Session Initiation Protocol

SIP protocol and its extensions

SIP – Need for signaling compression

Main Sources

IETF:

RFC 3261: SIP: Session Initiation Protocol
RFC 3262: Reliability of Provisional Responses in SIP (PRACK)
RFC 3265: SIP Specific Event Notification
RFC 3311: SIP UPDATE method
RFC 3428: SIP Extension for Instant Messaging
RFC 2327: SDP: Session Description Protocol
RFC 3264: An Offer/Answer Model with Session Description Protocol (SDP)

3G Releases 5 … 7:

3GPP TS 24.228 v5.15.0 (2006-10) Signaling flows for the IP MM call control based on SIP and SDP; stage 3 (Release 5)
3GPP TS 24.229 v7.11.0 (2008-03) (rel 7) IP multimedia call control protocol based on SIP and SDP, Stage 3 (Release 7)
3GPP TS 29.228 v7.4.0 (2006-12) (for Rel 7) IMS Cx and Dx interfaces, Signaling flows and message contents; (Release 7)
Etc…

Some SIP WG Documents

Internet-Drafts (total of 27 on March 24th 2008) E.g.:
Connection Reuse in the Session Initiation Protocol (SIP) (46022 bytes)
Location Conveyance for the Session Initiation Protocol (113455 bytes)
Certificate Management Service for The Session Initiation Protocol (SIP) (65151 bytes)
SIP SAML Profile and Binding (70037 bytes)
Conference Establishment Using Request-Contained Lists in the Session Initiation Protocol (SIP) (29891 bytes)
Indicating Support for Interactive Connectivity Establishment (ICE) in the Session Initiation Protocol (SIP) (15402 bytes)
Requirements and Analysis of Media Security Management Protocols (105192 bytes)
Domain Certificates in the Session Initiation Protocol (SIP) (34730 bytes)
An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Diff Event Package (30834 bytes)

Request For Comments (total of 50 on March 24th 2008), E.g.:
The SIP INFO Method (RFC 2976) (17736 bytes)
MIME media types for ISUP and QSIG Objects (RFC 3204) (19712 bytes) updated by RFC 3459
SIP: Session Initiation Protocol (RFC 3261) (647976 bytes) obsoletes RFC 2543/ updated by RFC 3853,RFC 4320,RFC 4916
Reliability of Provisional Responses in SIP (RFC 3262) (29643 bytes) obsoletes RFC 2543
The Session Initiation Protocol UPDATE Method (RFC 3311) (28125 bytes)
Session Initiation Protocol Extension for Instant Messaging (RFC 3428) (41475 bytes)
Compressing the Session Initiation Protocol (RFC 3486) (24181 bytes)
The Session Initiation Protocol (SIP) Refer Method (RFC 3515) (47788 bytes)
An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing (RFC 3581) (29121 bytes)
Caller Preferences for the Session Initiation Protocol (SIP) (RFC 3841) (61382 bytes)
The Session Inititation Protocol (SIP) 'Join' Header (RFC 3911) (35373 bytes)
The Stream Control Transmission Protocol (SCTP) as a Transport for the Session Initiation Protocol (SIP) (RFC 4168) (21079 bytes)
Guidelines for Authors of Extensions to the Session Initiation Protocol (SIP) (RFC 4485) (57278 bytes)
Management Information Base for the Session Initiation Protocol (SIP) (RFC 4780) (160460 bytes)
Rejecting Anonymous Requests in the Session Initiation Protocol (SIP) (RFC 5079) (15670 bytes)

Source: http://www.ietf.org/html.charters/sip-charter.html

Some SIMPLE and XCON WG Documents

Aug 2007 Conclusion of SIMPLE
Internet-Drafts(12 in total):

http://www.ietf.org/html.charters/simple-charter.html

Request For Comments (Total of 17) E.g.:
A Presence Event Package for the Session Initiation Protocol (SIP) (RFC 3856) (62956 bytes)
A Watcher Information Event Template-Package for the Session Initiation Protocol (SIP) (RFC 3857) (46221 bytes)
An Extensible Markup Language (XML) Based Format for Watcher Information (RFC 3858) (26416 bytes)
A Data Model for Presence (RFC 4479) (88399 bytes)
CIPID: Contact Information in Presence Information Data Format (RFC 4482) (22157 bytes)
The Extensible Markup Language (XML) Configuration Access Protocol (XCAP) (RFC 4825) (166627 bytes)
Relay Extensions for the Message Sessions Relay Protocol (MSRP) (RFC 4976) (84244 bytes)
The Message Session Relay Protocol (MSRP) (RFC 4975) (144254 bytes)

http://www.ietf.org/html.charters/xcon-charter.html

XCON (Centralized Conferencing)
Internet-Drafts (total of 3 in March 24th, 2008 – no actions recorded in 2007-08)

XCON Request For Comments:
Requirements for Floor Control Protocol (RFC 4376) (30021 bytes)
Conferencing Scenarios (RFC 4597) (38428 bytes)
The Binary Floor Control Protocol (BFCP) (RFC 4582) (154497 bytes)
Connection Establishment in the Binary Floor Control Protocol (BFCP) (RFC 5018) (20244 bytes)

Peer-to-Peer Session Initiation Protocol (p2psip)
- a new WG. One draft so far on March 24th, 2008

