

SIP – FUNCTIONALITY AND STRUCTURE OF THE PROTOCOL

Omar Abouabdalla & R. Sureswaran,

*School of Computer Sciences,
Universiti Sains Malaysia,
Penang, Malaysia
Tel: (604) 6594757
Fax: (604) 6594757
Email: omar@cs.usm.my
sures@cs.usm.my*

ABSTRACT

The Session Initiation Protocol (SIP) is an application-level control protocol for setting up, changing and terminating multimedia sessions between participants on IP data networks. In 1999 SIP was approved as an official standard and RFC2543 was published. SIP can enable a range of services, such as Internet telephony, multimedia conferencing, registration and redirection services, and simplifies connecting to VPNs. As its name implies, SIP is responsible for session signaling, a necessary exchange of messages to establish a communication session, be it voice, video or multimedia. In this paper we are going to provide information about the protocol functionality, and also we are going to describe the structure of the protocol.

1.0 INTRODUCTION

The session initiation protocol, or sip standard, is fast becoming one of the hottest topics in telecoms. The protocol's roots extend back to 1996 and predate the widespread emergence of IP telephony [1]. But its real value has only recently been uncovered. SIP traces its roots to several Internet Engineering Task Force (IETF) initiatives, and is tightly linked to Web and e-mail technologies and standards [2]. Sip's main advantage is that it is a short, simple and flexible protocol, which can run over most fixed and wireless networks. Sip uses the internet model and maps it onto the telecoms world, borrowing from existing internet protocols like http, xml and using a url address structure. These email-like addresses are used to identify users rather than the devices they happen to be using.

2.0 OVERVIEW OF SIP FUNCTIONALITY

Sip's strengths are inherent simplicity and flexibility. Sip needs only a small set of required messages and responses, and being independent of the packet layer, its requests can run over a wide variety of network types. SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions (conferences). 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 [3] - 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.

SIP is not a vertically integrated communications system. SIP is rather a component that can be used with other IETF protocols to build a complete multimedia architecture [7]. Typically, these architectures will include protocols such as the Real-time Transport Protocol (RTP) (RFC 1889) for transporting real-time data and providing QoS feedback, the Real-Time streaming protocol (RTSP) (RFC 2326) for controlling delivery of streaming media, the Media Gateway Control Protocol (MEGACO) (RFC 3015) for controlling gateways to the Public Switched Telephone Network (PSTN), and the Session Description Protocol (SDP) for describing multimedia sessions. Therefore, SIP should be used in conjunction with other protocols in order to provide complete services to the users. However, the basic functionality and operation of SIP does not depend on any of these protocols.

