Service Creation and Service Delivery All in One

Programmable SIP Application Servers for Next Generation Networks

May 2004
Abstract: The adoption of open, flexible, standards-based service creation environments (SCEs) is critical for the rapid introduction of new services and business models into emerging Next Generation Networks. These SCEs in turn will need to interact with a variety of the underlying building blocks of these new services. Out of all of these, the most important is Session Initiation Protocol (SIP), already in wide scale deployment around the world supporting Voice over IP hosted services, enterprise IP PBX systems and client/server desktop applications, as well as playing a vital part in the Internet Multimedia Subsystem (IMS) architectures of both fixed and mobile service providers.

This document explains the central role that SIP Application Servers and SIP Servlet-based Service Creation Environments have in supporting a wide range of new services and new communities of interaction.

1. Introduction

Slowly but surely, agreement is starting to form about the future shape of the world's telecommunications networks and services. No longer separated into the traditional domains of wired versus wireless and circuit-switched versus packet-switched, many networks already under construction are being characterised by openness, flexibility and the potential for interaction across a wide range of different devices and access technologies. Ideally, users will be able to move seamlessly across different technology and network domains, freely accessing mixed sets of voice, data and multimedia services and utilizing applications that add value to the basic connectivity service, such as presence and location information.

While many of the components needed to create these new services are already well defined, a number of issues need to be fully resolved.

First, because of the complexities of the underlying bearer and signalling technology that need to support services across the emerging NGN, it's essential that some form of abstraction layer is created to make this complexity manageable on a day to day basis.

Second, the cost and time overheads involved in traditional IN-based models of service creation are no longer supportable in today's commercial environment. By using the SIP Servlet API defined by the Java Community Process, it instead becomes possible for communications applications to be developed using programming skills using a fraction of the resources previously required.

Additionally, the Internet-oriented development of SIP Servlets means that third party developer communities such as J2EE and .NET will be able to apply the velocity of innovation seen on the Internet over the past decade to the NGN, where all future service provider capex investments are being made. This serves to create entirely new classes of telecommunications applications and new revenue streams for the whole industry.

Finally, because of SIP's inherent flexibility and power, specific interfaces can be built between SIP signalling-based networks and various other Service Creation Environments such as Parlay/OSA and JAIN (Java Applications for Intelligent Networks), bringing the benefits of other published APIs directly into the telecommunications network and allowing SIP applications to control call set ups and messaging.

2. Why SIP – and why now...?

At its simplest conceptual level, SIP actually does what its name suggests – that is, it sets up sessions between different applications and devices in very much the same way that HTTP works in the Internet already. These sessions may be voice calls, video streaming sessions or application requests – the actual technology is agnostic, making it a robust and highly flexible protocol with a low data overhead.

In contrast to the ITU's SS7 standard used for call set up and control and its H.323 video protocol suite, SIP operates independently of the underlying network transport protocol and is indifferent to the media transactions that it invokes. Instead, it defines how one or more participant's end devices can create, modify and terminate a connection irrespective of whether the actual content is voice, video, data or web-based.

SIP is also a major upgrade to protocols such as the Media Gateway Control Protocol (MGCP), which converts
PSTN audio signals to IP data packets. Because MGCP is a closed, voice only standard, enhancing it with signalling capabilities is complex and at time this has resulted in corrupted or discarded messages that handicap service providers’ creativity. With SIP, however, service developers can add new data – and hence new functionalities - to messages without compromising the integrity of the network and its associated services.

For example, service providers using SIP are already establishing entirely new services based on the previously separate media of voice, video and chat or instant messaging. With the previous technologies of MGCP, H.323 or SS7, the provider would have to wait for new protocol iterations to support these services.

Moreover, because SIP is analogous to HTTP in the way that messages are constructed, developers can more easily and quickly create applications using popular programming languages such as Java. Without the need for specialist and expensive telecommunications software skills, service providers are now able to create and deploy applications in just a few weeks or months – as opposed to the much longer development times when SS7 or IN-based technologies are used.

3. How does it work?

Like the Internet, SIP is easy to understand, extend and implement. As an IETF specification, SIP extends the open-standards spirit of the Internet to messaging, enabling disparate types of computers, phones, software and even televisions to communicate with one another.

As previously noted, a SIP message is very similar to HTTP (RFC2068) and much of the syntax in message headers and many HTTP codes is reused. Using SIP, for example, the error code for an address not found – “404” – is identical to the Web’s. SIP also reuses the SMTP for address schemes, with a SIP address – such as sip:guest@sipcenter.com has the exact structure of an email address. SIP even leverages Web architectures, such as Domain Name System or Service (DNS), making messaging amongst SIP users even more extensible.

Using SIP, service providers can freely choose amongst standards-based components and quickly harness new technologies. Users can locate and contact one another regardless of media content and the numbers of participants, adding, dropping and transferring users as appropriate during a session as well as changing session features on the fly.

