SIP: Session Initiation Protocol

From HTTP and Session Invitation to Setup and Control for Packet-based Multimedia Conferencing

Conference Establishment & Control

1. Create
   - Announcement Protocol
   - Announcement
   - Netnews
   - WWW
   - E-Mail
   - Invitation Protocol
   - Streaming Protocol

2a. Announcement
2b. Invitation
2c. Request

3. Join

3. Join

4. Media streams
History of Mbone conference initiation

**Session Invitation Protocol** *(Handley/Schooler)*
- Participant location
- Conference invitation
- Capability negotiation during setup

**Simple Conference Invitation Protocol** *(Schulzrinne)*
- Participant location
- Conference invitation
- Capability negotiation during setup
- Changing conference parameters
- Terminate/leave conference

1996 **Session Initiation Protocol**

Session Initiation Protocol (SIP)
First draft in December 1996
- Joint effort to merge SIP and SCIP
- IETF WG MMUSIC *(Multiparty Multimedia Session Control)*

Application-layer call signaling protocol:
- Creation, modification, termination of teleconferences
- Negotiation of used media configuration
- Re-negotiation during session
- User location → personal mobility
- Security
- Supplementary services

RFC 3261
- June 2002
- obsoletes RFC 2543
SIP and Conferencing over Time…

- **Origin:** MMUSIC: *Multiparty Multimedia Session Control*
- **From Invitation…** to initiation, modification, and termination
- **From Multiparty…** to point-to-point-focused
- **From Multimedia…** to voice-centric

**Past years:** Multiparty & multimedia rediscovered

**But:** Don’t believe in multicast (anymore)!

---

“Weight” of SIP Base Spec
IETF SIP-related Working Groups (1)

- MMUSIC WG
  - Sep 99
- RFC 2543 (Feb 1999)
- SIP WG
  - Mar 01
- SIPPING WG
  - Oct 03
- SIMPLE WG
  - Dec 00
- XCON WG
  - Oct 03
- SIPCORE
  - Dispatch

IETF SIP-related Working Groups (2)

- MMUSIC WG
  - SDP extensions
  - NAT Traversal for SIP
- SIPWG
  - SIP core spec maintenance
  - SIP protocol extensions
- SIPPING WG
  - Requirements for SIP
  - Specific SIP application services
- SIMPLE WG
  - SIP for Presence and Instant Messaging
- XCON WG
  - Centralized Conferencing

© 2010 Jörg Ott
IETF SIP-related Working Groups (3)

- atoca  Authority-to-Citizen Alert
- bliss  Basic Level of Interoperability for SIP Services
- cuss  Call Control UUI Service for SIP
- dispatch  Dispatch
- drinks  Data for Reachability of Inter/tra-NetworK SIP
- ecrit  Emergency Context Resolution with Internet Technologies
- geopriv  Geographic Location/Privacy
- martini  Multiple AoR reachability Information Indication
- p2psip  Peer-to-Peer Session Initiation Protocol
- salud  Sip Alerting for User Devices
- simple  SIP for Instant Messaging and Presence Leveraging Extensions
- sipcfg  SIP Common Log Format
- sipcore  Session Initiation Protocol Core
- siprec  SIP Recording
- soc  SIP Overload Control
- splices  looSely-couPLed sip deviCES
- xcon  Centralized Conferencing
- xmpp  Extensible Messaging and Presence Protocol
- speermint  Session PEERing for Multimedia INTerconnect
- enum  Telephone Number Mapping
- avt  Audio/Video Transport
- codec  Internet Wideband Audio Codec
- speechsc  Speech Services Control
- mediacr  Media Server Control
- mmusic  Multiparty Multimedia Session Control

“Productivity” (1): Internet Draft Pages

(rough estimate with errors!)
RFCs related to SIP (1)

Original base spec
- RFC 3261: SIP: Session Initiation Protocol
- RFC 3263: Locating SIP Servers
- RFC 3264: An Offer/Answer Model with SDP

