Session Initiation Protocol

Jouni Soitinaho
Contents

• Protocol Development
• Protocol Basics
• Expandability
• Extension Examples
• Application areas
• Conclusions
Protocol Development

• Started in Multiparty Multimedia Session Control (MMUSIC) WG
  • RFC2543 to proposed state in March 1999
  • MMUSIC still develops SDP

• Continued in the SIP WG

• SIP WG will be split to two
  • SIP WG continues development of the core protocol/extensions
  • SIPPING WG concentrates on applications

• Cooperation with other wg's
  • PSTN and Internet Internetworking wg (pint) uses SIP
  • Distributed Call Signaling Group (DCS) gives input to SIP for distributed telephony services
  • IP telephony wg (iptel) develops CPL
  • SIMPLE wg (SIP for Instant Messaging and Presence Leveraging)

• 3GPP active on SIP
  • SIP is the call control protocol for IM subsystem of 3G (Rel 5)
Protocol Basics/ Features

- Locating user: determination of the end system to be used for communication;
- Determining user capabilities: determination of the media and media parameters to be used;
- Determining user availability: determination of the willingness of the called party to engage in communications;
- Setting up the call: "ringing", establishment of call parameters at both called and calling party;
- Controlling the call: including transfer and termination of calls.
Protocol Basics/ Technicalities

- Text-based (ISO 10646 in UTF-8 encoding), similar to HTTP
  - easy to learn, implement, debug and extend.
  - extra transmission overhead

- Recommended transport protocol is UDP
  - support for multicasting signalling
  - reliability has to be built in SIP elements

- Application level routing based on Request-URI
  - signalling path controlled by the protocol itself
  - routing has to be built in SIP proxies
  - forking proxies (shortens seek time, complicates implementation)

- Cooperates with other protocols (capability descriptions, transport protocols, conference control protocols, etc)
  - can be developed independently

- Designed for IP networking
  - uses standard elements

- Supports stateless, efficient and "forward" compatible proxies (re-INVITE carries state, ignore the body, ignore extension methods)
Protocol Basics/ SIP is not

? Since SIP is independent of the session:
  ? it's not a media transport protocol
  ? it's not a conference control protocol
  ? it's not a resource allocation protocol

? Since SIP is mainly used over UDP
  ? it's not for carrying large packets (except REGISTER/TCP)
  ? it's not a replacement for HTTP

? It's not a PSTN signalling replacement or superset of ISUP

? Since it's session initiation protocol
  ? it's not for sessionless communication
Protocol Basics/ Network Elements

Domain A
- DNS
- Outbound Proxy
- Firewall/NAT
- UAC

Domain B
- Location Server
- Proxy/Registrar
- UAS

Media flow
SIP protocol
Non-SIP protocol
Protocol Basics/ Protocol Operations

UAC
User1@host1

Proxy/
Registrar

Location
Server

UAS
User2@host2

INVITE

Location update/OK

200 OK

REGISTER

Location query/Reply

INVITE

1-way media transmission

200 OK

ACK

ACK

2-way media transmission

BYE

200 OK

BYE

200 OK

200 OK
Expandability/ Requirements

Req1: The problem must fit in the *solution space* of SIP

Req2: The extension must conform to the SIP *architectural model*

Solution space: User/service discovery for delivering a message
The message may contain session description, capability query, instant message, etc.

Architectural model: Simplicity and heterogeneity
simple transactions, simple proxies, multi-protocol, multi-provider, etc.
Expandability/ Principles

- Protocol elements that can be extended without change in the protocol version: Method, Entity header, Response code, Event type
- Proxies process all new methods like BYE and ignore new header fields
- Extension negotiation is based on unique option tag and some headers (Require, Proxy-Require, Supported, Unsupported)
- If extension is required but not supported or allowed
  - UAS responds with 420 Bad Extension or 501 Not Implemented (method) or 405 Method Not Allowed
  - UAC sends CANCEL if unknown method or extension received
- All responses MAY include a body which can be extended independently since proxies ignore the body
- Capability query with OPTION method returns headers
  - Allow supported methods
  - Supported supported extensions (option tags)
  - Accept supported content types (body types)
Examples: Reliable Provisional Responses