SIP however is not a complete panacea. It is neither a session description protocol, nor does it provide conference control. To describe the payload of message content and characteristics, SIP uses the Internet’s Session Description Protocol (SDP) to describe the characteristics of the end devices. SIP also does not itself provide Quality of Service (QoS) and interoperates with the Resource Reservation Setup Protocol (RSVP) for voice quality. It also works with a number of other protocols, including the Lightweight Directory Access Protocol (LDAP) for location, the Remote Authentication Dial-In User Service (RADIUS) for authentication, plus RTP for real-time transmissions, among many others.

Essentially, SIP provides for the following basic requirements in communications:

1. User location services
2. Session establishment
3. Session participant management
4. Limited feature establishment

An important feature of SIP is that it does not define the type of session that is being established - only how it should be managed. This flexibility means that SIP can be used for an enormous number of applications and services, including interactive gaming, music and video-on-demand, as well as more standard voice, video and Web conferencing.

4. What you can do - typical SIP-based services

<table>
<thead>
<tr>
<th>Service Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ad-hoc conferencing</td>
<td>Leverages buddy lists to allow users to easily set up group calls</td>
</tr>
<tr>
<td>Chat</td>
<td>Allows users to talk at the same time that they are sharing information and applications</td>
</tr>
<tr>
<td>Enhanced messaging</td>
<td>Allows users to escalate from Instant Messaging to a voice call</td>
</tr>
<tr>
<td>Enhanced voice mail</td>
<td>Allows users to leave messages in either voice, text or email formats</td>
</tr>
<tr>
<td>Find-me</td>
<td>Allows users to choose to ring an alternative number</td>
</tr>
<tr>
<td>Location-based services</td>
<td>Allows users to specify areas of interest and have relevant information pushed to them based on their geographic position</td>
</tr>
<tr>
<td>Picture messaging</td>
<td>Leverages buddy lists and IM, allowing users to instantly share photos</td>
</tr>
<tr>
<td>Push-to-Talk</td>
<td>Near-instant one-to-one or one-to-many walkie-talkie style calls</td>
</tr>
<tr>
<td>Session-based transfers</td>
<td>Allows users to switch from voice to video to application sharing, as they wish in one communications session</td>
</tr>
<tr>
<td>Infotainment</td>
<td>Media push, allowing users to designate information or entertainment content they want and have it pushed to their phone or device as it becomes available</td>
</tr>
</tbody>
</table>
5. Key Characteristics of SIP

- SIP messages are text-based and hence are easy to read and debug, making the creation of new services far simpler and more intuitive for designers.
- SIP reuses MIME type descriptions in the same way that email clients do, allowing applications associated with sessions to be launched automatically.
- SIP reuses several existing and mature Internet services and protocols such as DNS, RTP and RSVP, so no new technologies need to be introduced to support the SIP infrastructure.
- SIP extensions are easily defined, enabling service providers to add them for new applications without running the risk of impacting the network or existing services. Older SIP-based equipment in the network will not impede newer SIP-based services, and instead an old SIP implementation will simply ignore a newer application’s method or header.
- SIP is transport layer independent and could even use IP over ATM. SIP employs the User Datagram Protocol (UDP), as well as the Transmission Control Protocol (TCP), flexibly connecting users independently of the underlying infrastructure.
- SIP supports multi-device feature levelling and negotiation. If a service or session initiates video and voice, voice can still be transmitted to non-video enabled devices, or other device features can be used, such as one-way video streaming.

6. The Anatomy of a SIP session

SIP sessions utilize up to four major components: SIP User Agents, SIP Registrar Servers, SIP Proxy Servers and SIP Redirect Servers. Together, these systems deliver messages embedded with the SDP protocol defining their content and characteristics to complete a SIP session. Below is a high-level description of each SIP component and the role it plays in this process.

SIP User Agents (UAs) are the end-user devices, such as cell phones, multimedia handsets, PCs, PDAs, etc. used to create and manage a SIP session. The User Agent Client initiates the message. The User Agent Server responds to it.

SIP Registrar Servers are databases that contain the location of all User Agents within a domain. In SIP messaging, these servers retrieve and send participants’ IP addresses and other pertinent information to the SIP Proxy Server.

SIP Proxy Servers accept session requests made by a SIP UA and query the SIP Registrar Server to obtain the recipient UA’s addressing information. It then forwards the session invitation directly to the recipient UA if it is located in the same domain or to a Proxy Server if the UA resides in another domain.

SIP Redirect Servers allow SIP Proxy Servers to direct SIP session invitations to external domains. SIP Redirect Servers may reside in the same hardware as SIP Registrar Servers and SIP Proxy Servers.