2.1 Overview of operation

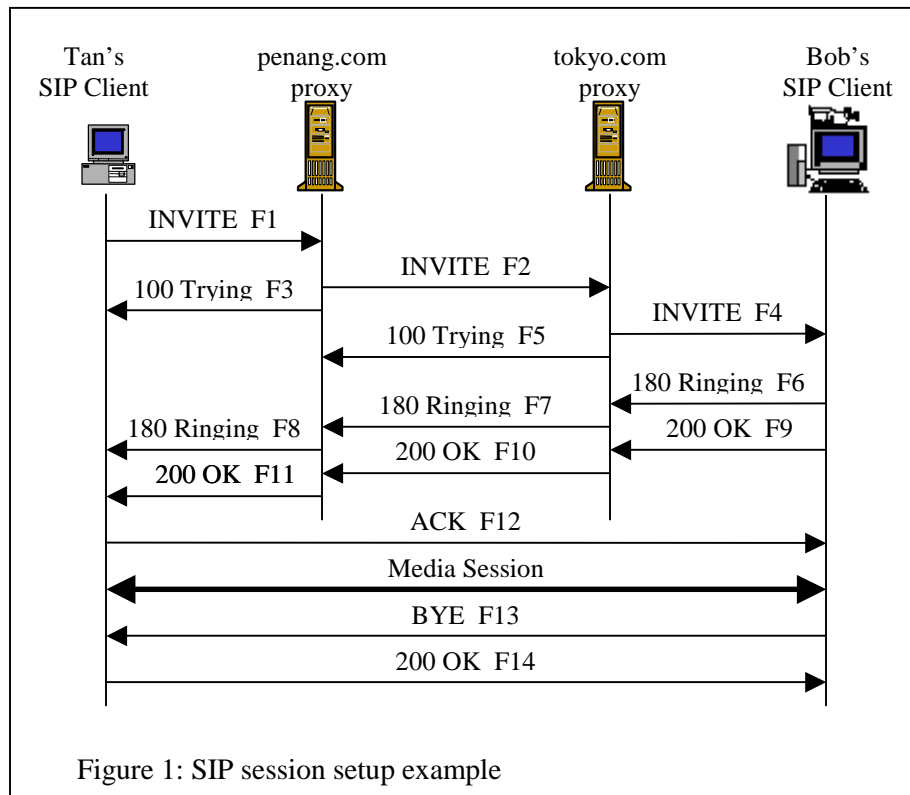
SIP is based on an HTTP-like request/response transaction model [5]. Each transaction consists of a request that invokes a particular method, or function on the server and at least one response. Figure 1 shows a typical example of a SIP message exchange between two users, Tan and Bob. In this example, Tan uses a SIP application on his PC to call Bob on his PC over the Internet. Also shown are two SIP proxy servers that act on behalf of Tan and Bob to facilitate the session establishment.

Tan "calls" Bob using his SIP identity, a type of Uniform Resource Identifier (URI) called a SIP URI. It has a similar form to an email address, typically containing a username and a host name. In this case, it is sip:bob@tokyo.com, where tokyo.com is the domain of Bob's SIP service provider. Tan has a SIP URI of sip:tan@penang.com. Tan might have typed in Bob's URI or perhaps clicked on a hyperlink or an entry in an address book. SIP also provides a secure URI, called a SIPS URI. An example would be sips:bob@tokyo.com. A call made to a SIPS URI guarantees that secure, encrypted transport (namely TLS) is used to carry all SIP messages from the caller to the domain of the callee. From there, the request is sent securely to the callee, but with security mechanisms that depend on the policy of the domain of the callee.

In the example, the transaction begins with Tan's Sip client sending an INVITE request addressed to Bob's SIP URI. INVITE is an example of a SIP method that specifies the action that the requestor (Tan) wants the server (Bob) to take. The INVITE request contains a number of header fields. The ones present in an INVITE include a unique identifier for the call, the destination address, Tan's address, and information about the type of session that Tan wishes to establish with Bob.

Tan's SIP client sends the INVITE to the SIP server that serves Tan's domain (penang.com). The proxy server receives the INVITE request and sends a 100 (Trying) response back to Tan's. The 100 (Trying) response indicates that the INVITE has been received and that the proxy is working on his behalf to route the INVITE to the destination. The penang.com proxy server locates the proxy server at tokyo.com, and obtains the IP address of the tokyo.com proxy server and forwards, or proxies, the INVITE request there. The tokyo.com proxy server receives the INVITE and responds with a 100 (Trying) response back to the penang.com proxy server to indicate that it has received the INVITE and is processing the request. The proxy server consults a database that contains the current IP address of Bob.

Bob's SIP client receives the INVITE and alerts Bob to the incoming call from Tan so that Bob can decide whether to answer the call. Bob's SIP phone indicates this in a 180 (Ringing) response, which is routed back through the two proxies in the reverse direction. When Tan's SIP client receives the 180 (Ringing) response, it passes this information to Tan. In this example, Bob decides to answer the call. When he clicks answer, his SIP client sends a 200 (OK) response to indicate that the call has been answered. The 200 (OK) contains a message body with the Session Description Protocol (SDP) media description of the type of session that Bob is willing to establish with Tan. As a result, there is a two-phase exchange of SDP messages: Tan sent one to Bob, and Bob sent one back to Tan. This two-phase exchange provides basic negotiation capabilities and is based on a simple offer/answer model of SDP exchange. If Bob did not wish to answer the call or was busy on another call, an error response would have been sent instead of the 200 (OK), which would have resulted in no media session being established.