Retransmission algorithm for INVITE effective

UAC

INVITE sip:uas@host SIP/2.0
Supported: 100rel

UAS

SIP/2.0 180 Ringing
Require: 100rel
RSeq: 776655

Retransmission algorithm for PRACK effective

PRACK sip:uas@host SIP/2.0
RAck: 776655 1 INVITE

(retransmission of 180)

(retransmission of PRACK)

SIP/2.0 200 OK (for PRACK)
Extensions for Call Stateful Proxies

- The core protocol makes implementation of stateless and transaction stateful proxy rather simple
- Some extensions are to simplify implementation of call stateful proxy
  - "Session Timer" (timer)
    - for releasing unterminated calls
    - based on keep alive mechanism
  - "Distributed Call State"
    - for stateful proxy to behave statelessly
    - based on extension headers carrying the call state (cookies)
Resource Management/1

- Req1: Call signalling (telephony service) and resource mgmt signalling (network layer) must be separated
- Req2: QoS and security establishment are preconditions before the phone rings. "Call blocking" acceptable before the phone rings but not after the phone rings (call defect).
- How to coordinate?
- Two-phase model for call establishment
- SDP defines the preconditions since they are media related
- SIP: COMET method, 580-Precondition-Failure response code
- Any end-to-end resource reservation mechanism (RSVP, IPSec) can be used, no new reservation mechanism defined
Resource Management/2

- "a=qos:" SP strength-tag SP direction-tag [SP confirmation-tag]
- "a=secure:" SP strength-tag SP direction-tag [SP confirmation-tag]
- "confirm" attribute present => recipient sends a COMET message to sender, with SDP attached, telling the status of each precondition as "success" or "failure"
- UAS returns a provisional response if it's capable of honoring the precondition or 580 if it's not
- Example: single-media session with a "mandatory" quality-of-service "sendrecv" precondition, where both the UAC and UAS can only perform a single-direction ("send") resource reservation.
- Backward compatible:
  - UAS ignores if it does not recognise the attributes
  - UAC may enforce with "Require: precondition" tag
Resource Management/3

UAC
INVITE
m=audio 49170 RTP/AVP 0
a=qos:mandatory sendrecv

183 Session-Progress
a=qos:mandatory sendrecv confirm

COMET
a=qos:success send

200 OK (of COMET)

180 Ringing

200 OK

ACK

UAS

(reservation)

(reservation)

UAS Guarantees that all preconditions are met before alerting the user

User picks-up the phone

UAS Guarantees that all preconditions are met before alerting the user
SIP Notification/1

• For asynchronous notification of events (callback services)
• Similar to proven software design pattern (Gamma et.al)
• New methods SUBSCRIBE and NOTIFY
• New headers: Event and Allow-Events
• New response codes: 202 Accepted and 489 Bad Event
• No extension token needed since caller first issues new method
• Request body may contain filters/throttles and response body the event/state
• Guidelines:
  • Not for very frequent events, not for very large data
  • Send the new state together with the event
  • Both subscriptions and notifications must be authenticated/authorised
• Open/vague issues:
  • Should/can implicit subscription be forbidden? (currently no)
  • Should general mechanism for filters/throttles be defined? (currently no)
  • Authorisation of subscription (QAUTH)
SIP Notification/2

<table>
<thead>
<tr>
<th>Subscriber</th>
<th>Notifier</th>
<th>End user</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBSCRIBE/202 Accepted</td>
<td></td>
<td>authorise</td>
</tr>
<tr>
<td>NOTIFY/200 OK</td>
<td></td>
<td>accepted</td>
</tr>
<tr>
<td>NOTIFY/200 OK</td>
<td></td>
<td>state change notification</td>
</tr>
<tr>
<td>unsubscribe</td>
<td>SUBSCRIBE Expires: 0/202 Accepted</td>
<td>final notification</td>
</tr>
<tr>
<td>NOTIFY/200 OK</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Call Transfer/1

- REFER method is one of the call control extensions to SIP
  REFER sip:carol@cleveland.com SIP/2.0
  Refer-To: sip:alice@atlanta.com;method=INVITE?Call-Id=...
  Referred-By: "Bob" <sip:bob@biloxi.com>;
    ref=<sip:alice@atlanta.com>;scheme=pgp;
    pgp-version="5.0";signature=...

