Session Initiation Protocol

From Wikipedia, the free encyclopedia

The **Session Initiation Protocol** (**SIP**) is a communications protocol for signaling and controlling multimedia communication sessions in applications of Internet telephony for voice and video calls, in private IP telephone systems, as well as in instant messaging over Internet Protocol (IP) networks.

The protocol defines the methodology of SIP communications and the specific format of messages exchanged for cooperation of the participants in multimedia sessions. A call established with SIP may consist of multiple media streams, but none are required in text messaging, for which the payload is carried directly in the SIP message. SIP is designed to be independent of the underlying transport layer protocol, and can be used with the User Datagram Protocol (UDP), the Transmission Control Protocol (TCP), and the Stream Control Transmission Protocol (SCTP). It is a text-based protocol, incorporating many elements of the Hypertext Transfer Protocol (HTTP) and the Simple Mail Transfer Protocol (SMTP).[1]

SIP works in conjunction with several other protocols that specify and carry the session media. Media type and parameter negotiation and media setup is performed with the Session Description Protocol (SDP), which is carried as payload in SIP messages. For the transmission of media streams (voice, video) SIP typically employs the Real-time Transport Protocol (RTP) or the Secure Real-time Transport Protocol (SRTP). For secure transmissions of SIP messages over insecure network links, the protocol may be encrypted with Transport Layer Security (TLS).

Contents

History

SIP was originally designed by Mark Handley, Henning Schulzrinne, Eve Schooler and Jonathan Rosenberg in 1996. The protocol was standardized as RFC 2543 in 1999. In November 2000, SIP was accepted as a 3GPP signaling protocol and permanent element of the IP Multimedia Subsystem (IMS) architecture for IP-based streaming multimedia services in cellular networks. In June 2002 the specification was revised in RFC 3261[2] and various extensions and clarifications have been published since.[3]

A motivating goal for SIP was to provide a signaling and call setup protocol for IP-based communications that can support the call processing functions and features present in the public switched telephone network (PSTN) and the vision of supporting new multimedia applications. It has been extended for video conferencing, streaming multimedia distribution, instant messaging, presence information, file transfer, fax over IP and online games.[4][5][6]

SIP is distinguished by its proponents for having roots in the Internet community rather than in the telecommunications industry. SIP has been standardized primarily by the IETF, while other protocols, such as H.323, have traditionally been associated with the International Telecommunication Union (ITU).

Protocol operation

SIP is only involved in the signaling portion of a media communication session, primarily used to set up and terminate voice or video calls. SIP can be used to establish two-party (unicast) or multiparty (multicast) sessions. It also allows modification of existing calls. The modification can involve changing addresses or ports, inviting more participants, and adding or deleting media streams. SIP has also found applications in messaging applications, such as instant messaging, and event subscription and notification.

SIP works in concert with several other protocols to specify the media format and coding, and the protocol for communicating the media once the call is set up. For call setup, the body of a SIP message contains a Session Description Protocol (SDP) data unit, which specifies the media format, codec and media communication protocol. Voice and video media is typically specified to be communicated between the terminals using the Real-time Transport Protocol (RTP) or Secure Real-Time Transport Protocol (SRTP).[7][8]

Each resource of a SIP network, such as a user agent or a voicemail box, is identified by a Uniform Resource Identifier (URI), which follows the general standard syntax also used in Web services and e-mail.[9] The URI scheme used for SIP is *sip* and a typical SIP URI has the form *sip:username@domainname* or *sip:username@hostport*, where *domainname* requires DNS SRV records to locate the servers for SIP domain while *hostport* can be an IP address or a fully qualified domain name of the host and port. If secure transmission is required, the scheme *sips* is used.[10][11]

SIP employs design elements similar to the HTTP request/response transaction model.[12] Each transaction consists of a client request that invokes a particular method or function on the server and at least one response. SIP reuses most of the header fields, encoding rules and status codes of HTTP, providing a readable text-based format.

SIP can be carried by several transport layer protocols including the Transmission Control Protocol (TCP), the User Datagram Protocol (UDP) or the Stream Control Transmission Protocol (SCTP).[13][14] SIP clients typically use TCP or UDP on port numbers 5060 or 5061 for SIP traffic to servers and other endpoints. Port 5060 is commonly used for non-encrypted signaling traffic whereas port 5061 is typically used for traffic encrypted with Transport Layer Security (TLS).

