Session Initiation Protocol

The **Session Initiation Protocol** (**SIP**) is a signaling protocol used for initiating, maintaining, and terminating real-time sessions that include voice, video and messaging applications.[1] SIP is used for signaling and controlling multimedia communication sessions in applications of Internet telephony for voice and video calls, in private IP telephone systems, in instant messaging over Internet Protocol (IP) networks as well as mobile phone calling over LTE (VoLTE).

The protocol defines the specific format of messages exchanged and the sequence of communications for cooperation of the participants. SIP is a text-based protocol, incorporating many elements of the Hypertext Transfer Protocol (HTTP) and the Simple Mail Transfer Protocol (SMTP).[2] A call established with SIP may consist of multiple media streams, but no separate streams are required for applications, such as text messaging, that exchange data as payload in the SIP message.

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

Contents

History

SIP was originally designed by Mark Handley, Henning Schulzrinne, Eve Schooler and Jonathan Rosenberg in 1996 to facilitate establishing multicast multimedia sessions on the Mbone. The protocol was standardized as RFC 2543 (https://datatracker.ietf.org/doc/html/rfc2543) 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 (https://datatracker.ietf.org/doc/html/rfc3261)[3] and various extensions and clarifications have been published since.[4]

SIP was designed to provide a signaling and call setup protocol for IP-based communications supporting the call processing functions and features present in the public switched telephone network (PSTN) with a vision of supporting new multimedia applications. It has been extended for video conferencing, streaming media distribution, instant messaging, presence information, file transfer, Internet fax and online games.[1][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 Internet Engineering Task Force (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 operations of a media communication session and is 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 conjunction with several other protocols that specify the media format and coding and that carry 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 streams are typically carried between the terminals using the Real-time Transport Protocol (RTP) or Secure Real-time Transport Protocol (SRTP).[2][7]

Every resource of a SIP network, such as user agents, call routers, and voicemail boxes, are identified by a Uniform Resource Identifier (URI). The syntax of the URI follows the general standard syntax also used in Web services and e-mail.[8] 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.[9][10]

SIP employs design elements similar to the HTTP request and response transaction model.[11] 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 Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Stream Control Transmission Protocol (SCTP).[12][13] 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-based telephony networks often implement call processing features of Signaling System 7 (SS7), for which special SIP protocol extensions exist, 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 the traditional SS7 architecture is in use 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.[14]

User agent

A user agent is a logical network endpoint that sends or receives SIP messages and manages SIP sessions. User agents have client and server components. The user agent client (UAC) sends SIP requests. The user agent server (UAS) 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 communications devices such as smartphones.

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 in the 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 call routing; it sends SIP requests to another entity closer to it destination. 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.

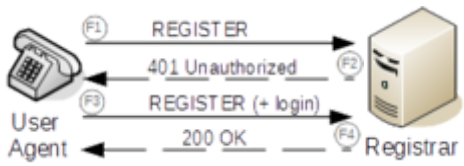
SIP proxy servers that route messages to more than one destination are called forking proxies. The forking of a SIP request establishes multiple dialogs from the single request. Thus, a call may be answered from one of multiple SIP endpoints. For identification of multiple dialogs, each dialog has an identifier with contributions from both endpoints.

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.

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.

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.

Session border controller

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

Gateway

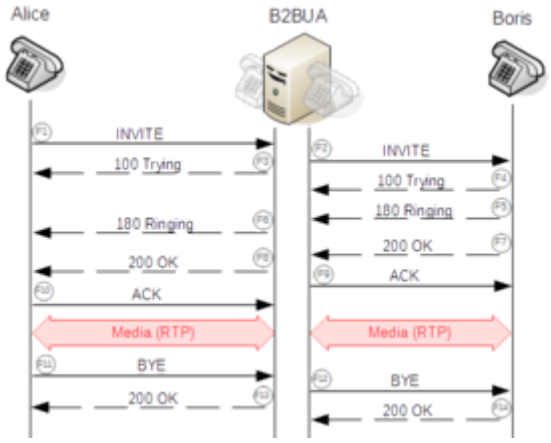
Gateways can be used to interconnect a SIP network to other networks, such as the PSTN, which use different protocols or technologies.

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 functionality of the protocol. They are sent by a user agent client to the server, and are answered with one or more SIP responses, which return a result code of the transaction, and generally indicate the success, failure, or other state of the transaction.