Extended Features
- RFC 2976: The SIP INFO Method
- RFC 3262: Reliability of Provisional Responses in SIP
- RFC 3265: SIP-specific Event Notification
- RFC 3311: SIP UPDATE Method
- RFC 3312, RFC 4032: Integration of Resource Management and SIP
- RFC 3326: Reason Header
- RFC 3327: Registering Non-Adjacent Contacts
- RFC 3428: Instant Messaging
- RFC 3487: Requirements for Resource Priority
- RFC 3515: SIP REFER Method
- RFC 3581: Symmetric Message Routing
- RFC 3680: SIP event package for registrations
- RFC 3725: Third-party Call Control (3PCC)
- RFC 3840, 3841: Callee capabilities and caller preferences
- RFC 3842: Message waiting indication / message summary
- RFC 3857, 3958: Watcher Information event package + XML format
- RFC 3891: Replaces: header
- RFC 3892: Referred-By: header
- RFC 3903: Event state publication (SIP PUBLISH method)
- RFC 3911: Join: header
- RFC 4028: Session timers
- RFC 4168: SCTP as transport protocol

RFCs related to SIP (2)

Extended features (continued)
- RFC 4244: Request history
- RFC 4320: Addressing issues with non-INVITE transactions
- RFC 4412: Communications resource priority for SIP
- RFC 4483: Content indication in SIP
- RFC 4488: Suppressing implicit subscriptions of REFER
- RFC 4508: Conveying feature tags with REFER
- RFC 4235: INVITE-initiated dialog event package
- RFC 4245: Requirements for SIP conferencing
- RFC 4353: SIP conferencing framework
- RFC 4376: Floor control requirements
- RFC 4411: SIP Reason header for preemption
- RFC 4453: Requirements for consent-based communications
- RFC 4475: SIP torture test messages
- RFC 4479: A data model for presence
- RFC 4480: RPID: rich presence
- RFC 4481: Extensions for timed presence
- RFC 4482: CPID: Contact information in presence
- RFC 4575: SIP conference event package
- RFC 4579: SIP call control: conferencing for user agents
- RFC 4596: Caller preferences extensions
- RFC 4597: Conferencing scenarios
- RFC 4660: Functional description of event filtering
- RFC 4661: XML for event filtering
- RFC 4662: Event notifications for resource lists
- RFC 4730: Key Press Stimulus Event Package (KPML)
- RFC 4916: Connected identity
RFCs related to SIP (3)

- Extended features (continued)
  - RFC 4825: XCAP
  - RFC 4826: XCAP Processing Rules for Resource Lists
  - RFC 4827: XCAP For Manipulating Presence Contents
  - RFC 4975: MSRP
  - RFC 4976: MSRP Relays

- Security
  - RFC 3323: A Privacy Mechanism for SIP
  - RFC 3325: Private Extension for Asserted Identity in Trusted Networks
  - RFC 3329: Security-Mechanism Agreement for SIP
  - RFC 3603: Proxy-to-Proxy Extensions
  - RFC 3702: AAA requirements for SIP
  - RFC 3853: S/MIME AES
  - RFC 3893: Authenticated Identity Body
  - RFC 4189: Requirements for end-to-middle security
  - RFC 4474: Enhancements for authenticated identity management
  - RFC 4484: Trait-based authentication requirements
  - RFC 4538: Request authorization through dialog identification

RFCs related to SIP (4)

- Others
  - RFC 3665, 3666: SIP Call Flows
  - RFC 3361: DHCP Option for SIP Servers
  - RFC 3608: Service Route Discovery
  - RFC 3398, 3578: ISUP and SIP Mapping
  - RFC 3420: Internet Media Type message/sipfrag
  - RFC 3427: SIP Change Process
  - RFC 3455: Header Extensions for 3GPP
  - RFC 3485, 3486: SIP header compression
  - RFC 3764, 3824: Using ENUM with SIP
  - RFC 3965: Early Session disposition type (early-session, session)
  - RFC 3960: Early Media and Ringing Tone Generation
  - RFC 3968, 3969: IANA SIP header field and URI registry
  - RFC 3976: SIP – IN Interworking
  - RFC 4117: 3rd party call control invocation of transcoding services
  - RFC 4123: SIP – H.323 Interworking requirements
  - RFC 4485: Guidelines for authors of SIP extensions
  - RFC 4497: SIP – QSIG interworking
  - RFC 4569: IANA media feature tag registration
  - RFC 4780: SIP MIB

