ISSN 2250-0987
Shashi Raj et al, UNIASCIT, Vol 2 (2), 2012, 234-238

SESSION INITIATION PROTOCOL BASED STACK DEVELOPMENT FOR USER AGENT CLIENT AND SERVER

Shashi raj. K[1], varadaraju d.v[2], naveen kumar. K[3]

[1 &3] *Department of Electronics and Communication Engineering, REVA Institute of Technology and Management, Kattigenahalli, Yelahanka, Bangalore, Karnataka, India- 560064*
[2]*Department of Electronics and Communication Engineering, The Oxford College of Engineering, Bommanahalli, Bangalore, Karnataka, India -560068*

Abstract **- This document describes Session Initiation Protocol (SIP), an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions include Internet telephone calls, multimedia distribution, and multimedia conferences.**

SIP invitations used to create sessions carry session descriptions that allow participants to agree on a set of compatible media types. SIP makes use of elements called proxy servers to help route requests to the user's current location, authenticate and authorize users for services, implement provider call routing policies, and provide features to users. SIP also provides a registration function that allows users to upload their current locations for use by proxy servers. SIP runs on top of several different transport protocols.

Keywords- **Session Initiation Protocol (SIP), User agent client, User agent server, Backus-Naur Form grammar (BNF), Transaction user.**

I. INTRODUCTION

There are many applications of the Internet that require the creation and management of a session, where a session is considered an exchange of data between an association of participants. The implementation of these applications is complicated by the practices of participants: users may move between endpoints, they may be addressable by multiple names, and they may communicate in several different media – sometimes simultaneously. Numerous protocols have been authored that carry various forms of real-time multimedia session data such as voice, video, or text messages.

The Session Initiation Protocol (SIP) works in concert with these protocols by enabling Internet endpoints to discover one another and to agree on a characterization of a session they would like to share. For locating prospective session participants, and for other functions, SIP enables the creation of an infrastructure of network hosts to which user agents can send registrations, invitations to sessions, and other requests. SIP is an agile, general-purpose tool for creating, modifying, and terminating sessions that works independently of underlying transport protocols and without dependency on the type of session that is being established.

II. OVERVIEW OF SIP FUNCTIONALITY

SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions (conferences) such as Internet telephony calls. SIP can also invite participants to already existing sessions, such as multicast conferences. Media can be added to (and removed from) an existing session. SIP transparently supports name mapping and redirection services, which supports personal mobility users can maintain a single externally visible identifier regardless of their network location.

SIP supports five facets of establishing and terminating multimedia communications:
User location: determination of the end system to be used for communication;
User availability: determination of the willingness of the called party to engage in communications;
User capabilities: determination of the media and media parameters to be used;
Session setup: "ringing", establishment of session parameters at both called and calling party;
Session management: including transfer and termination of sessions, modifying session parameters, and invoking services.

III. SESSION INITIATION PROTOCOL

The fig. 1 below shows the basic operations of Session Initiation Protocol (SIP). SIP is structured as a layered protocol, which means that its behavior is described in terms of a set of fairly independent processing stages with only a loose coupling between each stage. The protocol behavior is described as layers for the purpose of presentation, allowing the description of functions common across elements in a single section. It does not dictate an implementation in any way. When we say that an element "contains" a layer, we mean it is compliant to the set of rules defined by that layer. Not every element specified by the protocol contains every layer. Furthermore, the elements specified by SIP are

logical elements, not physical ones. A physical realization can choose to act as different logical elements, perhaps even on a transaction-by-transaction basis. The lowest layer of SIP is its syntax and encoding. Its encoding is specified using an augmented Backus-Naur Form grammar (BNF). The second layer is the transport layer.

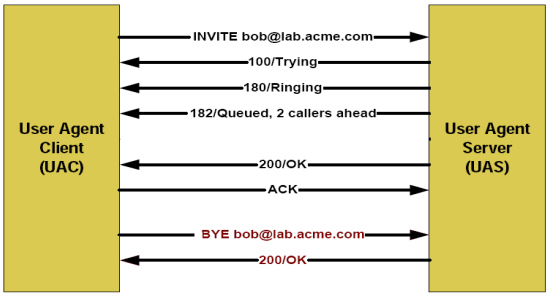

Fig. 1. Basic operation of SIP

It defines how a client sends requests and receives responses and how a server receives requests and sends responses over the network. All SIP elements contain a transport layer. The third layer is the transaction layer. Transactions are a fundamental component of SIP. A transaction is a request sent by a client transaction (using the transport layer) to a server transaction, along with all responses to that request sent from the server transaction back to the client. The transaction layer handles application-layer retransmissions, matching of responses to requests, and application-layer timeouts. Any task that a user agent client (UAC) accomplishes takes place using a series of transactions.