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

SIP requests

Request name	Description	Notes	RFC references
REGISTER	Register the URI listed in the To-header field with a location server and associates it with the network address given in a *Contact* header field.	The command implements a location service.	RFC 3261 (https://datatracker.ietf.org/doc/html/rfc3261)
INVITE	Initiate a dialog for establishing a call. The request is sent by a user agent client to a user agent server.	When sent during an established dialog (*reinvite*) it modifies the sessions, for example placing a call on hold.	RFC 3261 (https://datatracker.ietf.org/doc/html/rfc3261)
ACK	Confirm that an entity has received a final response to an INVITE request.		RFC 3261 (https://datatracker.ietf.org/doc/html/rfc3261)
BYE	Signal termination of a dialog and end a call.	This message may be sent by either endpoint of a dialog.	RFC 3261 (https://datatracker.ietf.org/doc/html/rfc3261)
CANCEL	Cancel any pending request.	Usually means terminating a call while it is still ringing, before answer.	RFC 3261 (https://datatracker.ietf.org/doc/html/rfc3261)
UPDATE	Modify the state of a session without changing the state of the dialog.		RFC 3311 (https://datatracker.ietf.org/doc/html/rfc3311)
REFER	Ask recipient to issue a request for the purpose of call transfer.		RFC 3515 (https://datatracker.ietf.org/doc/html/rfc3515)
PRACK	Provisional acknowledgement.	PRACK is sent in response to provisional response (1xx).	RFC 3262 (https://datatracker.ietf.org/doc/html/rfc3262)
SUBSCRIBE	Initiates a subscription for notification of events from a notifier.		RFC 6665 (https://datatracker.ietf.org/doc/html/rfc6665)
NOTIFY	Inform a subscriber of notifications of a new event.		RFC 6665 (https://datatracker.ietf.org/doc/html/rfc6665)
PUBLISH	Publish an event to a notification server.		RFC 3903 (https://datatracker.ietf.org/doc/html/rfc3903)
MESSAGE	Deliver a text message.	Used in instant messaging applications.	RFC 3428 (https://datatracker.ietf.org/doc/html/rfc3428)
INFO	Send mid-session information that does not modify the session state.	This method is often used for DTMF relay.	RFC 6086 (https://datatracker.ietf.org/doc/html/rfc6086)

OPTIONS	Query the capabilities of an endpoint.	It is often used for NAT keepalive purposes.	RFC 3261 (https://datatracker.ietf.org/doc/html/rfc3261)

Responses

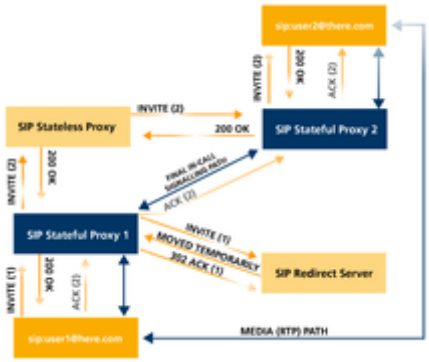
Responses are sent by the user agent server indicating the result of a received request. Several classes of responses are recognized, determined by the numerical range of result codes:[19]

- 1xx: Provisional responses to requests indicate the request was valid and is being processed.
- 2xx: Successful completion of the request. As a response to an INVITE, it indicates a call is established. The most common code is 200, which is an unqualified success report.
- 3xx: Call redirection is needed for completion of the request. The request must be completed with a new destination.
- 4xx: The request cannot be completed at the server for a variety of reasons, including bad request syntax (code 400).
- 5xx: The server failed to fulfill an apparently valid request, including server internal errors (code 500).
- 6xx: The request cannot be fulfilled at any server. It indicates a global failure, including call rejection by the destination.

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*.



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 choose 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. Message Session Relay Protocol (MSRP) 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 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

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 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 a similar marketing term preferred for when the service is used to simplify a telecom infrastructure by sharing the carrier access circuit for voice, data, and Internet traffic while removing the need for 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 (https://datatracker.ietf.org/doc/html/rfc3261) in full and a number of extension RFCs including RFC 6665 (https://datatracker.ietf.org/doc/html/rfc6665) (event notification) and RFC 3262 (https://datatracker.ietf.org/doc/html/rfc3262) (reliable provisional responses).

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

SIP-ISUP interworking

SIP-I, 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.[a] SIP-I was defined by the ITU-T, whereas SIP-T was defined by the IETF.[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 SIP communication be secured with Transport Layer Security (TLS). SIPS URIs take the form sips:user@example.com.

End-to-end encryption of SIP is only possible if there is a direct connection between communication endpoints. While a direct connection can be made via Peer-to-peer SIP or via a VPN between the endpoints, most SIP communication involves multiple hops, with the first hop being from a user agent to the user agent's ITSP. For the multiple-hop case, SIPS will only secure the first hop; the remaining hops will normally not be secured with TLS and the SIP communication will be insecure. In contrast, the HTTPS protocol provides end-to-end security as it is done with a direct connection and does not involve the notion of hops.

The media streams (audio and video), which are separate connections from the SIPS signaling stream, may be encrypted using SRTP. The key exchange for SRTP is performed with SDES (RFC 4568 (https://datatracker.ietf.org/doc/html/rfc4568)), or with ZRTP (RFC 6189 (https://datatracker.ietf.org/doc/html/rfc6189)). When SDES is used, the keys will be transmitted via insecure SIP unless SIPS is used. One may also add a MIKEY (RFC 3830 (https://datatracker.ietf.org/doc/html/rfc3830)) exchange to SIP to determine session keys for use with SRTP.