- Related: RTP, SDP, Security basics, 3GPP requirements and extensions

A Hitchhikers Guide to the Session Initiation Protocol (SIP)

RFC 5411

Many more RFCs > 5000 + many Internet Drafts
SIP is not …

- Tied to any specific media
- Intended for conference control by itself
  - No floor control
  - No participant lists
  - No policies, voting, …
- Designed for distribution of multimedia data
  - Some extensions allow for carrying images, audio files, etc.
- A generic transport protocol!
- Another RPC mechanism
  - SIP has no inherent support for distributed state information
- Something to put into every device on the planet
  - No general IP infrastructure part (yet?)
  - Nevertheless: Application layer routing gets more and more important
- (but proposals for “misuse” show up again and again)

Base Terminology

- User Agent Client (UAC):
  - Endpoint, initiates SIP transactions
- User Agent Server (UAS):
  - Handles incoming SIP requests
- Redirect server:
  - Retrieves addresses for callee and returns them to caller
- Proxy (server):
  - Autonomously processes and routes requests → forward incoming messages (limited modifications only)
- Registrar:
  - Stores explicitly registered user addresses
- Location Service:
  - Provides information about a target user’s location
- Back-to-Back User Agent (B2BUA)
  - Keeps call state; more powerful intervention than proxy
Protocol Characteristics

- Transaction oriented
  - Request–response sequences
- Independent from lower layer transport protocol
  - Works with a number of unreliable and reliable transports
    - UDP, TCP, SCTP
    - Secure transport: TLS over TCP, IPSec
  - Retransmissions to achieve reliability over UDP
  - Optionally use IP multicast → anycast service
- Independent of the session to be (re-)configured
- Re-use syntax of HTTP 1.1
  → Text-based protocol (UTF-8 encoding)
- Enable servers maintaining minimal state info
  - Stateless proxies
  - Transaction-stateful proxies
  - Dialog (call) state in endpoints (optional for proxies)
Functional Layers

- **Transaction User**: session creation, application-specific processing
- **Transaction**: Transaction handling, request retransmission
- **Transport**: send/receive SIP messages
- **Syntax / Encoding**: Message parsing
- **UDP**
- **TCP**
- **SCTP**
- **TLS**

SIP Transactions

- RPC-like approach:
  - Initial request
  - Wait for final response
- Provisional responses:
  - Additional status information
  - May be unreliable
- Unique identifier (transaction id) (originator, recipient, unique token, sequence number, ...)
- Independent completion

For exclusive use with TKK Netlab course S-38.3150 Networked Multimedia Protocols and Services
Dialogs

- Signaling vs. media session
- Distributed state between endpoints
  - State change if transaction succeed
  - No change on error
- Unique dialog identifier

Dialog Example: Media Sessions

Special case: three-way handshake for INVITE transaction
### SIP Message Syntax: Request

**Start line**

```
INVITE sip:user@example.com SIP/2.0
```

**Message headers**

```
To: John Doe <sip:user@example.com>
From: sip:jo@tzi.uni-bremen.de;tag=4711
Subject: Congratulations!
Content-Length: 117
Content-Type: application/sdp
Call-ID: 2342344233@134.102.218.1
CSeq: 49581 INVITE
Contact: sip:jo@134.102.224.152:5083
;transport=udp
Via: SIP/2.0/UDP 134.102.218.1;
branch=z9hG4bK776asdhds
```