SIPPING WG documents

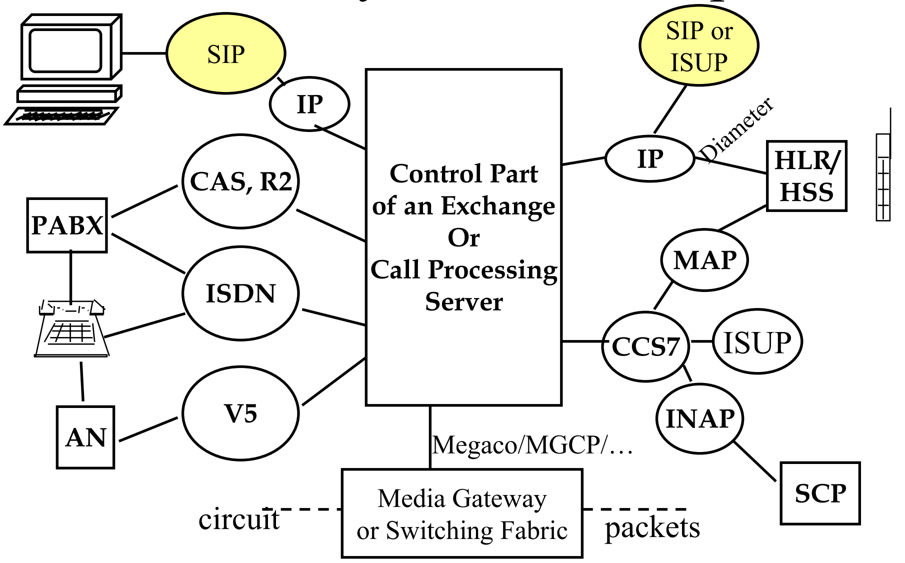
Source: http://www.ietf.org/html.charters/sipping-charter.html

Internet-Drafts (Total of 25 on March 24th, 2008) E.g.:
Session Initiation Protocol Service Examples (289129 bytes)
A Call Control and Multi-party usage framework for the Session Initiation Protocol (SIP) (100427 bytes)
Session Initiation Protocol Call Control - Transfer (126484 bytes)
The Session Initiation Protocol (SIP) Conference Bridge Transcoding Model (24627 bytes)
A Session Initiation Protocol (SIP) Event Package for Session-Specific Session Policies (42978 bytes)
Requirements from SIP (Session Initiation Protocol) Session Border Control Deployments (53815 bytes)
SIP (Session Initiation Protocol) Usage of the Offer/Answer Model (61640 bytes)
Request For Comments (Total of 34 on March 24th) E.g.:
User Requirements for SIP in Support of Deaf, Hard of Hearing and Speech-impaired individuals (RFC 3351) (33894 bytes)
Session Initiation Protocol (SIP) for Telephones (SIP-T): Context and Architectures (RFC 3372) (49893 bytes)
ISDN User Part (ISUP) to Session Initiation Protocol (SIP) Mapping (RFC 3398) (166207 bytes)
SIP and SDP Static Dictionary for Signaling Compression (SigComp) (RFC 3485) (80195 bytes) updated by RFC 4896
Mapping of ISUP Overlap Signalling to the Session Initiation Protocol (SIP) (RFC 3578) (26667 bytes)
Session Initiation Protocol Basic Call Flow Examples (RFC 3665) (163159 bytes)
Session Initiation Protocol PSTN Call Flows (RFC 3666) (200478 bytes)
Authentication, Authorization and Accounting Requirements for the Session Initiation Protocol (RFC 3702) (31243 bytes)
A Session Initiation Protocol (SIP) Event Package for Registrations (RFC 3680) (56306 bytes)
Using E.164 numbers with the Session Initiation Protocol (SIP) (RFC 3824) (36535 bytes)
Requirements for End-to-Middle Security for the Session Initiation Protocol (SIP) (RFC 4189) (25842 bytes)
A Framework for Conferencing with the Session Initiation Protocol (SIP) (RFC 4353) (67405 bytes)
Interworking between the Session Initiation Protocol (SIP) and QSIG (RFC 4497) (149992 bytes)
Guidelines for Usage of the Session Initiation Protocol (SIP) Caller Preferences Extension (RFC 4596) (82954 bytes)
Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents (RFC 4579) (96506 bytes)
A Session Initiation Protocol (SIP) Event Package for Key Press Stimulus (KPML) (RFC 4730) (120186 bytes)
The Session Initiation Protocol (SIP) and Spam (RFC 5039) (73341 bytes)

Recent additions in IMS

- VCC – voice call continuity allows users to roam between IP-CAN/IMS (e.g. WLAN access) and Circuit switched domain without loosing their call (24.206 v 7.4.0)
  - IP-CAN can be any IP access network based on cellular, WLAN, ADSL or other technology
- Better NAT traversal (SIP outbound and ICE)
- IBCF – Interconnection Border Control Function procedures. IBCF hides networks from each other, is an application level gateway (another term: session border controller)
- Emergency Service
- PSTN/ISDN simulation services for NGN

Summary of course scope



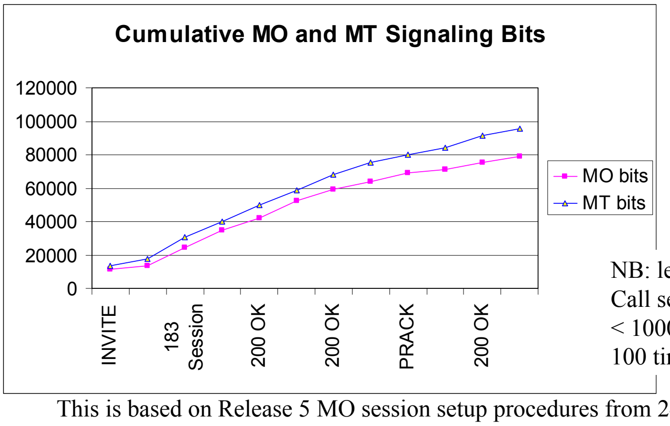
SIP Requirements and fundamentals

- Part of IETF toolkit
  - Reusing other protocols & mechanisms: HTTP, etc.
  - Flexible
  - Extensible
- Moves intelligence to End System entities
  - End-to-end protocol
- Interoperability
- Scalability (although some state in network)
- Service creation easy
- URLs and Addresses are reused
- Same routing as SMTP
- Reuses infrastructure (all applications will use SIP entities for different services)