User agents contain a transaction layer, as do stateful proxies. Stateless proxies do not contain a transaction layer. The transaction layer has a client component (referred to as a client transaction) and a server component (referred to as a server transaction), each of which are represented by a finite state machine that is constructed to process a particular request. The layer above the transaction layer is called the transaction user (TU). Each of the SIP entities, except the stateless proxy, is a transaction user. When a TU wishes to send a request, it creates a client transaction instance and passes it the request along with the destination IP address, port, and transport to which to send the request. A TU that creates a client transaction can also cancel it. When a client cancels a transaction, it requests that the server stop further processing, revert to the state that existed before the transaction was initiated, and generate a specific error response to that transaction. This is done with a CANCEL request, which constitutes its own transaction, but references the transaction to be cancelled.

The SIP elements, that is, user agent clients and servers, stateless and stateful proxies and registrars, contain a core that distinguishes them from each other. Cores, except for the stateless proxy, are transaction users.

While the behavior of the UAC and UAS cores depends on the method, there are some common rules for all methods. For a UAC, these rules govern the construction of a request; for a UAS, they govern the processing of a request and generating a response. Since registrations play an important role in SIP, a UAS that handles a REGISTER is given the special name registrar. Certain other requests are sent within a dialog. A dialog is a peer-to-peer SIP relationship between two user agents that persists for some time. The dialog facilitates sequencing of messages and proper routing of requests between the user agents. The INVITE method is the only way defined in this specification to establish a dialog. When a UAC sends a request that is within the context of a dialog, it follows the common UAC rules as discussed in Section 8 but also the rules for mid-dialog requests. The most important method in SIP is the INVITE method, which is used to establish a session between participants. A session is a collection of participants, and streams of media between them, for the purposes of communication.

A. SIP Messages

SIP is a text-based protocol and uses the UTF-8 charset. A SIP message is either a request from a client to a server, or a response from a server to a client. Both Request and Response messages use the basic format of RFC 2822, even though the syntax differs in character set and syntax specifics. Both types of messages consist of a start-line, one or more header fields, an empty line indicating the end of the header fields, and an optional message-body.

```
generic-message = start-line
                        *message-header
                        CRLF
[ message-body ]
start-line = Request-Line /
        Status-Line
```

The start-line, each message-header line, and the empty line MUST be terminated by a carriage-return linefeed sequence (CRLF). Note that the empty line MUST be present even if the message-body is not.

Except for the above difference in character sets, much of SIP's message and header field syntax is identical to HTTP/1.1 specification.

B. Common Message Components

There are certain components of SIP messages that appear in various places within SIP messages.

- SIP and SIPS Uniform Resource Indicators:

A SIP or SIPS URI identifies a communications resource. Like all URIs, SIP and SIPS URIs may be placed in web pages, email messages, or printed literature. They contain sufficient information to initiate and maintain a communication session with the resource. A SIPS URI specifies that the resource be contacted

securely. This means, in particular, that TLS is to be used between the UAC and the domain that owns the URI. From there, secure communications are used to reach the user, where the specific security mechanism depends on the policy of the domain. Any resource described by a SIP URI can be "upgraded" to a SIPS URI by just changing the scheme, if it is desired to communicate with that resource securely.

- S/MIME:

SIP messages carry MIME bodies and the MIME standard includes mechanisms for securing MIME contents to ensure both integrity and confidentiality. Implementers should note, however, that there may be rare network intermediaries (not typical proxy servers) that rely on viewing or modifying the bodies of SIP messages (especially SDP), and that secure MIME may prevent these sorts of intermediaries from functioning.

The certificates that are used to identify an end-user for the purposes of S/MIME differ from those used by servers in one important respect - rather than asserting that the identity of the holder corresponds to a particular hostname, these certificates assert that the holder is identified by an end-user address. This address is composed of the concatenation of the userinfo "@" and domain name portions of a SIP or SIPS URI, most commonly corresponding to a user's address-of-record.

These certificates are also associated with keys that are used to sign or encrypt bodies of SIP messages. Bodies are signed with the private key of the sender (who may include their public key with the message as appropriate), but bodies are encrypted with the public key of the intended recipient. Obviously, senders must have foreknowledge of the public key of recipients in order to encrypt message bodies. Public keys can be stored within a UA on a virtual keying.

Some of the Limitations are- Although these security mechanisms, when applied in a judicious manner, can thwart many threats, there are limitations in the scope of the mechanisms that must be understood by implementers and network operators. One of the primary limitations of using HTTP Digest in SIP is that the integrity mechanisms in Digest do not work very well for SIP. Specifically, they offer protection of the Request-URI and the method of a message, but not for any of the header fields that UAs would most likely wish to secure.