SIP-enabled telephony networks often implement call processing features of Signaling System 7 (SS7), although the two protocols themselves are very different. SS7 is a centralized protocol, characterized by a complex central network architecture and dumb endpoints (traditional telephone handsets). SIP is a client-server protocol of equipotent peers. SIP features are implemented in the communicating endpoints, while many traditional SS7 architectures use the suite only between switching centers.

Network elements

The network elements that use the Session Initiation Protocol for communication are called *SIP user agents*. Each user agent (UA) performs the function of a user agent client (UAC) when it is requesting a service function, and that of a user agent server (UAS) when responding to a request. Thus, any two SIP endpoints may in principle operate without any intervening SIP infrastructure. However, for network operational reasons, for provisioning public services to users, and for directory services, SIP defines several specific types of network server elements. Each of these service elements also communicates within the client-server model implemented in user agent clients and servers.

User agent

A user agent is a logical network end-point used to create or receive SIP messages. The user agent manages SIP sessions. As a client (UAC), it sends SIP requests, and as a server (UAS) it receives requests and returns a SIP response. Unlike other network protocols that fix the roles of client and server, e.g., in HTTP, in which a web browser only acts as a client, and never as a server, SIP requires both peers to implement both roles. The roles of UAC and

UAS only last for the duration of a SIP transaction.[5]

A SIP phone is an IP phone that implements client and server functions of a SIP user agent and provides the traditional call functions of a telephone, such as dial, answer, reject, call hold, and call transfer.[15][16] SIP phones may be implemented as a hardware device or as a softphone. As vendors increasingly implement SIP as a standard telephony platform, the distinction between hardware-based and software-based SIP phones is blurred and SIP elements are implemented in the basic firmware functions of many IP-capable devices.

In SIP, as in HTTP, the user agent may identify itself using a message header field (*User-Agent*), containing a text description of the software, hardware, or the product name. The user agent field is sent in request messages, which means that the receiving SIP server can evaluate this information to perform device-specific configuration or feature activation. Operators of SIP network elements sometimes store this information in customer account portals,[17] where it can be useful in diagnosing SIP compatibility problems or display of service status.

Proxy server

A proxy server is a network server with UAC and UAS components that functions as an intermediary entity for the purpose of performing requests on behalf of other network elements. A proxy server primarily plays the role of routing, meaning that its job is to ensure that a request is sent to another entity closer to the targeted user. Proxies are also useful for enforcing policy, such as for determining whether a user is allowed to make a call. A proxy interprets, and, if necessary, rewrites specific parts of a request message before forwarding it.

Registrar

A registrar is a SIP endpoint that provides a location service. It accepts REGISTER requests, recording the address and other parameters from the user agent. For subsequent requests it provides an essential means to locate possible communication peers on the network. The location service links one or more IP addresses to the SIP URI of the registering agent. Multiple user agents may register for the same URI, with the result that all registered user agents receive the calls to the URI.

SIP user agent registration to SIP registrar with authentication.

Call flow through redirect server and proxy.

SIP registrars are logical elements, and are often co-located with SIP proxies. To improve network scalability, location services may instead be located with a redirect server.

Redirect server

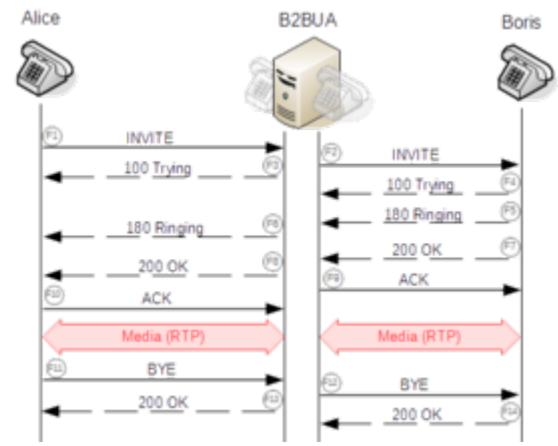
A redirect server is a user agent server that generates 3xx (redirection) responses to requests it receives, directing the client to contact an alternate set of URIs. A redirect server allows proxy servers to direct SIP session invitations to external domains.

Session border controller

Session border controllers serve as middle boxes between UA and SIP servers for various types of functions, including network topology hiding and assistance in NAT traversal.

Gateway

Gateways can be used to interconnect a SIP network to other networks, such as the public switched telephone network, which use different protocols or technologies.



Establishment of a session through a back-to-back user agent.

SIP messages

SIP is a text-based protocol with syntax similar to that of HTTP. There are two different types of SIP messages: requests and responses. The first line of a request has a *method*, defining the nature of the request, and a Request-URI, indicating where the request should be sent.[18] The first line of a response has a *response code*.

