprise NATs and firewalls is still a difficult problem to solve given the requirements to provide both security and full-feature transparency.

Both STUN and ICE require that firewall ports are opened for SIP traffic, and the protocol then employs algorithms to map specific fields of the SIP messages between the inside and the outside networks. To assist in the process they use services run on the outside network such as a STUN server. Exploits and other malware can penetrate these holes in the same way regular and compliant SIP packages can, and all package inspection is in addition to what STUN and ICE provide. A denial-of-service attack targeting the internal SIP communications server is even possible from the outside, exploiting holes opened in an enterprise's perimeter security for the purpose of NAT traversal.

In an enterprise context, therefore, SIP traffic sent over insecure networks is often encapsulated into enterprise VPN tunnels, which if configured correctly, make NAT traversal unnecessary and provide strong security. Using SBCs (session border controllers) is another alternative. Such appliances provide both NAT and firewall traversal combined with the necessary security mechanisms such as deep-packet inspection, rate limiting to prevent denial-of-service attacks, and encryption of both SIP signaling and media. Far-end NAT traversal is then addressed using STUN and ICE protocols with additional elements such as STUN servers provided by the SBC.

Unified Communications with SIP

to the old concept of using a Class 4 switch. Dialplan flexibility in terms of number portability between offices can be accomplished through a central directory that provides number mapping to SIP addresses. Such a directory is nothing but the already-in-place DNS server with the added information of how to map internal telephone numbers to SIP addresses. This telephone number-mapping capability is called private ENUM and is part of the SIP standard.

VENDOR LOCK-IN—A REALITY OF THE LAST MILLENNIUM

Shortly after the introduction of the first IBM PC, the IT industry transitioned away from monolithic and vertically integrated solutions in favor of a model that brought tremendously faster innovation and significantly lower prices. Much later, this was called “industry disintermediation,” which led to specialization and the emergence of many innovative and interoperable products.

The PBX industry operated within the old model until very recently. Tied together by proprietary call control, PBX vendors were able to control both their customers and channels, forcing them to buy the entire solution from one vendor. That’s like buying all your servers, laptops, PCs, applications, middleware, and operating systems from one vendor. What kind of networks, computers, and applications would we be using today if innovation in the IT industry over the past 20 years had been confined to R&D labs of a few large companies working on closed and therefore incompatible solutions?

With the introduction of SIP, it is now possible to choose the components of voice communications systems one at a time. More and more phone manufacturers such as Polycom, Snom, Grandstream, Aastra, Hitachi, and ClearOne have brought to market SIP-compliant phones. (As part of the SIP Forum SIPit interoperability activities, Pingtel offers a free phone interoperability test portal at <http://interop.pingtel.com>, where phones can be tested against the SIP standard for many of the key features.)

Manageability of a total solution, often a major concern with solutions that consist of a collection of best-of-breed components, still represents a challenge. The open

source SIPxchange project has made a pioneering effort to offer plug-and-play management for an increasing list of phones. Plug-and-play management means that the user gets the same ease of use as is expected from a single-vendor solution in a system that accommodates many different phones from different manufacturers. Phones are provisioned automatically where all configuration data is generated, distributed, and backed up by SIPxchange.

Choosing a new phone system for your company should not be dominated by the question of whether the CEO and other key employees prefer one brand over another. Selecting a phone should be a different decision from selecting an IP PBX, as is the case with selecting laptops and PCs to work with your servers and applications.

FEATURES ADDED AS NETWORK SERVICES

The battle of features continues to rage unabated: “We support 500 telephony features. How many do you have?” Or, “We give you more features thanks to our *enhancements* to the SIP standard.”

This debate is misleading. There simply is no such thing as a list of 500 basic telephony features. In the absence of a generally accepted industry definition of a “basic telephony feature,” the counting can get quite tricky. With the introduction of multimedia communications augmented by federated presence and computer-telephony integration, the set of basic system features has become a moving target that is constantly evolving. Hundreds of legacy features are no longer relevant, but many new features are added in support of a presence-based multimedia experience.

Back when a PBX system came as a single-box solution, all of its features ended up on a long list and counted toward the system’s “basic features.” With SIP, however, an IP PBX became a collection of distributed and interoperable components, with each server providing a distinct function or feature. These feature servers, such as the call-center solution or the conferencing solution, should be separate systems for the reasons outlined earlier. The counting has to take this into account.

