The Session Initiation Protocol (SIP)

Henning Schulzrinne
Dept. of Computer Science
Columbia University
New York, New York
(sip:)schulzrinne@cs.columbia.edu
Overview

- protocol architecture
- typical component architectures
- protocol operation
- reliability
- features
- security
Introduction

- core protocol for establishing *sessions* in the Internet
- transports session description information from initiator (caller) to callees
- allows to change parameters in mid-session
- terminate session
Protocol architecture

Languages/ APIs
- JAIN
- CPL
- voiceXML
- Parlay
- servlets
- sip-cgi

Directory/Discovery
- DNS/enum
- LDAP
- TRIP
- SLP

Signaling
- peer-to-peer
- SIP
- SDP
- MGCP
- H.248
- RTSP
- master-slave

QoS
- DiffServ
- IntServ

Transport
- RTP
- SCTP
- TLS

May 2001
SIP applications

- setting up voice-over-IP calls
- setting up multimedia conferences
- event notification (subscribe/notify) ➞ IM and presence
- text and general messaging
- signaling transport
Personal mobility

SIP uses email-style addresses to identify users

- alice17@yahoo.com
- alice@columbia.edu
- 7000@columbia.edu
- Alice.Cary@columbia.edu
- tel:12015551234
- alice@host.columbia.edu

(also used by bob@columbia.edu)

- alice@columbia.edu
- tel:12128541111
- tel:12015551234

yahoo.com

columbia.edu
SIP addressing

- typically, same as user’s email address:
  alice@example.com
  12125551212@gateways-r-us.com

- written as URL, e.g., sip:alice@example.com

- also can use tel URLs for telephone numbers, e.g., tel:+12125551212 or fax:+358.555.1234567
Building blocks

- SIP user agent: IP phone, PC, conference bridge
- SIP redirect server: returns new location for requests
- SIP stateless proxy: routes call requests
- SIP (forking) proxy: routes call requests
- SIP registrar: maintains mappings from names to addresses
Back-to-back UA (B2BUA)

- two (or more) user agents, where incoming calls trigger outgoing calls to somebody else
- also, “third-party call control” (later)
- useful for services and anonymity

```
SIP UA1 (UAS)   SIP UA2 (UAC)
```

```
INVITE b2b   INVITE callee
```

```
200 OK       200 OK
```

May 2001
Maintaining state in SIP entities

Stateless: each request and response handled independently

(Transaction) stateful: remember a whole request/response transaction

Call stateful: remember a call from beginning to end
# SIP building block properties

<table>
<thead>
<tr>
<th></th>
<th>media</th>
<th>stateless</th>
<th>stateful</th>
<th>call state</th>
</tr>
</thead>
<tbody>
<tr>
<td>UA (UAC, UAS)</td>
<td>yes</td>
<td>no</td>
<td>unlikely</td>
<td>common</td>
</tr>
<tr>
<td>proxy</td>
<td>no</td>
<td>yes</td>
<td>common</td>
<td>possible (firewall)</td>
</tr>
<tr>
<td>redirect registrar</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>N/A</td>
</tr>
</tbody>
</table>
SIP architecture: peer-to-peer
SIP architecture: outbound proxy

wonderland.com
alice@ph7.wonderland.com

outbound proxy

Internet

macrosoft.com
bob@p42.macrosoft.com

REGISTER sip:macrosoft.com SIP/2.0
To: sip:bob@macrosoft.com
From: sip:bob@macrosoft.com
Contact: sip:bob@p42.macrosoft.com

INVITE sip:bob@macrosoft.com SIP/2.0
 INVITE sip:bob@p42.macrosoft.com SIP/2.0

INVITE sip:bob@macrosoft.com SIP/2.0

SIP architecture: VoIP to PSTN
SIP architecture: PSTN to VoIP

INVITE sip:alice@wonderland.com

 enum database
 DNS
 4.3.2.1.5.5.2.1.2.1.e164.arpa
 enum
 sip:alice@wonderland.com

May 2001
SIP operation in proxy mode

1. INVITE henning@columbia.edu
2. INVITE hgs@play
3. 200 OK
4. 200 OK
5. play
6. tune
7. 200 OK
8. ACK hgs@play
9. media stream
SIP operation in redirect mode