Finally, Tan's SIP client sends an acknowledgement message, ACK to Bob's SIP client to confirm the reception of the final response (200 (OK)). In this example, the ACK is sent directly from Tan's SIP client to Bob's SIP client, bypassing the two proxies. This occurs because the endpoints

have learned each other's address from the Contact header fields through the INVITE/200 (OK) exchange, which was not known when the initial INVITE was sent. The lookups performed by the two proxies are no longer needed, so the proxies drop out of the call flow. This completes the INVITE/200/ACK three-way handshake used to establish SIP sessions.

Tan and Bob's media session has now begun, and they send media packets using the format to which they agreed in the exchange of SDP. In general, the end-to-end media packets take a different path from the SIP signaling messages. During the session, either Tan or Bob may decide to change the characteristics of the media session. At the end of the call, Bob disconnects (hangs up) first and generates a BYE message. This BYE is routed directly to Tan's SIP client, again bypassing the proxies. Tan confirms receipt of the BYE with a 200 (OK) response, which terminates the session and the BYE transaction.

Registration is another common operation in SIP. Registration is one way that the tokyo.com server can learn the current location of Bob. Upon initialization, Bob's SIP client sends REGISTER messages to a server in the tokyo.com domain known as a SIP registrar. The REGISTER messages associate Bob's SIP or SIPS URI (sip:bob@biloxi.com) with the machine into which he is currently logged. The registrar writes this association, also called a binding, to a database, called the location service, where it can be used by the proxy.

3.0 SIP STRUCTURE

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 lowest layer of SIP is its syntax and encoding. Its encoding is specified using an augmented Backus-Naur Form grammar (BNF) defined in RFC 2234 [4].

The second layer is the transport layer. 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. 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 [5]. 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, 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.

3.1 Definitions

The following terms have special significance for SIP.

Address-of-Record: An address-of-record (AOR) is a SIP or SIPS URI that points to a domain with a location service that can map the URI to another URI where the user might be available.

Back-to-Back User Agent: A back-to-back user agent (B2BUA) is a logical entity that receives a request and processes it as a user agent server (UAS). In order to determine how the request should be answered, it acts as a user agent client (UAC) and generates requests. Unlike a proxy server, it maintains dialog state and must participate in all requests sent on the dialogs it has established.

Call: A call is an informal term that refers to some communication between peers, generally set up for the purposes of a multimedia conversation.

Client: A client is any network element that sends SIP requests and receives SIP responses. Clients may or may not interact directly with a human user. User agent clients and proxies are clients.

Conference: A multimedia session that contains multiple participants.

Dialog: A dialog is a peer-to-peer SIP relationship between two UAs that persists for some time. A dialog is established by SIP messages. A dialog is identified by a call identifier, local tag, and a remote tag [5].

Header: A header is a component of a SIP message that conveys information about the message. It is structured as a sequence of header fields.

Header Field: A header field is a component of the SIP message header. A header field can appear as one or more header field rows. Header field rows consist of a header field name and zero or more header field values. Multiple header field values on a given header field row are separated by commas. Some header fields can only have a single header field value, and as a result, always appear as a single header field row.

Header Field Value: A header field value is a single value; a header field consists of zero or more header field values [6].

Home Domain: The domain providing service to a SIP user. Typically, this is the domain present in the URI in the AOR of a registration.

Initiator, Calling Party, Caller: The party initiating a session (and dialog) with an INVITE request. A caller retains this role from the time it sends the initial INVITE that established a dialog until the termination of that dialog.

Invitation: An INVITE request.