The existing replay protection mechanisms described in RFC 2617 also have some limitations for SIP. The next-nonce mechanism, for example, does not support pipelined requests. The nonce-count mechanism should be used for replay protection. Another limitation of HTTP Digest is the scope of realms. Digest is valuable when a user wants to authenticate themselves to a resource with which they have a pre-existing association, like a service provider of which the user is a customer (which is quite a common scenario and thus Digest provides an extremely useful function). By way of contrast, the scope of TLS is interdomain or multirealm, since certificates are often globally verifiable, so that the UA can authenticate the server with no pre-existing association.

The largest outstanding defect with the S/MIME mechanism is the lack of a prevalent public key infrastructure for end users. If self-signed certificates (or certificates that cannot be verified by one of the participants in a dialog) are used, the SIP-based key exchange mechanism is susceptible to a man-in-the-middle attack with which an attacker can potentially inspect and modify S/MIME bodies. The attacker needs to intercept the first exchange of keys between the two parties in a dialog, remove the existing CMS-detached signatures from the request and response, and insert a different CMS-detached signature containing a certificate supplied by the attacker (but which seems to be a certificate for the proper address-of- record). Each party will think they have exchanged keys with the other, when in fact each has the public key of the attacker.

C. SIP Advantages/Disadvantages

- Simple, scalable, and extensible.
- Requires about four packets for call-setup.
- Provides floor control within a session Cannot provide.
- Minimal capability exchange, enough for IP Telephony.
- Basically runs on UDP. Reliability achieved through retransmissions. Supports TCP also, if UDP is not supported.

IV. APPLICATIONS

SIP has been described as "a simple protocol with profound implications". It addresses many of the major issues of the development of Internet telephony — a technology that is predicted to change the way businesses and people talk to each other. The main applications implemented with SIP are:

• Unified Communication- A SIP session can contain any combination of media (voice, data, video, etc.). These sessions can be modified at any time by adding new parties or by changing the nature of the session. SIP allows browsers to become augmented with multimedia capability. Using SIP, simple, but very powerful, services like click-to-dial become possible. User profiles can be managed through a web interface and voice plug-ins are incorporated into browser technology. SIP uses MIME, the de- facto standard for describing content on the Internet, to convey information about the protocol used to describe the session and has an URL-style addressing system. It uses the Domain Name System (DNS) to deliver requests to the server that can appropriately handle them.

• Unified Messaging- e-mail, voicemail, faxes, and phone messages are accessible from the same box. Alternatively, people use many different devices to communicate. Unified messaging helps people that use different communication devices, media, and

technologies to communicate at any time and under their own control.

• Directory Services- Directory services are to a network what white pages are to the telephone system. They store information about things in the real world, such as people, computers, printers, and so on, as objects with descriptive attributes. People can use the service to look up objects by name; or, like the yellow pages, they can be used to look up services. Network managers use directories to manage user accounts and network resources. From a manager's viewpoint, a directory service is like an
inventory of all the devices on the network. Any device can be located by using a graphic interface or by searching for its name or some properties (e.g., "color printer"). Once located, a manager can control the device (e.g., disable it or block certain users from accessing it). The directory is a central database where all objects and users are managed.

• IP-PBX functionality- Software based IP_PBX that is compliant with the SIP standard can be utilized in a single office setting or multiple office locations, offering flexibility and options for future expansions.
• Instant Messaging (IM) and Presence — because a SIP session can consist of any form of communication, it is SIP Technology Copyright © Ixia, 2004, possible to promote an IM session to a telephone call or even a whiteboard or video session at the click of a button. It is also easy to invite other people to join your session, creating spontaneous conference calls. Using third party call control, a conference service could even check the presence status of people due to join a conference and when all the parties are available it could establish the session by connecting them all to a conference bridge. Presence goes hand-in-hand with evolution of voice services. A network that dynamically updates information about an user's preferences and availability can perform intelligent call routing than today's PSTN or find-me/follow-me services.

• Mobile phones and PDAs- Because SIP client software is lightweight, it can be embedded in mobile phones and PDAs so that these services can cross all platforms. Using SIP as the signaling protocol means that sessions can be established between different devices that then negotiate the appropriate media capability. These devices becomes means of accessing those services associated with a user instead of being closed, proprietary systems.

• Wireless LAN VoIP Telephone Handsets - dedicated portable telephone handsets, supporting Voice over IP on an 802.11 wireless LAN connection. They may use SIP and other proprietary protocols (i.e., Skinny) and may also support wireless telephony protocols (i.e., GSM).

• Desktop Call Management- SIP enables a convergence at the desktop. Voice services can be assimilated into other applications to change the way we use our computers. The information management capabilities of the Internet can be used to transform communication systems and improve productivity. Using SIP features such as user profiling, presence management and instant messaging, third party call control and integration with media, many services can be created by service providers or enterprise IT departments. All the advanced telephony services inherited from the Intelligent Network are supported by SIP. This includes services such as call forwarding, call hold, call waiting, etc.