SIP overview

- Simplicity
  - Text based (UTF-8 encoded) - simple tools for development but long messages
  - Lower call setup time than in H.323 (depends on sig ch speed)
  - basic protocol + extensions structure adopted
- Caller preferences, Ability to support many media types
- Runs over UDP or TCP or SCTP
- Used between both service and call control entities
- Has been adopted for 3G IP Multimedia signaling
- Originally subscriber signaling, in 3G also network to network signaling
- Development in 3GPP has added a lot of complexity
- Market view: there are 3 important variants of SIP: 3G, IETF and Microsoft Messanger.
- Text based + transaction oriented → lengthy messages and flows

Text based Signaling in IMS produces a lot of bits to the air interface in a cellular network

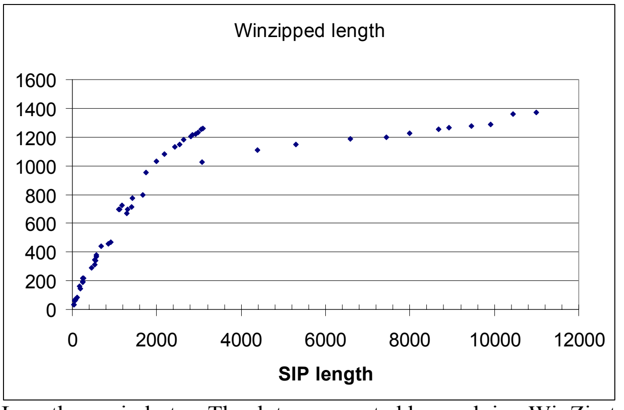


Cmp to ISDN

NB: length of DSS1(L2+L3) Call setup and release < 1000 bits→ SIP/SDP uses 100 times more bits

This is based on Release 5 MO session setup procedures from 24.228
+ The MO flow will create as many air interface bits as talking for 18 seconds using AMR 4.75 codec.
+ About 70% of the bits are created by redundant lines of text.
+ There is more redundancy on information element level

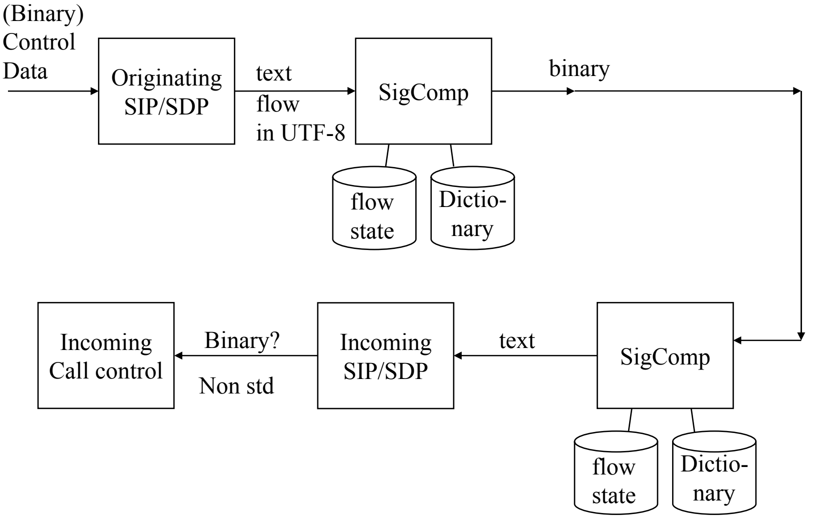
Zipping analysis of MO flow content shows that SIP/SDP carries a lot of context information in each transaction



The MO flow is for 3G release 5.

Lengths are in bytes. The dots are created by applying WinZip to all incrementally longer subflows of the MO flow. The curve shows that by keeping state, we would save quite a lot in signaling bits.

Signaling compression model



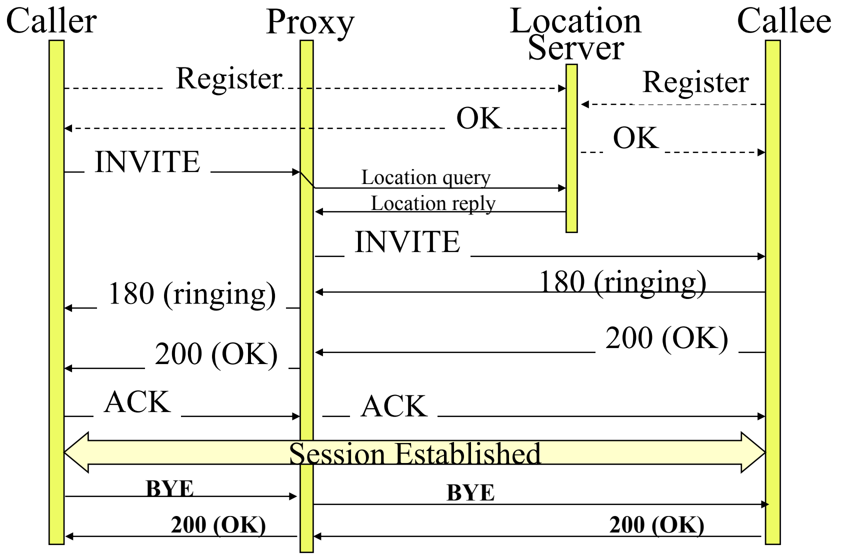
SigComp allows cmpression of Signaling Messages