Invitee, Invited User, Called Party, Callee: The party that receives an INVITE request for the purpose of establishing a new session. A callee retains this role from the time it receives the INVITE until the termination of the dialog established by that INVITE.

Message: Data sent between SIP elements as part of the protocol. SIP messages are either requests or responses.

Outbound Proxy: A proxy that receives requests from a client, even though it may not be the server resolved by the Request-URI. Typically, a UA is manually configured with an outbound proxy, or can learn about one through auto-configuration protocols.

Proxy, Proxy Server: An intermediary entity that acts as both a server and a client for the purpose of making requests on behalf of other clients. A proxy server primarily plays the role of routing, which means its job is to ensure that a request is sent to another entity "closer" to the targeted user.

Registrar: A registrar is a server that accepts REGISTER requests and places the information it receives in those requests into the location service for the domain it handles.

Regular Transaction: A regular transaction is any transaction with a method other than INVITE, ACK, or CANCEL.

Request: A SIP message sent from a client to a server, for the purpose of invoking a particular operation.

Response: A SIP message sent from a server to a client, for indicating the status of a request sent from the client to the server.

Server: A server is a network element that receives requests in order to service them and sends back responses to those requests. Examples of servers are proxies, user agent servers, redirect servers, and registrars.

Session: From the SDP specification: "A multimedia session is a set of multimedia senders and receivers and the data streams flowing from senders to receivers. A multimedia conference is an example of a multimedia session." (RFC 2327 [1]).

SIP Transaction: A SIP transaction occurs between a client and a server and comprises all messages from the first request sent from the client to the server up to a final response sent from the server to the client.

Stateful Proxy: A logical entity that maintains the client and server transaction state machines defined by this specification during the processing of a request, also known as a transaction stateful proxy.

Stateless Proxy: A logical entity that does not maintain the client or server transaction state machines defined in this specification when it processes requests.

Transaction User (TU): The layer of protocol processing that resides above the transaction layer. Transaction users include the UAC core, UAS core, and proxy core.

User Agent Client (UAC): A user agent client is a logical entity that creates a new request, and then uses the client transaction state machinery to send it. The role of UAC lasts only for the duration of that transaction. In other words, if a piece of software initiates a request, it acts as a UAC for the duration of that transaction. If it receives a request later, it assumes the role of a user agent server for the processing of that transaction.

User Agent Server (UAS): A user agent server is a logical entity that generates a response to a SIP request. The response accepts, rejects, or redirects the request. This role lasts only for the duration of that transaction. In other words, if a piece of software responds to a request, it acts as a UAS for the duration of that transaction. If it generates a request later, it assumes the role of a user agent client for the processing of that transaction.

User Agent (UA): A logical entity that can act as both a user agent client and user agent server.

4.0 CONCLUSION

This paper gives an overview of Session Initiation Protocol (SIP) functionality, an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. It also shows a simple example of initiating a session, and shows some definitions to some of the SIP elements.

Future work will be to bridge between SIP and other multimedia control protocols like RSW control protocol.

5.0 REFERENCES

[1] George Malim, "Opportunity knocks", Communications International, London, Jul 2002, pp. 16.

[2] Ian Grobel, "SIP is a key part in multimedia sessions", Network World, Framingham, Aug 2002, Vol. 19, Iss. 32, pp. 35.

[3] R. Pandya, "Emerging mobile and personal communication systems", IEEE Communications Magazine, June 1995, Vol. 33, pp. 44-52.

[4] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[5] Handley, M., Schulzrinne, H., Schooler, E. and J. Rosenberg, "SIP: Session Initiation Protocol", RFC 2543, March 1999.

[6] Troost, R., Dorner, S. and K. Moore, "Communicating Presentation Information in Internet Messages: The Content-Disposition Header Field", RFC 2183, August 1997.

[7] E. M. Schooler, "Case study: multimedia conference control in a packet-switched teleconferencing system," Journal of Internetworking: Research and Experience, June 1993, Vol. 4, pp. 99-120.