(302: redirection for single call; 301 permanently)
Locating users: registrars and location servers

REGISTER
alice@example.com
Contact:
alice@pc17

INVITE
alice@example.com

INVITE alice@pc17.example.com

SQL, LDAP, Corba, proprietary, ...

registrar

A@
B@
C@

proxy

location server
Basic user location mechanism

1. host(SIP URL) → host name of proxy
2. DNS: host name of proxy → SIP server(s)
3. if SIP UAS: alert user; done
4. if SIP proxy/redirect server: map URL$_n$ → URL$_{n+1}$, using any information in request
5. go to step 1

One minor exception...
Basic SIP “routing” mechanisms

- will fill in details later
- route using request URIs
- all but first request in call typically bypass proxies and go direct UAC – UAS
- however, can use “record-routing” to force certain proxies to be visited all the time
- responses always traverse the same route as requests
Outbound proxies

- normally, proxy serves one or more domains
- outbound proxies are used for *all* outbound requests from within a domain
- typically, for managing corporate firewalls and policy enforcement
- may also provide dial plans or route tel/fax URLs
- other uses: lawyer client billing, ...
Locating users: DNS SRV

- **email**: DNS MX record allows mapping of domain to mail host, e.g.

  ```
  host -t mx yahoo.com
  yahoo.com          MX     1  mx2.mail.yahoo.com
  yahoo.com          MX     1  mx3.mail.yahoo.com
  yahoo.com          MX     1  mx1.mail.yahoo.com
  yahoo.com          MX     9  mta-v1.mail.yahoo.com
  ```

- **SIP**: use a newer record for general-purpose mapping, SRV (RFC 2782)

  ```
  _sip._tcp        SRV  0  0  5060  sip-server.cs.columbia.edu.
  SRV  1  0  5060  backup.ip-provider.net.
  
  _sip._udp        SRV  0  0  5060  sip-server.cs.columbia.edu.
  SRV  1  0  5060  backup.ip-provider.net.
  ```

- allows priority (for back-up) and weight (for load balancing)

May 2001
Using DNS SRV for scalable load-balancing

May 2001
### Differences to classical signaling

<table>
<thead>
<tr>
<th>name</th>
<th>examples</th>
<th>network</th>
<th>“channel”</th>
</tr>
</thead>
<tbody>
<tr>
<td>in-band</td>
<td>E&amp;M, DTMF</td>
<td>same</td>
<td>same</td>
</tr>
<tr>
<td>out-of-band</td>
<td>ISUP, Q.931</td>
<td>different</td>
<td>different</td>
</tr>
<tr>
<td>IP</td>
<td>SIP</td>
<td>typically same</td>
<td>different</td>
</tr>
</tbody>
</table>

IP signaling meets media only at end systems, while PSTN out-of-band intersects at every switch.
Aside: Alternative architecture: master-slave

- master-slave: MGC (media gateway controller) controls one or more gateways
- allows splitting of signaling and media functionality
- “please send audio from circuit 42 to 10.1.2.3”
- uses MGCP (implemented) or Megaco/H.248 (standardized, but just beginning to be implemented)
- gateway can be residential
- basis of PacketCable NCS (network control system) architecture
- service creation similar to digital PBX or switch
- end system has no semantic knowledge of what’s happening
- → can charge for caller id, call waiting
MGCP/SIP architecture
SIP requests and responses

- text, not binary, format
- look very similar to HTTP/1.1
- requests and responses are similar except for first line
- requests and responses can contain *message bodies*: typically session descriptions, but also ASCII or HTML
# SIP syntax

