

EXHIBIT K

Network Working Group
Request for Comments: 2543
Category: Standards Track

M. Handley
ACIRI
H. Schulzrinne
Columbia U.
E. Schooler
Cal Tech
J. Rosenberg
Bell Labs
March 1999

SIP: Session Initiation Protocol

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

IESG Note

The IESG intends to charter, in the near future, one or more working groups to produce standards for "name lookup", where such names would include electronic mail addresses and telephone numbers, and the result of such a lookup would be a list of attributes and characteristics of the user or terminal associated with the name. Groups which are in need of a "name lookup" protocol should follow the development of these new working groups rather than using SIP for this function. In addition it is anticipated that SIP will migrate towards using such protocols, and SIP implementors are advised to monitor these efforts.

Abstract

The Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol for creating, modifying and terminating sessions with one or more participants. These sessions include Internet multimedia conferences, Internet telephone calls and multimedia distribution. Members in a session can communicate via multicast or via a mesh of unicast relations, or a combination of these.

SIP invitations used to create sessions carry session descriptions which allow participants to agree on a set of compatible media types. SIP supports user mobility by proxying and redirecting requests to the user's current location. Users can register their current location. SIP is not tied to any particular conference control protocol. SIP is designed to be independent of the lower-layer transport protocol and can be extended with additional capabilities.

Table of Contents

1	Introduction	7
1.1	Overview of SIP Functionality	7
1.2	Terminology	8
1.3	Definitions	9
1.4	Overview of SIP Operation	12
1.4.1	SIP Addressing	12
1.4.2	Locating a SIP Server	13
1.4.3	SIP Transaction	14
1.4.4	SIP Invitation	15
1.4.5	Locating a User	17
1.4.6	Changing an Existing Session	18
1.4.7	Registration Services	18
1.5	Protocol Properties	18
1.5.1	Minimal State	18
1.5.2	Lower-Layer-Protocol Neutral	18
1.5.3	Text-Based	20
2	SIP Uniform Resource Locators	20
3	SIP Message Overview	24
4	Request	26
4.1	Request-Line	26
4.2	Methods	27
4.2.1	INVITE	28
4.2.2	ACK	29
4.2.3	OPTIONS	29
4.2.4	BYE	30
4.2.5	CANCEL	30
4.2.6	REGISTER	31
4.3	Request-URI	34
4.3.1	SIP Version	35
4.4	Option Tags	35
4.4.1	Registering New Option Tags with IANA	35
5	Response	36
5.1	Status-Line	36
5.1.1	Status Codes and Reason Phrases	37
6	Header Field Definitions	39
6.1	General Header Fields	41
6.2	Entity Header Fields	42
6.3	Request Header Fields	43

6.4	Response Header Fields	43
6.5	End-to-end and Hop-by-hop Headers	43
6.6	Header Field Format	43
6.7	Accept	44
6.8	Accept-Encoding	44
6.9	Accept-Language	45
6.10	Allow	45
6.11	Authorization	45
6.12	Call-ID	46
6.13	Contact	47
6.14	Content-Encoding	50
6.15	Content-Length	51
6.16	Content-Type	51
6.17	CSeq	52
6.18	Date	53
6.19	Encryption	54
6.20	Expires	55
6.21	From	56
6.22	Hide	57
6.23	Max-Forwards	59
6.24	Organization	59
6.25	Priority	60
6.26	Proxy-Authenticate	60
6.27	Proxy-Authorization	61
6.28	Proxy-Require	61
6.29	Record-Route	62
6.30	Require	63
6.31	Response-Key	63
6.32	Retry-After	64
6.33	Route	65
6.34	Server	65
6.35	Subject	65
6.36	Timestamp	66
6.37	To	66
6.38	Unsupported	68
6.39	User-Agent	68
6.40	Via	68
6.40.1	Requests	68
6.40.2	Receiver-tagged Via Header Fields	69
6.40.3	Responses	70
6.40.4	User Agent and Redirect Servers	70
6.40.5	Syntax	71
6.41	Warning	72
6.42	WWW-Authenticate	74
7	Status Code Definitions	75
7.1	Informational 1xx	75
7.1.1	100 Trying	75
7.1.2	180 Ringing	75