Requests

Requests initiate a SIP transaction between two SIP entities for establishing, controlling, and terminating sessions. Critical methods include the following.

- INVITE: Used to establish a dialog with media exchange between user agents.
- BYE: Terminates an existing session.
- REGISTER: The method implements a location service for user agents, which indicate their address information to the server.

Responses

Responses are sent by the user agent server indicating the result of a received request. Several classes of responses are recognized, determined by the numerical range of result codes:[19]

- 1xx: Provisional responses to requests indicate the request was valid and is being processed.
- 2xx: 200-level responses indicate a successful completion of the request. As a response to an INVITE, it indicates a call is established.
- 3xx: This group indicates a redirection is needed for completion of the request. The

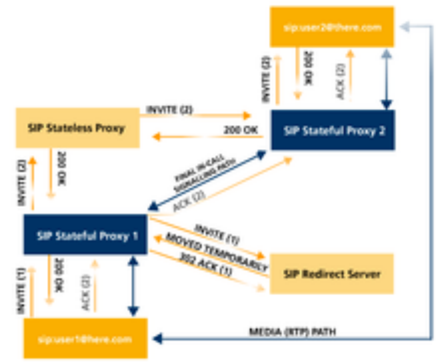
request has to be completed with a new destination.
- 4xx: The request contained bad syntax or cannot be fulfilled at the server.
- 5xx: The server failed to fulfill an apparently valid request.
- 6xx: This is a global failure, as the request cannot be fulfilled at any server.

Transactions

SIP defines a transaction mechanism to control the exchanges between participants and deliver messages reliably. A transaction is a state of a session, which is controlled by various timers. Client transactions send requests and server transactions respond to those requests with one or more responses. The responses may include provisional responses with a response code in the form *1xx*, and one or multiple final responses (2xx – 6xx).

Transactions are further categorized as either type *Invite* or type *Non-Invite*. Invite transactions differ in that they can establish a long-running conversation, referred to as a *dialog* in SIP, and so include an acknowledgment (ACK) of any non-failing final response, e.g., *200 OK*.

Because of these transactional mechanisms, unreliable transport protocols, such as the User Datagram Protocol (UDP), are sufficient for SIP operation.



Example: User1's UAC uses an *Invite Client Transaction* to send the initial INVITE (1) message. If no response is received after a timer controlled wait period the UAC may chose to terminate the transaction or retransmit the INVITE. Once a response is received, User1 is confident the INVITE was delivered reliably. User1's UAC must then acknowledge the response. On delivery of the ACK (2) both sides of the transaction are complete. In this case, a dialog may have been established.[20]

Instant messaging and presence

The Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE) is the SIP-based suite of standards for instant messaging and presence information. MSRP (Message Session Relay Protocol) allows instant message sessions and file transfer.

Conformance testing

The SIP developer community meets regularly at conferences organized by SIP Forum to test interoperability of SIP implementations.[21] The TTCN-3 test specification language, developed by a task force at ETSI (STF 196), is used for specifying conformance tests for SIP implementations.[22]

Performance testing

When developing SIP software or deploying a new SIP infrastructure, it is very important to test capability of servers and IP networks to handle certain call load: number of concurrent calls and number of calls per second. SIP performance tester software is used to simulate SIP and RTP traffic to see if the server and IP network are stable under the call load.[23] The software measures performance indicators like answer delay, answer/seizure ratio, RTP jitter and packet loss, round-trip delay time.

Applications

A *SIP connection* is a marketing term for voice over Internet Protocol (VoIP) services offered by many Internet telephony service providers (ITSPs). The service provides routing of telephone calls from a client's private branch exchange (PBX) telephone system to the public switched telephone network (PSTN). Such services may simplify corporate information system infrastructure by sharing Internet access for voice and data, and removing the cost for Basic Rate Interface (BRI) or Primary Rate Interface (PRI) telephone circuits.

SIP trunking is similar marketing term preferred for when the service is used to simplifies a telecom infrastructure by sharing the carrier access circuit for voice, data and Internet traffic, and removing the need for Primary Rate Interface (PRI) circuits.[24][25]

SIP-enabled video surveillance cameras can initiate calls to alert the operator of events, such as motion of objects in a protected area.