- NB: *Signaling Compression* is different from *Header Compression.* Signaling Compression deals with the compression of the payload= signaling messages themselves and does not touch TCP/IP header (or RTP/UDP/IP header)!
- RFC 3320 and RFC 3321 specify a layer between the signaling transport and the signaling application
- Uses Global and User Specific Dictionaries to store state data over many SIP sessions
  - Dictionary is enlarged dynamically
  - Efficiency is improved by using all previous messages in a flow as source of the dictionary – incremental application of compression – this requires quite a lot of memory
  - After dictionary replacement, Huffman coding can be applied.
- Overall Compression/decompression architecture is based on a bytecode driven Universal Decompression Virtual Machine
  - Bytecode can be sent in SigComp messages by the Compressor
  - leaves a lot of detail for the implementor

Sip Entities

- User Agents
  - Can act as client and as server
- Servers (network based):
  - Redirect Servers
    - Send back alternative location of the user (similar as HTTP servers)
  - Proxy servers
    - Act on behalf of client (forwards requests)
    - Forking proxies
  - Registrars
    - Accepts registrations and maps public SIP URIs to user locations.
  - Location Servers (not part of SIP architecture)
    - Gives back location of user (received from registrars)
    - E.g. HSS in 3GPP IMS architecture
    - Protocol between Location server and SIP server not defined by SIP specs (e.g. LDAP)

Basic SIP call setup and release



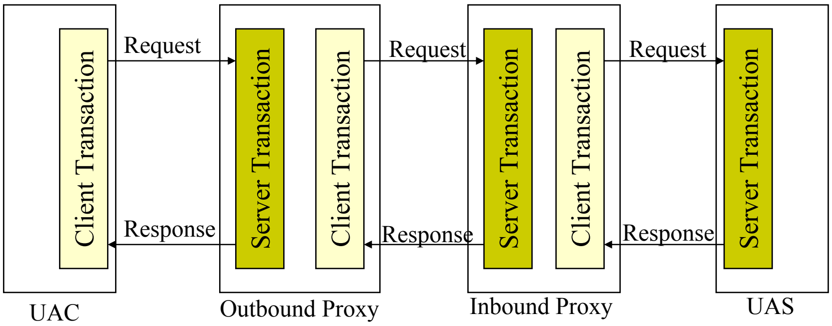
”Basic call” Example (IETF)

- Caller sends INVITE
- Callee can accept, reject, forward the call
- If the callee accepts the call: responds with an optional provisional (1xx), and a final (≥200) response
- The caller confirms final response via ACK
- Conversation
- Caller or callee sends BYE
- BYE is acknowledged by 200 OK
- Low call setup times, post dial delay: 1.5 RTT !

SIP messages have headers and a body

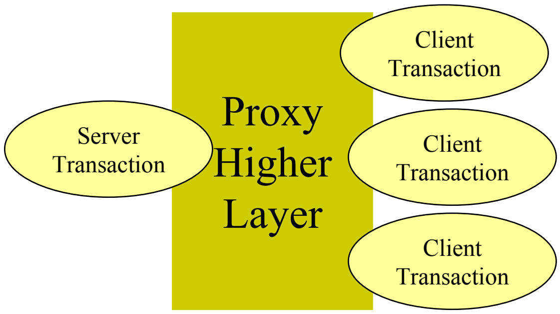
- Headers carry control information and are processed e.g. by Proxies
- Body can be e.g. SDP – session description protocol
  - end-to-end information (cmp H.245 in the H.323 framework) describing session requirements e.g. coding methods, etc
- Message delivery is transaction oriented= have request + reply: e.g INVITE+200 OK

User Agent is split into User Agent Client (UAC) and User Agent Server(UAS)



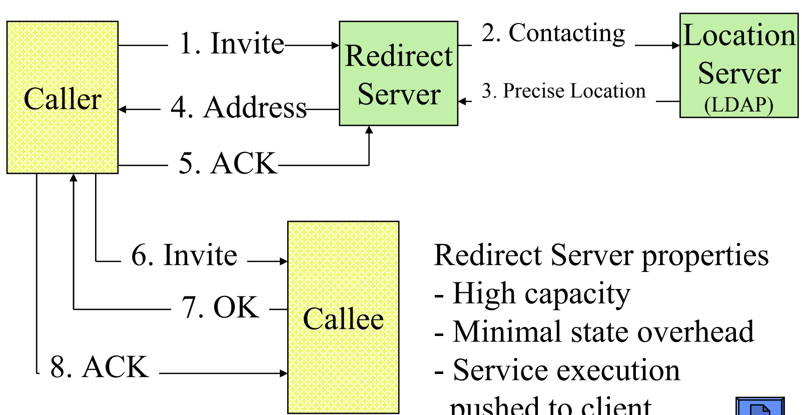
- All Communication follows this transaction model, except 2xx and ACK.
- Server transactions filter incoming requests, absorbing retransmissions
- Only UAS can generate 2xx and only UAC can generate ACK (to success).