- NOTIFY is implicitly used for reporting the results to the referer
  - referrer and referee must implement SIP events
  - call transfer is dependent on SIP events (Event: refer)
  - referrer gets rid of the call only after the notification

- Notification body contains the response status of the referred action
  Content-Type: application/sip
  "SIP/2.0 200 OK" ("SIP/2.0 503 Service Unavailable")

- Risk of privacy to the Refer-To party since
  - REFER is not restricted to a call context
  - UAS of Refer-To party may not recognise a referred call
  - body may contain all response headers of the referred action
Call Transfer/2

- Several types of call transfer can be implemented with REFER
- Unattended transfer:
  - Transferor not participating the call simultaneously with both Transferee and Transfer Target
  - drafted for several variations
- Attended transfer:
  - the three actors participating the call simultaneously (ad-hoc conference)
  - not drafted yet since conferencing has not been addressed in the call control framework
- Consultation:
  - Transferor establishes a second call with Transfer target before transferring the first call
- Open: How is it guaranteed that Transferee reaches the same instance of Transfer Target Transferor was calling?
Call Transfer/3
Unattended Transfer with Consultation Hold

Transferor

Transferer

INVITE/200/ACK
INVITE(hold)/200/ACK
INVITE/200/ACK
BYE/200
REFER/202 Accepted
NOTIFY/200
BYE/200
INVITE/200/ACK
BYE/200

Transferee

Consultation

Transfer Target

Call put on hold

Call terminated

Call put on hold
SIP Presence and Instant Messaging

- **Presence**: User's reachability, capabilities and willingness to communicate with other users

- **Instant Messaging**: Exchange of messages between participants in real time

- **Protocol development**:
  - SIMPLE wg (SIP for Instant Messaging and Presence Leveraging)
  - Keep the two protocols separate (applications tend to mix)

- **Foundation for using SIP**:
  - SIP registrar holds the presence information
  - SIP proxies route the messages
  - SIP events for delivering the presence information
SIP Presence

- **SIP Presence logical elements:**
  - Presence Agent (PA) accepts and stores subscriptions, detects state changes and generates notifications
  - Presence User Agent (PUA) pushes data into the presence system using REGISTER, for example
  - CPIM gateway (subscription stateful when presence is managed by CPIM))

- **SIP Presence extension suggests**
  - protocol independent presence URI `pres:user@domain`
  - REGISTER to have presence body and description parameter for Contact
  - new event package `presence`
  - XML body and MIME type for the presence information in NOTIFY
  - no body format defined for SUBSCRIBE
  - standard SIP mechanism for signing/encrypting (PGP, signed-by)

- **Open/odd issues:**
  - REGISTER body has not been defined
  - SUBSCRIBE establishes a session without INVITE
  - The draft is not very mature yet.
SIP Presence Framework

UA1 -> SUBSCRIBE
UA1 -> NOTIFY

Proxy/Registrar

UA2 -> REGISTER

UA3

SIP Presense System

UA2

UA3

Presence User Agent

SIP/CPIM Gateway

CPIM Presence System

---

SIP protocol

Non-SIP protocol
Example of SIP Instant Messaging

User1

MESSAGE im:user2@domain.com SIP/2.0
From: im:user1@domain.com
To: im:user2@domain.com
Contact: sip:user1@pc1.domain.com

Proxy

MESSAGE sip:user2@pc2.domain.com SIP/2.0
From: im:user1@domain.com
To: im:user2@domain.com
Contact: sip:user1@pc1.domain.com

SIP/2.0 200 OK
From: im:user1@domain.com
To: im:user2@domain.com;tag=ab8asd9
Contact: sip:user2@pc2.domain.com

User2

SIP/2.0 200 OK
From: im:user1@domain.com
To: im:user2@domain.com;tag=ab8asd9
Contact: sip:user2@pc2.domain.com

MESSAGE sip:user1@pc1.domain.com
From: im:user2@domain.com;tag=ab8asd9
To: im:user1@domain.com
Contact: sip:user2@pc2.domain.com
Conclusions

• The core protocol must be simple for easy implementation and interoperability. How to keep it that way?
• The semantics of core protocol must be solid. Some semantics is still open and the specification still in proposed state!
• Extension mechanism has been prosperous. How to prevent overlapping/contradictory extensions?
• Does the standardisation process stand the 3GPP schedule without divergence?
Thanks for your attention!

Jouni