SIP is used in audio over IP for broadcasting applications where it provides an interoperable means for audio interfaces from different manufacturers to make connections with one another.[26]

Implementations

The U.S. National Institute of Standards and Technology (NIST), Advanced Networking Technologies Division provides a public-domain Java implementation[27] that serves as a reference implementation for the standard. The implementation can work in proxy server or user agent scenarios and has been used in numerous commercial and research projects. It supports RFC 3261 in full and a number of extension RFCs including RFC 6665 (event notification) and RFC 3262 (reliable provisional responses).

Numerous other commercial and open-source SIP implementations exist. See List of SIP software.

SIP-ISUP interworking

SIP-I, or the Session Initiation Protocol with encapsulated ISUP, is a protocol used to create, modify, and terminate communication sessions based on ISUP using SIP and IP networks. Services using SIP-I include voice, video telephony, fax and data. SIP-I and SIP-T[28] are two protocols with similar features, notably to allow ISUP messages to be transported over SIP networks. This preserves all of the detail available in the ISUP header, which is important as there are many country-specific variants of ISUP that have been implemented over the last 30 years, and it is not always possible to express all of the same detail using a native SIP message. SIP-I was defined by the ITU-T, whereas SIP-T was defined via the IETF RFC route.[29]

Encryption

Concerns about the security of calls via the public Internet have been addressed by encryption of the SIP protocol for secure transmission. The URI scheme *sips* is used to mandate that each hop over which the request is forwarded up to the target domain must be secured with Transport Layer Security (TLS). The last hop from the proxy of the target domain to the user agent has to be secured according to local policies. TLS protects against attackers who try to listen on the signaling link but it does not provide end-to-end security to prevent espionage and law enforcement interception, as the encryption is only hop-by-hop and every single intermediate proxy has to be trusted.

End-to-end security may also be achieved with secure tunneling and IPsec, but most service providers that offer secure connections use TLS for securing signaling. The relationship between SIP (port 5060) and SIPS (port 5061), is similar to HTTP and HTTPS, and uses URIs in the form *sips:user@example.com*. The media streams, which occur on different connections to the signaling stream, may be encrypted with SRTP. The key exchange for SRTP is performed with SDES (RFC 4568), or with ZRTP (RFC 6189), which can automatically upgrade RTP to SRTP using dynamic key exchange, and a verification phrase. One may also add a MIKEY (RFC 3830) exchange to SIP to determine session keys for use with SRTP.

See also

- Voice over IP
- Rendezvous protocol
- Peer-to-peer SIP
- Computer telephony integration (CTI)
- Computer-supported telecommunications applications (CSTA)
- H.323 protocols H.225.0 and H.245
- IP Multimedia Subsystem (IMS)
- Extensions to the Session Initiation Protocol for the IP Multimedia Subsystem
- Media Gateway Control Protocol (MGCP)
- Message Session Relay Protocol (MSRP)
- Mobile VoIP
- MSCML (Media Server Control Markup Language)
- Network convergence
- RTP audio video profile
- SIGTRAN (Signaling Transport)
- SIP trunking
- SIP provider
- Skinny Client Control Protocol (SCCP)
- XIMSS (XML Interface to Messaging, Scheduling, and Signaling)
- ZRTP