A Stateful Proxy can fork a transaction



Forking = multicast of INVITEs to N addresses

Redirect Server pushes processing to clients



Redirect Server properties
- High capacity
- Minimal state overhead
- Service execution pushed to client

17.3.04

Stateful Proxy vs Stateless Proxy

- Maintains call context
- Replicates UAS/UAC to process requests and responses
- Call state and transaction state can be maintained
- Forking proxies require state
- TCP proxies must be stateful for reliability
- Enhanced services require state
- Can collect charging info
- All transactions in a session are processed by the same proxy (computer)

- No call context
- Response is not based on UA replication
- Provides client anonymity
- Restricted gateway access
- High processing capacity
- Easier to replicate than the stateful proxy
- Consecutive transactions in a session can be processed on different computers → load sharing is easy to arrange for e.g. x00 million users
- Also semi-stateful is possible

UA = User Agent, UAC = UA Client
UAS = UA Server

List of SIP methods

ACK	Acknowledges the establisment of a session
BYE	Terminates a session
CANCEL	Cancels a pending request
INFO	Transports PSTN telephony signaling
INVITE	Establishes a session
NOTIFY	Notifies a User Agent of a particular event
OPTIONS	Queries a server about its capabilities
PRACK	Acknowledges a provisional response
PUBLISH	Uploads information to a server
REGISTER	Maps a public URI with the current location of the user
SUBSCRIBE	Requests to be notified about a particular event
UPDATE	Modifies some characteristic of a session
MESSAGE	Carries an instant message
REFER	Instructs a server to send a request

Blue methods are candidates for AS processing
includes both the base protocol and extensions from year 2004

Notes on SIP methods: INVITE

- Requests users to participate in a session. Body contains description of the session. If someone wants to modify the parameters of the session, he/she must re-INVITE carrying the modified parameters.
- One or more Provisional and one Final response are expected.

Notes on SIP methods: ACK

- Acknowledges the Final Response to INVITE even if INVITE was cancelled → result is 3-way handshake: INVITE-final-resp – ACK.
- Proxies can only ACK non-successful Final Resp.
- Purpose:
  - Let's the server know that session establishment was successful.
  - Forking may result in many final responses. Sending ACKs to every destination that sent a final response is essential to ensure working over UDP.
  - Allows sending INVITEs without session description. In this case the description is postponed to ACK.
- Has the same Cseq as the INVITE it acknowledges (see later for SIP headers).

Notes on SIP methods: CANCEL

- Purpose: to cancel pending transactions. Will be ignored by completed transactions = final response already sent.
  - useful for forking proxies. If one destination answered, the forking proxy can cancel all other pending INVITEs.
- Has the same Cseq as the request it cancels (see later for SIP headers).

Notes on SIP methods: REGISTER

- Purpose: to register the user's current location.
  - This means that regular SIP needs a network based server to work. Need for other network based servers depends on existence of NAT's, operator preferences etc.
  - There is an effort called Peer-to-Peer SIP that intends to replace Registrar by another solution
- A user can be registered in several locations at the same time. Forking is used to find out where the user wants to answer the session invitation.
- A user can register from anywhere to his registrar → provides mobility.

Notes on SIP methods: OPTIONS

- UA can query a server: which
  - methods and
  - extensions and
  - which session description protocols it supports.
  - which encoding for message bodies the server understands (e.g. compression to save bw).

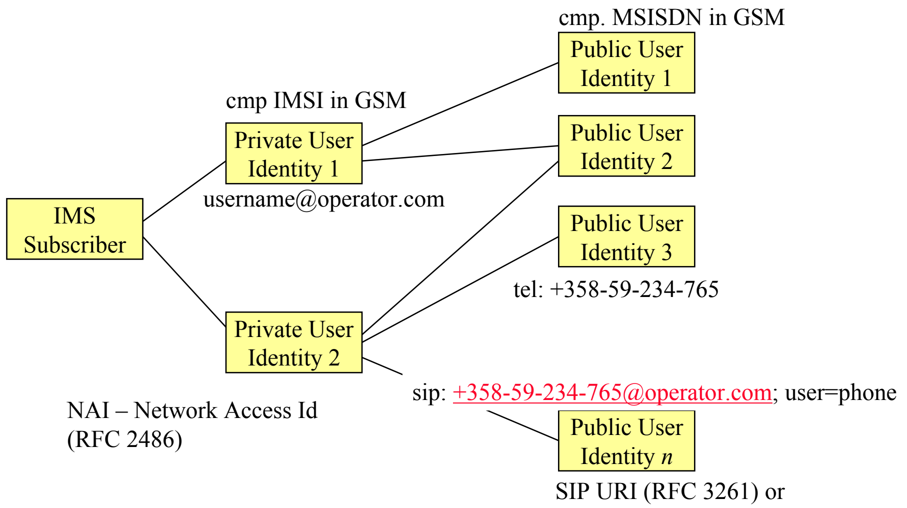
Some SIP issues

• Parties can release the "call session" but since they have obtained each others IP-addresses, they can continue sending media streams to each other!! • How to push INVITE to B-party, if B-party does not have a permanent IP address which is most often the case!	Integration of Proxy with Firewall and NAT, STUN, ICE SIP outbound
• Response messages (e.g. 180) are not reliably delivered. This may cause tear down of the call if it was initiated from ISDN	PRACK method
• If BYE is lost, Proxy does not know that call has ended	KeepAlive = re-INVITE mechanism
• Text based coding increases the signaling overhead → problem in Radio access	SigComp