With the introduction of SIP, a lot of intelligence moved out to the phones; therefore, the counting of features depends on the brand and type of phone. The real challenge is not how to implement the most features under the assumption that the same vendor sells both the IP PBX and the phones, but rather the challenge is how many different phones are interoperable with a given IP PBX while still offering the features you are looking for in a consistent manner. Proprietary features are worth much less when compared with standards-based features,

as the total cost of ownership for a proprietary system is significantly higher.

Imagine your e-mail system relying on a set of proprietary features that work only on a system procured from a particular vendor. Every time you wanted to send an e-mail, you would first have to consider whether the recipient's system is capable of receiving what you are about to send. As ridiculous as this might sound, this is the situation we are in with our phone system. Try to set up a shared line with someone connected to a different system, or try to subscribe to the presence state of your trading partner's line so that you can see whether he or she is available to take your call. Assuming you have the required authorization and credentials, this should easily be possible—but it is not.

Consider the following example of a network-based service: call park and retrieve is one of the key basic telephony features. Its implementation, according to the relevant IETF standard, requires the phone to transfer the call to a park server when the park button is pressed. That park server is not necessarily the IP PBX server, but could be anywhere in the network. Pressing the park button again transfers the call back. This provides great flexibility and scalability, while not putting the burden on the phone to provide music on park to potentially many active lines. Since several park servers can be used on the same network, it is even possible to provide resiliency for the call-park and retrieve feature. If you are used to B2BUA-based IP PBXs, this looks quite odd, since in this case putting a call on park means that the IP PBX plays music without requiring the call to be transferred or any other SIP signaling to occur. Proprietary logic and a bit of "pixie dust" instead of standards-based signaling quickly get in the way of interoperability.

CHALLENGES IN SYSTEMS MANAGEMENT AND ADMINISTRATION

Telephone systems used to be separate autonomous systems. Making a change, moving an employee to a new office, or adding a new employee required a request to a special group responsible for administering the PBX. A smaller company would often have to call a technician from its local reseller to help with such additions, moves, and changes.

SIP-based communications systems are IT systems that are managed as part of a company's IT infrastructure. Users, along with their credentials and privileges, are created in Microsoft Active Directory or LDAP (Lightweight Directory Access Protocol), and applications such as the e-mail system and the communications system are

expected to synchronize with this directory and make changes accordingly.

The human resources manager would like to have a Web application as part of the company's intranet that allows the easy creation of all the necessary accounts for new employees or deletion of accounts for departing employees. Such applications are typically built as Web services using protocols such as SOAP to interact with the communications system. Reports such as CDRs (call detail records) or call statistics from the call-center application should be accessible as a Web service, too. That way they can easily be integrated into existing reporting applications or made available through Web portals.

Another big challenge in systems management is administering phones and gateways. Every phone and gateway has a unique configuration that needs to be created, updated, and backed up. Firmware updates have to be administered. IP phones have hundreds of parameters, many of which are critical for all features of a PBX system to work properly with a particular phone. Therefore, the IP PBX system needs to be able to automatically generate phone configuration profiles based on default parameters, once the system discovers a phone or gateway. An administrator learning how to configure phones manually would require extensive training and is therefore not an affordable option.

OPEN SOURCE IS A CREDIBLE ALTERNATIVE

This article used the open source SIPxchange solution to illustrate some of the key features and challenges of a SIP-based communications system. As a second-generation open source IP PBX system, the SIPxchange project strictly adheres to the SIP standard in an attempt to build a fully distributed, open, and interoperable SIP infrastructure for the enterprise. Open source is here to stay and has grown into a credible alternative to proprietary and often expensive systems. Building a unified communications core with SIP is one area where you can begin to use it to your advantage. Q

LOVE IT, HATE IT? LET US KNOW

feedback@acmqueue.com or www.acmqueue.com/forums

MARTIN STEINMANN is senior vice president of marketing at Pingtel. He is also a founder and member of the board at SIPfoundry, an open source community dedicated to VoIP technology. He earned B.S. and M.S. degrees in electrical engineering from the Swiss Federal Institute of Technology and a Ph.D. in physics from the Danish Technical University.
© 2007 ACM 1542-7730/07/0300 \$5.00