References

1. Johnston, Alan B. (2004). *SIP: Understanding the Session Initiation Protocol, Second Edition*. Artech House. ISBN 1-58053-168-7.
2. "SIP core working group charter" (http://www.ietf.org/dyn/wg/charter/sipcore-charter.html). Ietf.org. 2010-12-07. Retrieved 2011-01-11.
3. "Search Internet-Drafts and RFCs" (https://datatracker.ietf.org/doc/search/?name=SIP&rfcs=on&sort=date). Internet Engineering Task Force.
4. "What is SIP?" (http://www.networkworld.com/article/2332980/lan-wan/what-is-sip-.html). Network World. May 11, 2004.
5. "RFC 3261 – SIP: Session Initiation Protocol" (https://tools.ietf.org/html/rfc3261). IETF. 2002.
6. Margaret Rouse. "Session Initiation Protocol (SIP)" (http://searchunifiedcommunications.techtarget.com/definition/Session-Initiation-Protocol). TechTarget.
7. Johnston, Alan B. (2004). *SIP: Understanding the Session Initiation Protocol, Second Edition*. Artech House. ISBN 1-58053-168-7.
8. Coll, Eric (2016). *Telecom 101*. Teracom Training Institute. pp. 77–79. ISBN 9781894887038.
9. RFC 3986, *Uniform Resource Identifiers (URI): Generic Syntax*, IETF, The Internet Society (2005)
10. Miikka Poikselkä et al. 2004.
11. Brian Reid & Steve Goodman 2015.
12. William Stallings, p.209
13. RFC 4168, *The Stream Control Transmission Protocol (SCTP) as a Transport for the Session Initiation Protocol (SIP)*, IETF, The Internet Society (2005)
14. Montazerolghaem, Ahmadreza; Hosseini Seno, Seyed Amin; Yaghmaee, Mohammad Hossein; Tashtarian, Farzad (2016-06-01). "Overload mitigation mechanism for VoIP networks: a transport layer approach based on resource management" (http://onlinelibrary.wiley.com/doi/10.1002/ett.3038/abstract). *Transactions on Emerging Telecommunications Technologies*. **27** (6): 857–873. ISSN 2161-3915 (https://www.worldcat.org/issn/2161-3915). doi:10.1002/ett.3038 (https://doi.org/10.1002%2Fett.3038).
15. Azzedine (2006). *Handbook of algorithms for wireless networking and mobile computing* (https://books.google.com/books?id=b8oisvv6fDAC&pg=PT774). CRC Press. p. 774. ISBN 978-1-58488-465-1.
16. Porter, Thomas; Andy Zmolek; Jan Kanclirz; Antonio Rosela (2006). *Practical VoIP Security* (https://books.google.com/books?id=BYxdyekyRlwC&pg=PA76). Syngress. pp. 76–77. ISBN 978-1-59749-060-3.
17. "User-Agents We Have Known " (http://web.archive.org/web/20110716170218/http://www.voipuser.org/forum_topic_14998.html)VoIP User.org
18. Stallings, p.214
19. Stallings, pp.216-217
20. James Wright. "SIP - An Introduction" (http://www.konnetic.com/Documents/KonneticSIPIntroduction.pdf) (PDF). Konnetic. Retrieved 2011-01-11.
21. http://www.sipit.net/

22. Experiences of Using TTCN-3 for Testing SIP and also OSP (http://portal.etsi.org/ptcc/downloads/TTCN3SIPOSP.pdf) Archived (https://web.archive.org/web/20140330061038/http://portal.etsi.org/ptcc/downloads/TTCN3SIPOSP.pdf) March 30, 2014, at the Wayback Machine.
23. "Performance and Stress Testing of SIP Servers, Clients and IP Networks" (http://startrinity.com/VoIP/TestingSipPbxSoftswitchServer.aspx). StarTrinity. 2016-08-13.
24. "AT&T Discusses Its SIP Peering Architecture" (http://sip-trunking.tmcnet.com/topics/enterprise-voip/articles/109840-att-discusses-its-sip-peering-architecture.htm). *sip-trunking.tmcnet.com*. Retrieved 2017-03-20.
25. "From IIT VoIP Conference & Expo: AT&T SIP transport PowerPoint slides" (http://hdvoicenews.com/2010/10/18/from-iit-voip-conference-expo-att-sip-transport-powerpoint-slides/). *HD Voice News*. 2010-10-19. Retrieved 2017-03-20.
26. Jonsson, Lars; Mathias Coinchon (2008). "Streaming audio contributions over IP" (http://tech.ebu.ch/webdav/site/tech/shared/techreview/trev_2008-Q1_coinchon.pdf) (PDF). *EBU Technical Review*. Retrieved 2010-12-27.
27. "JAIN SIP project" (http://java.net/projects/jsip). Retrieved 2011-07-26.
28. "RFC3372: SIP-T Context and Architectures" (http://www.ietf.org/rfc/rfc3372.txt). September 2002. Retrieved 2011-01-11.
29. White Paper: "Why SIP-I? A Switching Core Protocol Recommendation" (http://www.4gamericas.org/documents/3G_Americas_SIP-I_White_Paper_August_2007-FINAL.pdf)

Bibliography

- Brian Reid; Steve Goodman (22 January 2015), *Exam Ref 70-342 Advanced Solutions of Microsoft Exchange Server 2013 (MCSE)*, Microsoft Press, p. 24, ISBN 978-0-73-569790-4
- Miikka Poikselkä; Georg Mayer; Hisham Khartabil; Aki Niemi (19 November 2004), *The IMS: IP Multimedia Concepts and Services in the Mobile Domain*, John Wiley & Sons, p. 268, ISBN 978-0-47-087114-0

External links

- Computers/Internet/Protocols/SIP/ (https://dmoztools.net/Computers/Internet/Protocols/SIP/) at DMOZ
- IANA: SIP Parameters (http://www.iana.org/assignments/sip-parameters)
- IANA: SIP Event Types Namespace (http://www.iana.org/assignments/sip-events/sip-events.xhtml)