Identification of users

- **sip:user@host[parameters][headers]**
- SIP URIs are like URLs, with prefix sip: which gives schema
  - sip:joe.smith@hut.fi
  - sip:joe.smith@hut.fi?subject=Protocol
  - sip:sales@hotel.xy;geo.position:=48.54_-123.84_120
- Address must include host, other parameters are optional (username, port, etc…)
- Email-addresses can be reused
- “Click-to-call” on web-pages, MM messages, etc… are easily implemented

Identification of users in 3G IMS in R6



In Release 5 only one Private User id

Requests invoke SIP methods

- SIP methods are invoked on servers when requests arrive:
  - A REGISTER request sends location information of users to Registrars, registers with the location service
  - An INVITE request invites a user to participate in a session or conference
    - The message body contains a description of the session (usually SDP)
  - ACK requests are used to confirm responses for INVITE, for reliable message exchanges
  - CANCEL requests cancel the pending request of the session
  - BYE requests are used to terminate active sessions
    - Any party of the session can send it
  - OPTIONS requests are used to query information about servers' capabilities
  - PRACK requests are used to confirm provisional responses

SIP responses are classified by first digit

- HTTP look-alike
- Hierarchically organized three digit codes: status code - text associated with the code
- Provisional and final responses:
  - 1xx responses are informational messages e.g., 180 Ringing
  - 2xx response shows a successful transaction e.g., 200 OK
  - 3xx responses are redirect messages e.g., 301 Moved Permanently
  - 4xx responses indicate errors in requests e.g., 400 Bad Request
  - 5xx responses indicate server errors e.g., 500 Version not supported
  - 6xx responses indicate global failures e.g., 600 Busy everywhere

SIP Message Format

- START-LINE
  - SIP version used
  - In requests: address and method used
  - In responses: status code
- HEADERS
  - Information about call
- BODY (payload)
  - Usually SDP message

```
C->S: INVITE sip:bob@biloxi.com SIP/2.0          <- Start line
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710                          } Headers
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142

 v=0
 o=UserA 2890844526 2890844526 IN IP4 here.com
 s=Session SDP
 c=IN IP4 pc33.atlanta.com                       } Body
 t=0 0
 m=audio 49172 RTP/AVP 0
 a=rtpmap:0 PCMU/8000
```

A Request line contains the Request URI = the name of the user that is the destination. This request URI is used for SIP routing.


```
C->S: INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142

 v=0
 o=UserA 2890844526 2890844526 IN IP4 here.com
 s=Session SDP
 c=IN IP4 pc33.atlanta.com
 t=0 0
 m=audio 49172 RTP/AVP 0
 a=rtpmap:0 PCMU/8000
```

To and From header fields

- **To: specifies the logical call destination**
- **From: specifies the logical call source**
- **Present in all SIP messages**

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
```

NB: In IMS To and From header field are end to end information and not used nor asserted by the network. Instead the P-CSCF inserts the **P-Asserted-Identity** header field into the SIP messages indicating the network asserted identity of the originator of the call

Call-ID and CSeq header fields

- **Call-ID: It helps to uniquely identify a particular SIP dialog or registration**
  - It helps to match requests and responses
  - It helps to detect duplicated messages
- **CSeq: It is a number that uniquely identifies the transaction in a call**
- **Present in all SIP messages**

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
```

Content-Type and Content-Length header fields

- **Content-Type: It describes the media type of the message body**
- **Content-Length: The number of octets in the message <u>body</u>**
  - It is mandatory in all SIP messages.

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
```


Max-Forwards

- **Max-Forwards field must be used with any SIP method**
- **It limits the number for proxies or gateways on the way of SIP message to the destination.**

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
```

VIA header indicates path taken by the request so far

- **Branch parameter is used to detect loops**
- **Contains transport protocol, client's host name and possibly port number, and can contain other parameters**

```
INVITE sip:bob@192.0.2.4 SIP/2.0
Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.1
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com;branch=z9hG4bK77ef4c2312983.1
 ;received=192.0.2.2
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
 ;received=192.0.2.1
Max-Forwards: 68
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
```


Record-route and Route

- **Record-Route: header is added by proxy, when proxy wants to stay in the route of all sip messaging**
- **Route is added by User Agent Client, after response came, with all Record-route headers in it (then UAC knows which relays want to stay on the signaling path)**
- **NOT the same as Via: headers**

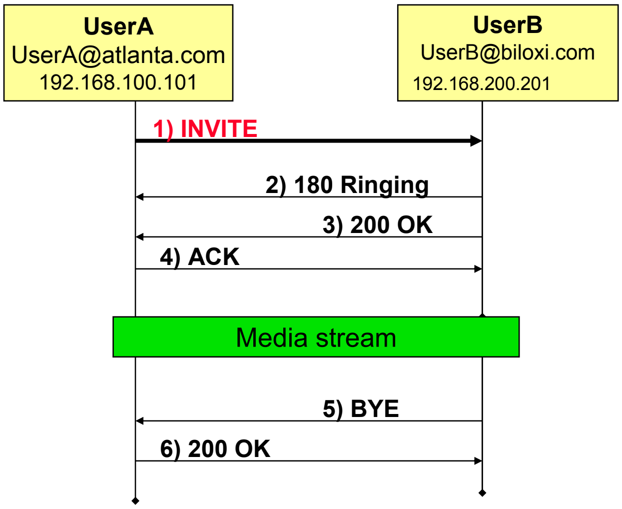
```
INVITE sip:callee@u2.domain.com SIP/2.0
Contact: sip:caller@u1.example.com
Record-Route: <sip:p2.domain.com;lr>
Record-Route: <sip:p1.example.com;lr>
```

Inserted by proxies p1.example.com and p2.example.com.