SIP can be integrated into products such as:

• IP phones.
• Media Gateways.
• Web-enabled telephony portals.
• Internet call centers.
• Soft switches.
• Application servers.

V. CONCLUSION AND FUTURE WORK

The architecture provides reliability and scalability inherent in additional to interoperability with existing SIP infrastructure. The advantages come at the cost of increased call setup latency. We analyze various design alternatives, propose a P2P-SIP architecture using Chord as the underlying DHT, and describe various user location and registration steps in detail. We also present an overview of various advanced services such as offline messaging, conferencing, NAT and firewall traversal and security issues. We are implementing a SIP node for multimedia communication using our SIP C library. We will be doing performance measurement for reliability and scalability on our actual system instead of using simulations. Since the implementation is based on Chord, more simulations may not add any research value to the existing simulation results.
More work is needed in advanced services such as large scale application level multicast conferencing using SIP, distributed reputation system for peers, and PSTN interworking related issues such as authentication and accounting. There should be a reasonable incentive to become a super-node to provide services to other peers. We are working on allowing an internal node inside a firewall and NAT to become a super-node. This reduces the load on public super-nodes, since most of the users typically will be behind some firewall and NAT. Alternatively, the private nodes in a domain can form a secondary P2P overlay connected to the public DHT via a few external connections to reduce the port utilization on the NAT device.

Some of the open questions described in are relevant to P2P-SIP architecture also. Some kind of hybrid system may be implemented that takes the advantages of many different structured P2P algorithms to further reduce the

latency and maintenance cost. For example, there has been recent proposal on one hop lookups for P2P assuming large storage space. Applying this in P2P-SIP is for further study.

Finally, we conclude on a note that unless the SIP servers (proxies, registrars) are widely deployed, we will need SIP based IP tools so that everyone can use the system. Such SIP architecture can be extended to other protocols such as H.323.

REFERENCES

[1]. M. Handley and V. Jacobson, "SDP: session description protocol," RFC 2327, Internet Engineering Task Force, Apr. 1998.
[2]. S. Bradner, "Key words for use in RFCs to indicate requirement levels," RFC 2119, Internet Engineering Task Force, Mar. 1997.
[3]. "Internet message format," RFC 2822, Internet Engineering Task Force, Apr. 2001.
[4]. J. Rosenberg and H. Schulzrinne, "Session initiation protocol (SIP): locating SIP servers," RFC 3263, Internet Engineering Task Force, June 2002.
[5]. T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform resource identifiers (URI): generic syntax," RFC 2396, Internet Engineering Task Force, Aug. 1998.
[6]. F. Yergeau, "UTF-8, a transformation format of ISO 10646," RFC 2279, Internet Engineering Task Force, Jan. 1998.
[7]. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertexttransfer protocol – HTTP/1.1," RFC 2616, Internet Engineering Task Force, June 1999.
[8]. A. Vaha-Sipila, "URLs for telephone calls," RFC 2806, Internet Engineering Task Force, Apr. 2000.
[9]. "Augmented BNF for syntax specifications: ABNF," RFC 2234, Internet Engineering Task Force, Nov. 1997.
[10]. N. Freed and N. Borenstein, "Multipurpose internet mail extensions (MIME) part two: Media types," RFC 2046, Internet Engineering Task Force, Nov. 1996.
[11]. D. Eastlake, S. Crocker, and J. Schiller, "Randomness recommendations for security," RFC 1750, Internet Engineering Task Force, Dec. 1994.
[12]. J. Rosenberg and H. Schulzrinne, "An offer/answer model with session description protocol (SDP)," RFC 3264, Internet Engineering Task Force, June 2002.
[13]. J. Postel, "User datagram protocol," RFC 768, Internet Engineering Task Force, Aug. 1980.
[14]. J. Postel, "DoD standard transmission control protocol," RFC 761, Internet Engineering Task Force, Jan. 1980.
[15]. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream control transmission protocol," RFC 2960, Internet Engineering Task Force, Oct. 2000.

SHASHI RAJ K received Bachelor of Engineering degree in Electronics and Communication Engineering from the Vivekananda Institute of Technology, Bangalore and pursuing Master of Technology degree in Reva Institute of Technology and Management, Bangalore, affiliated to Visvesvaraya Technological University, in the field of VLSI Design and Embedded Systems. His research interests are VLSI based Agent applications, Wireless Communication and Computer Networking. He has published one national conference paper and one international journal. He is a member IEEE (MIEEE, India).

VARADARAJU D V received Bachelor of Engineering degree in Information Science Engineering from the Vivekananda Institute of Technology, Bangalore and pursuing Master of Technology degree in Oxford College of Engineering, Bangalore, affiliated to Visvesvaraya Technological University, in the field of Digital Communication and Networking. His research interests are Wireless Communication and Computer Networking. He has published one national conference paper and one international journal.