## Request

<table>
<thead>
<tr>
<th>Method URL</th>
<th>SIP/2.0</th>
<th>SIP/2.0 status reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Via:</td>
<td>SIP/2.0</td>
<td>protocol host:port</td>
</tr>
<tr>
<td>From:</td>
<td>user <a href="">sip:from_user@source</a></td>
<td></td>
</tr>
<tr>
<td>To:</td>
<td>user <a href="">sip:to_user@destination</a></td>
<td></td>
</tr>
<tr>
<td>Call-ID:</td>
<td>localid@host</td>
<td></td>
</tr>
<tr>
<td>CSeq:</td>
<td>seq# method</td>
<td></td>
</tr>
<tr>
<td>Content-Length:</td>
<td>length of body</td>
<td></td>
</tr>
<tr>
<td>Content-Type:</td>
<td>media type of body</td>
<td></td>
</tr>
<tr>
<td>Header:</td>
<td>parameter ;par1=value ;par2=&quot;value&quot; ;par3=&quot;value folded into next line&quot;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>blank line</th>
</tr>
</thead>
<tbody>
<tr>
<td>V=0</td>
</tr>
<tr>
<td>o= origin_user timestamp timestamp IN IP4 host</td>
</tr>
<tr>
<td>c=IN IP4 media destination address</td>
</tr>
<tr>
<td>t=0 0</td>
</tr>
<tr>
<td>m= media type port RTP/AVP payload types</td>
</tr>
</tbody>
</table>

## Response
SIP syntax

- field names and some tokens (e.g., media type) are case-insensitive
- everything else is case-sensitive
- white space doesn’t matter except in first line
- lines can be folded
- multi-valued header fields can be combined as a comma-list
# SIP methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVITE</td>
<td>initiate call</td>
</tr>
<tr>
<td>ACK</td>
<td>confirm final response</td>
</tr>
<tr>
<td>BYE</td>
<td>terminate (and transfer) call</td>
</tr>
<tr>
<td>CANCEL</td>
<td>cancel searches and “ringing”</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>features support by other side</td>
</tr>
<tr>
<td>REGISTER</td>
<td>register with location service</td>
</tr>
<tr>
<td>INFO</td>
<td>mid-call information (ISUP, DTMF)</td>
</tr>
<tr>
<td>COMET</td>
<td>precondition met</td>
</tr>
<tr>
<td>PRACK</td>
<td>provisional acknowledgement</td>
</tr>
<tr>
<td>SUBSCRIBE</td>
<td>subscribe to event</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>notify subscribers</td>
</tr>
</tbody>
</table>
Tagging To

- after forking and merging, hard to tell who responded
- UAS responds with random tag added to disambiguate
  
  To: "A. G. Bell" <sip:agb@bell-telephone.com>
  ;tag=a48s

- future requests are ignored if they contain the wrong tag
SIP call legs

- *call leg*: From, To, Call-ID
- requests from callee to caller reverse To and From
- caller and callee keep their own CSeq space
- either side can send more INVITEs or BYE
SIP responses

Informational

100 Trying
180 Ringing
181 Call forwarded
182 Queued
183 Session Progress

Success

200 OK

Redirection

300 Multiple Choices
301 Moved Perm.
302 Moved Temp.
380 Alternative Serv.

Request Failure

400 Bad Request
401 Unauthorized
403 Forbidden
404 Not Found
405 Bad Method
415 Unsupp. Content
420 Bad Extensions
486 Busy Here

Server Failure

500 Server Error
501 Not Implemented
503 Unavailable
504 Timeout

Global Failure

600 Busy Everwhere
603 Decline
604 Doesn’t Exist
606 Not Acceptable
SIP response routing

- requests are routed via URL
- response traces back request route \textit{without proxy server state}
- forward to host, port in next \texttt{Via}
- TCP: re-use connection if possible, create new one if needed
- UDP: may send responses to same port as requests

\texttt{Via: SIP/2.0/UDP server.domain.org:5060;received=128.1.2.3}
SIP response routing

alice@example.com  bob_doe@yahoo.com  bob@columbia.edu

---

May 2001
Forcing request paths

- usually, bypass proxies on subsequent requests
- some proxies want to stay in the path → call-stateful:
  - firewalls
  - anonymizer proxies
  - proxies controlling PSTN gateways
- use `Record-Route` and `Route`
SIP request forking

INVITE sales@macrosoft.com
INVITE carol@c.macrosoft.com
INVITE bob@b.macrosoft.com
200 OK
ACK
BYE carol@c.macrosoft.com
200 OK

macrosoft.com

a.wonderland.com

200 OK

CANCEL bob@c
INVITE carol@c

May 2001
SIP request forking

- branches tried in sequence or parallel (or some combination)
- recursion: may try new branches if branch returns 3xx
- return best final answer = lowest status code
- forward provisional responses
SIP transport issues