See also

- Computer telephony integration (CTI)
- Computer-supported telecommunications applications (CSTA)
- H.323 protocols H.225.0 and H.245
- IP Multimedia Subsystem (IMS)
- Media Gateway Control Protocol (MGCP)
- Mobile VoIP
- MSCML (Media Server Control Markup Language)
- Network convergence
- Rendezvous protocol
- RTP payload formats
- SIGTRAN (Signaling Transport)
- SIP extensions for the IP Multimedia Subsystem
- SIP provider
- Skinny Client Control Protocol (SCCP)
- T.38
- XIMSS (XML Interface to Messaging, Scheduling, and Signaling)

Notes

a. ISUP detail 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.

References

1. "What is SIP?" (http://www.networkworld.com/article/2332980/lan-wan/what-is-sip-.html). Network World. May 11, 2004.

2. Johnston, Alan B. (2004). *SIP: Understanding the Session Initiation Protocol* (Second ed.). Artech House. ISBN 978-1-58053-168-9.
3. "SIP core working group charter" (http://www.ietf.org/dyn/wg/charter/sipcore-charter.html). Internet Engineering Task Force. 2010-12-07. Retrieved 2011-01-11.
4. "Search Internet-Drafts and RFCs" (https://datatracker.ietf.org/doc/search/?name=SIP&rfcs=on&sort=date). Internet Engineering Task Force.
5. *SIP: Session Initiation Protocol* (https://tools.ietf.org/html/rfc3261). 2002. doi:10.17487/RFC3261 (https://doi.org/10.17487%2FRFC3261). RFC 3261 (https://tools.ietf.org/html/rfc3261).
6. Margaret Rouse. "Session Initiation Protocol (SIP)" (http://searchunifiedcommunications.techtarget.com/definition/Session-Initiation-Protocol). TechTarget.
7. Coll, Eric (2016). *Telecom 101*. Teracom Training Institute. pp. 77–79. ISBN 9781894887038.
8. *Uniform Resource Identifiers (URI): Generic Syntax* (https://tools.ietf.org/html/rfc3986). 2005. doi:10.17487/RFC3986 (https://doi.org/10.17487%2FRFC3986). RFC 3986 (https://tools.ietf.org/html/rfc3986).
9. Miikka Poikselkä et al. 2004.
10. Brian Reid & Steve Goodman 2015.
11. "SIP: Session Initiation Protocol" (https://www.ietf.org/rfc/rfc3261). *IETF*.
12. *The Stream Control Transmission Protocol (SCTP) as a Transport for the Session Initiation Protocol (SIP)* (https://tools.ietf.org/html/rfc4168). 2005. doi:10.17487/RFC4168 (https://doi.org/10.17487%2FRFC4168). RFC 4168 (https://tools.ietf.org/html/rfc4168).
13. 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". *Transactions on Emerging Telecommunications Technologies*. **27** (6): 857–873. doi:10.1002/ett.3038 (https://doi.org/10.1002%2Fett.3038). ISSN 2161-3915 (https://www.worldcat.org/issn/2161-3915).
14. Montazerolghaem, A.; Moghaddam, M. H. Y.; Leon-Garcia, A. (March 2018). "OpenSIP: Toward Software-Defined SIP Networking". *IEEE Transactions on Network and Service Management*. **15** (1): 184–199. arXiv:1709.01320 (https://arxiv.org/abs/1709.01320). doi:10.1109/TNSM.2017.2741258 (https://doi.org/10.1109%2FTNSM.2017.2741258). ISSN 1932-4537 (https://www.worldcat.org/issn/1932-4537). S2CID 3873601 (https://api.semanticscholar.org/CorpusID:3873601).
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" (https://web.archive.org/web/20110716170218/http://www.voipuser.org/forum_topic_14998.html). VoIP User. Archived from the original (http://www.voipuser.org/forum_topic_14998.html) on 2011-07-16.
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. "SIPit Wiki" (http://www.sipit.net/). Retrieved 2017-10-07.
22. *Experiences of Using TTCN-3 for Testing SIP and also OSP* (https://web.archive.org/web/20140330061038/http://portal.etsi.org/ptcc/downloads/TTCN3SIPOSP.pdf) (PDF), archived from the original (http://portal.etsi.org/ptcc/downloads/TTCN3SIPOSP.pdf) (PDF) on March 30, 2014

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. *SIP-T Context and Architectures* (https://tools.ietf.org/html/rfc3372). September 2002. doi:10.17487/RFC3372 (https://doi.org/10.17487%2FRFC3372). RFC 3372 (https://tools.ietf.org/html/rfc3372).
29. "Why SIP-I? A Switching Core Protocol Recommendation" (https://web.archive.org/web/20120317091603/http://www.4gamericas.org/documents/3G_Americas_SIP-I_White_Paper_August_2007-FINAL.pdf) (PDF). Archived from the original (http://www.4gamericas.org/documents/3G_Americas_SIP-I_White_Paper_August_2007-FINAL.pdf) (PDF) on 2012-03-17.

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

- IANA: SIP Parameters (https://www.iana.org/assignments/sip-parameters)
- IANA: SIP Event Types Namespace (https://www.iana.org/assignments/sip-events/sip-events.xhtml)