7.1.3	181 Call Is Being Forwarded	75
7.1.4	182 Queued	76
7.2	Successful 2xx	76
7.2.1	200 OK	76
7.3	Redirection 3xx	76
7.3.1	300 Multiple Choices	77
7.3.2	301 Moved Permanently	77
7.3.3	302 Moved Temporarily	77
7.3.4	305 Use Proxy	77
7.3.5	380 Alternative Service	78
7.4	Request Failure 4xx	78
7.4.1	400 Bad Request	78
7.4.2	401 Unauthorized	78
7.4.3	402 Payment Required	78
7.4.4	403 Forbidden	78
7.4.5	404 Not Found	78
7.4.6	405 Method Not Allowed	78
7.4.7	406 Not Acceptable	79
7.4.8	407 Proxy Authentication Required	79
7.4.9	408 Request Timeout	79
7.4.10	409 Conflict	79
7.4.11	410 Gone	79
7.4.12	411 Length Required	79
7.4.13	413 Request Entity Too Large	80
7.4.14	414 Request-URI Too Long	80
7.4.15	415 Unsupported Media Type	80
7.4.16	420 Bad Extension	80
7.4.17	480 Temporarily Unavailable	80
7.4.18	481 Call Leg/Transaction Does Not Exist	81
7.4.19	482 Loop Detected	81
7.4.20	483 Too Many Hops	81
7.4.21	484 Address Incomplete	81
7.4.22	485 Ambiguous	81
7.4.23	486 Busy Here	82
7.5	Server Failure 5xx	82
7.5.1	500 Server Internal Error	82
7.5.2	501 Not Implemented	82
7.5.3	502 Bad Gateway	82
7.5.4	503 Service Unavailable	83
7.5.5	504 Gateway Time-out	83
7.5.6	505 Version Not Supported	83
7.6	Global Failures 6xx	83
7.6.1	600 Busy Everywhere	83
7.6.2	603 Decline	84
7.6.3	604 Does Not Exist Anywhere	84
7.6.4	606 Not Acceptable	84
8	SIP Message Body	84
8.1	Body Inclusion	84

8.2	Message Body Type	85
8.3	Message Body Length	85
9	Compact Form	85
10	Behavior of SIP Clients and Servers	86
10.1	General Remarks	86
10.1.1	Requests	86
10.1.2	Responses	87
10.2	Source Addresses, Destination Addresses and Connections	88
10.2.1	Unicast UDP	88
10.2.2	Multicast UDP	88
10.3	TCP	89
10.4	Reliability for BYE, CANCEL, OPTIONS, REGISTER Requests	90
10.4.1	UDP	90
10.4.2	TCP	91
10.5	Reliability for INVITE Requests	91
10.5.1	UDP	92
10.5.2	TCP	95
10.6	Reliability for ACK Requests	95
10.7	ICMP Handling	95
11	Behavior of SIP User Agents	95
11.1	Caller Issues Initial INVITE Request	96
11.2	Callee Issues Response	96
11.3	Caller Receives Response to Initial Request	96
11.4	Caller or Callee Generate Subsequent Requests	97
11.5	Receiving Subsequent Requests	97
12	Behavior of SIP Proxy and Redirect Servers	97
12.1	Redirect Server	97
12.2	User Agent Server	98
12.3	Proxy Server	98
12.3.1	Proxying Requests	98
12.3.2	Proxying Responses	99
12.3.3	Stateless Proxy: Proxying Responses	99
12.3.4	Stateful Proxy: Receiving Requests	99
12.3.5	Stateful Proxy: Receiving ACKs	99
12.3.6	Stateful Proxy: Receiving Responses	100
12.3.7	Stateless, Non-Forking Proxy	100
12.4	Forking Proxy	100
13	Security Considerations	104
13.1	Confidentiality and Privacy: Encryption	104
13.1.1	End-to-End Encryption	104
13.1.2	Privacy of SIP Responses	107
13.1.3	Encryption by Proxies	108
13.1.4	Hop-by-Hop Encryption	108
13.1.5	Via field encryption	108
13.2	Message Integrity and Access Control: Authentication	109

redirect server maintains transaction state for the whole SIP transaction. It is up to the client to detect forwarding loops between redirect servers.

12.2 User Agent Server

User agent servers behave similarly to redirect servers, except that they also accept requests and can return a response of class 2xx.

12.3 Proxy Server

This section outlines processing rules for proxy servers. A proxy server can either be stateful or stateless. When stateful, a proxy remembers the incoming request which generated outgoing requests, and the outgoing requests. A stateless proxy forgets all information once an outgoing request is generated. A forking proxy SHOULD be stateful. Proxies that accept TCP connections MUST be stateful.

Otherwise, if the proxy were to lose a request, the TCP client would never retransmit it.

A stateful proxy SHOULD NOT become stateless until after it sends a definitive response upstream, and at least 32 seconds after it received a definitive response.

A stateful proxy acts as a virtual UAS/UAC. It implements the server state machine when receiving requests, and the client state machine for generating outgoing requests, with the exception of receiving a 2xx response to an INVITE. Instead of generating an ACK, the 2xx response is always forwarded upstream towards the caller. Furthermore, ACK's for 200 responses to INVITE's are always proxied downstream towards the UAS, as they would be for a stateless proxy.

A stateless proxy does not act as a virtual UAS/UAC (as this would require state). Rather, a stateless proxy forwards every request it receives downstream, and every response it receives upstream.

12.3.1 Proxying Requests

To prevent loops, a server MUST check if its own address is already contained in the Via header field of the incoming request.

The To, From, Call-ID, and Contact tags are copied exactly from the original request. The proxy SHOULD change the Request-URI to indicate the server where it intends to send the request.