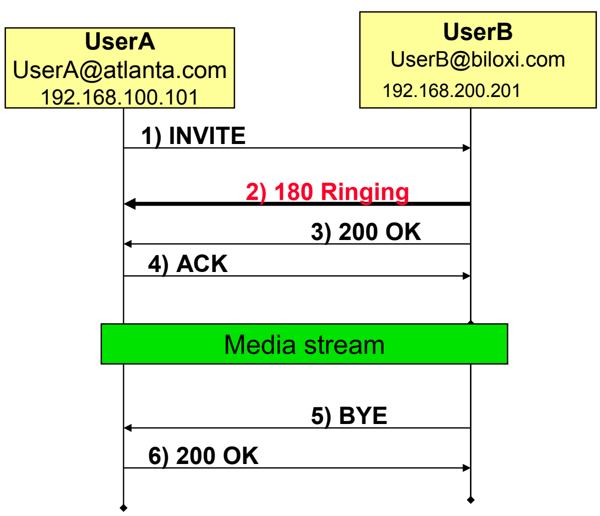
```
BYE sip:callee@u2.domain.com SIP/2.0
Route: <sip:p1.example.com;lr>,<sip:p2.domain.com;lr>
```

UA can specify through which proxies this message must go

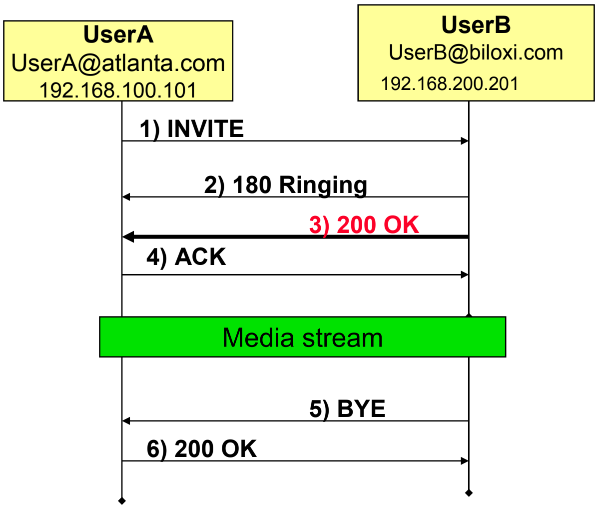
”Basic Call” call flow



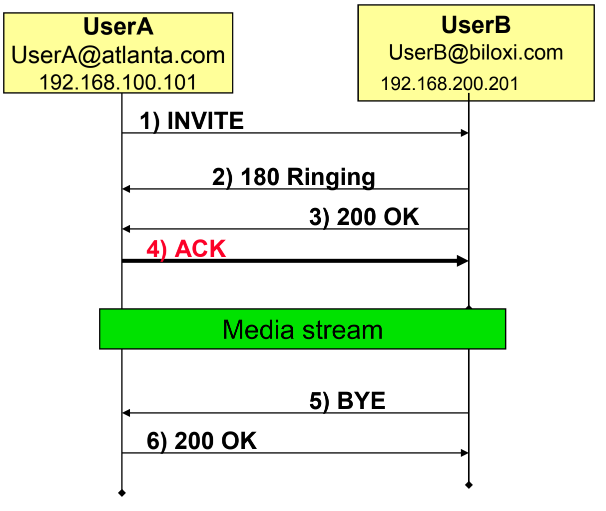
”Basic Call” call flow



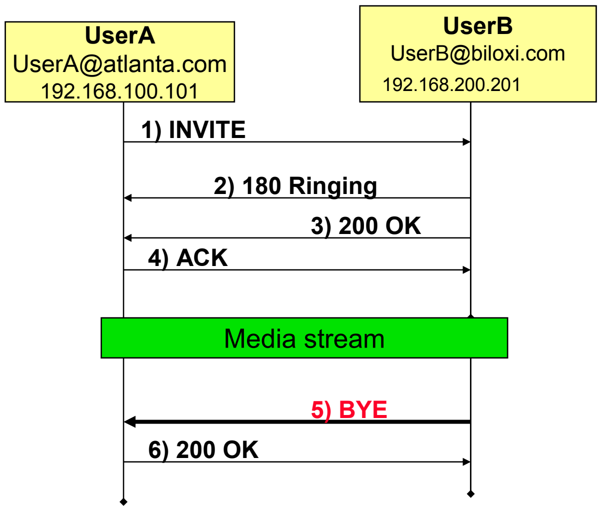
”Basic Call” call flow



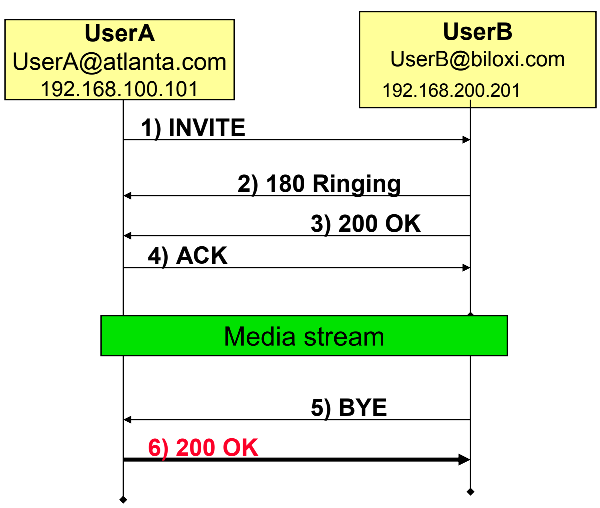
”Basic Call” call flow



”Basic Call” call flow



”Basic Call” call flow



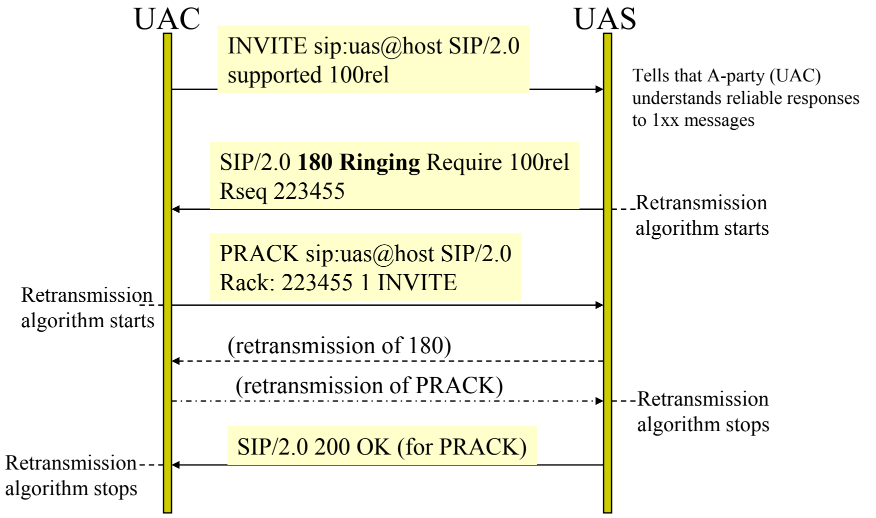
SIP Extensions

- Needed to satisfy additional requirements
- Must conform to design rules
- SIP is not intended to solve every problem (another protocol might be used instead)

Feature Negotiation (OPTIONS)

- *Supported* features can be specified in requests and responses
  - **Supported** UAC and UAS tell features they support
- *Required* features can be specified in requests and responses
  - **Require** UAC tells UAS about required options
  - **Proxy-Require** required options for proxy/redirect servers
  - Many extensions use Require and Proxy-Require to specify their support
- New methods can be added without changing the protocol
  - server can respond with **405 Not Supported**