The following scenarios demonstrate how SIP components work in harmony to establish SIP sessions between UAs in the same and different domains:

Establishing A SIP Session Within the Same Domain

The diagram below illustrates the establishment of a SIP session between two users who subscribe to the same ISP and, hence, use the same domain. User A relies on a SIP phone. User B has a PC running a soft client that can support voice and video. Upon powering up, both users register their availability and their IP addresses with the SIP Proxy Server in the ISP’s network. User A, who is initiating this call, tells the SIP Proxy Server he/she wants to contact User B. The SIP Proxy Server then asks for and receives User B’s IP address from the SIP Registrar Server. The SIP Proxy Server relays User A’s invitation to communicate with User B, including – using SDP – the medium or media User A wants to use. User B informs the SIP Proxy Server that User A’s invitation is acceptable and that he/she is ready to receive the message. The SIP Proxy Server communicates this to User A, establishing the SIP session. The users then create a point-to-point RTP connection enabling them to interact.
1. Call User B
2. Query “Where is User B?”
3. Response “User B SIP Address”
4. ‘Proxied’ Call
5. Response
6. Response
7. Multimedia Channel Established

Establishing A SIP Session In Dissimilar Domains

The difference between this scenario and the first is that when User A invites User B – who is now using a multimedia handset – for a SIP session the SIP Proxy Server in Domain A recognizes that User B is outside its domain. The SIP Proxy Server then queries the SIP Redirect Server – which can reside in either or both Domain A or B – for User B’s IP address. The SIP Redirect Server feeds User B’s contact information back to the SIP Proxy Server, which forwards the SIP session invitation to the SIP Proxy Server in Domain B. The Domain B SIP Proxy Server delivers User A’s invitation to User B, who forwards his/her acceptance along the same path that the invitation traveled.
The diagram above shows how a SIP Application Server – such as Ubiquity’s – fits into the application environment envisaged for an IP Multimedia Subsystem (IMS) in both the fixed and mobile domains. At the heart of the application creation environment are a powerful set of application building blocks (ABBs) such as presence, session control, conference and IVR controls and instant messaging that can be mixed and matched to create new services and reused across multiple applications. By providing a ‘top 10’ list of pre-built Servlets, this dramatically reduces the time and effort needed to build many SIP applications.

This capability comes through the use of the standard Java-based SIP Servlet ‘container’ architecture – analogous to the HTTP Servlet architecture prevalent in the web services world – as opposed to the legacy ‘silo’ architectures found in today’s stove-piped telecommunications applications. Thanks to the emergence of SIP Application Servers, both fixed and mobile service providers are now able to move towards a ‘drag and drop’ application development environment similar to that already used in e-commerce and the Internet. This allows service providers to easily and quickly offer the latest hot application – even if doing this required adding new functionality to the network.

At nine o’clock on the diagram, the SIP AS must provide open programmable connectors and interfaces into the J2EE and .NET developer communities. This means that any one of the millions of developers already familiar with Java or .Net will find it easy to create applications using a SIP AS development kit. A key benefit of this standards-based approach to telecommunications application development is that new services can be developed and launched quickly.

Returning to the diagram, at twelve o’clock, we can see how the SIP AS also provides standard interfaces to functions typically required in an IP-based wireline and mobile networks. Services created and deployed within...
today’s wireline or 2.5G or 3G environments should to able to successfully migrate through to a complete IMS infrastructure in the future.

At three o’clock, we also see how the SIP AS is designed to enable multiple third party applications such as Push-to-Talk over Cellular (PoC), infotainment and conferencing from a single horizontal application layer platform. Additionally, the SIP AS allows service providers to share common resources – such as presence, IM and session control – across multiple services. Reusing resources leads to faster development times and lower costs – all adding up to a faster ROI for service providers.

On the development side, the ability to reuse resources means that developers can reuse code and features across many applications, thus increasing efficiencies and reducing costs. For example, the same user registration database and click-to-dial capabilities can be used for both a conference service as well as for a gaming application. Alternatively, the same presence management server could be used to support both simple instant messaging applications as well as more complex, premium-priced multimedia messaging services.

No matter how powerful the SCE, the SIP Application Server must be ‘carrier-grade’ to meet the requirements of service providers as a service logic execution environment. This means scaling linearly across clusters of CPUs to meet busy hour call lodes of millions of consumers and thousands of enterprises. It also means a ‘High Availability’ software architecture that is redundant and resilient in order to achieve five 9’s of uptime.

8. Summary – The outlook for SIP
The challenges for service providers of all sizes are similar:

- How to focus on becoming value-added service providers rather than just commodity bit pipe providers and avoid becoming caught in self-defeating price wars or losing market share.
- How to make up for declining voice revenues, both by expanding data services as well as encouraging additional voice traffic.
- How to create and market distinctive and appealing new services to increasingly fragmented market niches.

Service providers can move the odds of success in their favour by focusing on key aspects of the customer experience such as ease of use, customisation and interoperability. Successful services such as SMS, PoC and mobile web portals enable a high level of user interaction and provide information rich or entertainment related content.

About Ubiquity
Ubiquity Software Corporation develops and markets SIP-based communications software, including its award-winning SIP Application Server (SIP A/S) and Speak Conference Director. The SIP A/S is both a carrier-class deployment platform and a programmable, standards-based application-creation environment (ACE) that allows providers to develop and deploy next-generation converged communications services. Speak Conference Director is a rich media conferencing application built using the SIP A/S and sold to service providers worldwide. Use of the SIP A/S is extended to service providers and Independent Software Vendors (ISVs) through open, standards-based application programming interfaces (APIs). Ubiquity has corporate offices in the US, UK and Canada.

For more information, please visit www.ubiquitysoftware.com or email info@ubiquitysoftware.com.