- SIP operates over any packet network, reliable or unreliable
- choices:
  - **UDP**: most common
    - low state overhead
    - small max. packet size
  - **TCP**: can combine multiple signaling flows over one link
    - use with SSL
    - connection setup overhead
    - HOL blocking for trunks
  - **SCTP**: new protocol
    - no HOL blocking
    - fallback address (but SRV provides this already)
    - connection setup overhead

May 2001
Transport reliability for all but INVITE

- used for BYE, OPTIONS, SUBSCRIBE, NOTIFY, ...
- 1xx sent by UAS or proxy only if no final answer expected within 200 ms
- if provisional response, re-transmit with $T_2$ (4) seconds
INVITE reliability

- INVITE is special – long time between request and final response
- 100 (by proxy) indicates request has been received
- proxy usually forwards 1xx from all branches
- only retransmit until 100
- ACK confirms receipt of final response
## Extending SIP

<table>
<thead>
<tr>
<th>extension</th>
<th>behavior</th>
<th>determine?</th>
</tr>
</thead>
<tbody>
<tr>
<td>new headers</td>
<td>ignored</td>
<td>–</td>
</tr>
<tr>
<td>new headers</td>
<td>mandatory</td>
<td>Supported</td>
</tr>
<tr>
<td>new method</td>
<td></td>
<td>OPTIONS</td>
</tr>
<tr>
<td>new body type</td>
<td></td>
<td>Accept</td>
</tr>
<tr>
<td>new status code</td>
<td>class-based</td>
<td></td>
</tr>
<tr>
<td>new URL type</td>
<td></td>
<td>?</td>
</tr>
</tbody>
</table>

May 2001
SIP extensions and feature negotiation

- if crucial, mark with “Require: feature”
- IANA-registered features are simple names, private features use reverse domain names
- indicate features supported in Supported:

  C→S: INVITE sip:watson@bell-telephone.com SIP/2.0
  Require: com.example.billing
  Supported: 100rel
  Payment: sheep_skins, conch_shells

  S→C: SIP/2.0 420 Bad Extension
  Unsupported: com.example.billing

  S→C: SIP/2.0 421 Extension Required
  Require: 183
**Invitation modes**

<table>
<thead>
<tr>
<th>signaling</th>
<th>media</th>
</tr>
</thead>
<tbody>
<tr>
<td>unicast</td>
<td>multicast</td>
</tr>
<tr>
<td>unicast</td>
<td>telephony</td>
</tr>
<tr>
<td>multicast</td>
<td>reach first</td>
</tr>
<tr>
<td></td>
<td>dept. conference</td>
</tr>
</tbody>
</table>

*SIP for all modes, SAP/SDP also for multicast/multicast*
SIP-based services

Call forwarding: basic INVITE behavior (proxy/redirect)

Call transfer: REFER method (see later)

DTMF carriage: carry as RTP payload (RFC 2833)

Calling card: B2BUA + voice server

Voice mail: UA with special URL(s) + possibly RTSP
# SIP security

<table>
<thead>
<tr>
<th>layer/mechanism</th>
<th>approach</th>
<th>characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>network layer</td>
<td>IPsec</td>
<td>adjacent nodes, all or nothing, hard to configure</td>
</tr>
<tr>
<td>transport layer</td>
<td>TLS</td>
<td>adjacent nodes, all or nothing</td>
</tr>
<tr>
<td>SIP INVITE</td>
<td>basic/digest</td>
<td>shared secrets with random parties</td>
</tr>
<tr>
<td>SIP REGISTER</td>
<td>basic/digest</td>
<td>securing headers?</td>
</tr>
<tr>
<td>SIP general</td>
<td>S/MIME</td>
<td>in progress</td>
</tr>
</tbody>
</table>

Basic (plaintext password) and digest (challenge-response) are very similar to HTTP security mechanisms.
For more information...

**SIP:**  http://www.cs.columbia.edu/sip

**SDP:**  http://www.cs.columbia.edu/~hgs/internet/sdp.html

**RTP:**  http://www.cs.columbia.edu/~hgs/rtp

**Papers:**  http://www.cs.columbia.edu/IRT

May 2001