**Message body (SDP content)**

```
v=0
o=jo 75638353 98543585 IN IP4 134.102.218.1
s=SIP call
t=0 0
c=IN IP4 134.102.224.152
m=audio 47654 RTP/AVP 0 1 4
```

### SIP Message Syntax: Response

**Start line**

```
SIP/2.0 200 OK
```

**Message headers**

```
To: John Doe <sip:user@example.com>;tag=428
From: sip:jo@tzi.uni-bremen.de;tag=4711
Subject: Congratulations!
Content-Length: 121
Content-Type: application/sdp
Call-ID: 2342344233@134.102.218.1
CSeq: 49581 INVITE
Contact: sip:jdoe@somehost.domain
Via: SIP/2.0/UDP 134.102.218.1;
branch=z9hG4bK776asdhds
```

**Message body (SDP content)**

```
v=0
o=jdoe 28342 98543601 IN IP4 134.102.20.22
s=SIP call
t=0 0
c=IN IP4 134.102.20.38
m=audio 61002 RTP/AVP 0 4
```
SIP URI Addressing Scheme

- Follows basic URI syntax per RFC 2396
- Separating names (permanent) and addresses (temporary)
  - Basic mobility support
- Two roles reflected in SIP
  - Naming a user; typically sip:user@domain
  - Contact address of a user; typically contains host name or IP address, port, transport protocol, ...
- URIs may carry additional parameters

\[
\text{sip: } [ \text{user} [ ': ' \text{passwd} ] @ ] \text{host} [ ': ' \text{port} ] \text{params} [ '?' \text{headers} ]
\]

\[
\text{params ::= ( ';' \text{name} [ '=' \text{value} ] )*}
\]

\[
\text{headers ::= \text{field} = \text{value}? \ [ '&\' \text{headers} ]}
\]

- URIs may also identify services

SIP URI Addressing Examples

- Registration domain or IP address
  - sip:tzi.org
  - sip:192.168.42.1

- SIP URI to call (Address of Record)
  - sip:john@example.com
  - sip:john@host1.example.com
  - sip:john@192.168.42.9:9950

- SIP Contact Address (actual user location)
  - sip:voicemail@service.com
  - sip:conf-1234@confserv.com
  - sip:user34@anonymizer.org

- Service identifier; semantics opaque to the user

Use URI scheme ‘sips’ to request secure communications.
SIP URI Addressing Examples (2)

URI parameters may carry detailed information on specific URI components:

- `sip:john@Example.COM;maddr=10.0.0.1`
- `sip:+1555123456@tel-gw.myitsp.com;user=phone`

Nested URI Encoding (e.g. for Service Description)

Encapsulation

```
sip:sip:3A%40%20%2E%20%3A%20%2D%20%3B%3Bmaddr=134.102.3.99@example.com
```

Need to encode reserved characters

Service indication example

```
sip:voicemail.replay=abl%20817m@media-engine;msgid=78
```

Additional header fields (line breaks inserted for readability)

```
sip:sales@warehouse.com;method=INVITE \
?Subject=gw%20c26111%20Call-ID=c239xa2-as921b%40warehouse.com \
sip:jo@example.com?Replaces=abcd@example.com%3B \
from-tag%3D26%3Bto-tag%3D234bl%20Accept-Contact= %3Csip \
%3A%40%20%2E%20%2D%20%3E%3Bonly%3Dtrue
```

Separator characters
URIs in Header Fields

URI-parameters vs. header parameters

Contact: sip:bob@p2.example.com:55060
;methods="NOTIFY"
;expires=3600

→ angle brackets:

Contact: < sip:bob@p2.example.com:55060
;methods="NOTIFY" >
;expires=3600

Required if
- URI contains comma, question mark or semicolon
- The header field contains a display name

Further Common URI Schemes

Telephony (RFC 2806)

tel:+1-555-12345678
tel:7595;phone-context=+49421218

ITU-T H.323 Protocol

h323:user@example.com

Instant Messaging

im:user@example.com

Presence

pres:user@example.com