  - returns list of supported methods in **Allow** header
  - client can ask which methods are supported using OPTIONS

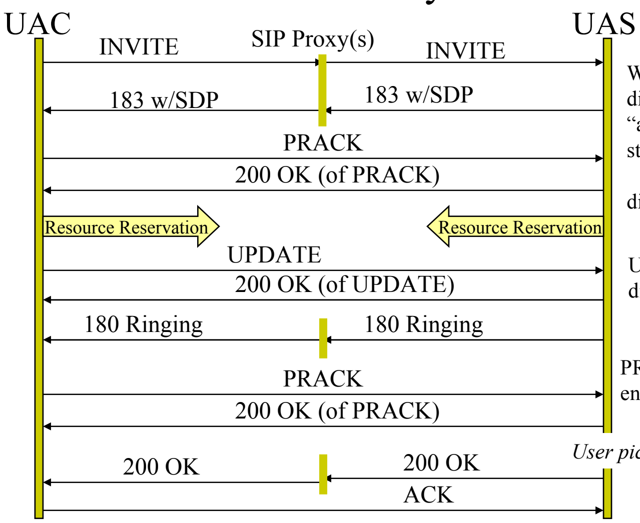
Reliable Provisional response in SIP



QoS support - UPDATE

- Usage rule for 183-Session-Progress
  - If “a=qos” appeared in SDP, UAS sends 183 with “Session: qos” and SDP
- Additional Method - UPDATE
  - If “a=qos” appeared in SDP with “confirm” attribute, UAS/UAC sends UPDATE with success/failure status of each precondition.
  - 200 OK must acknowledge the UPDATE message
  - user B does not need to be prompted
- Additional Status Response - 580 Precondition Failure
  - If a mandatory precondition can’t be met, UAS terminates INVITE with this status response

Phone should not ring before QoS and Security are OK



W/SDP = "a-qos:" strength direction
"a-secure:" strength direction
strength = mandatory|optional |success|failure
direction = send|recv|sendrecv

UPDATE confirms that preconditions are OK at the originator

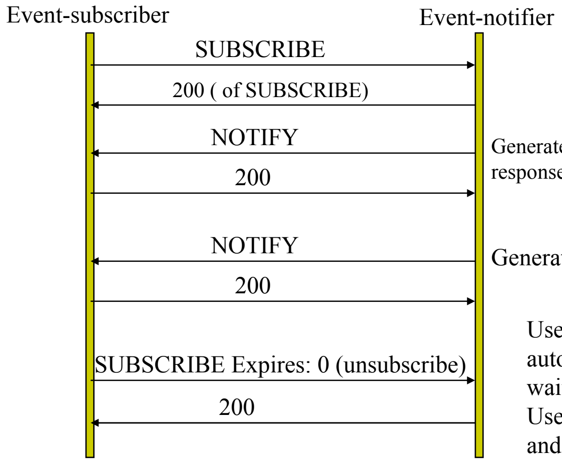
PRACK method is used to ensure delivery of 183 and 180

User picks up the phone

Use case: 3G signaling!

SDP = Session Description Protocol (carried in SIP message body)

SIP event notifications tell about remote significant events to the local party

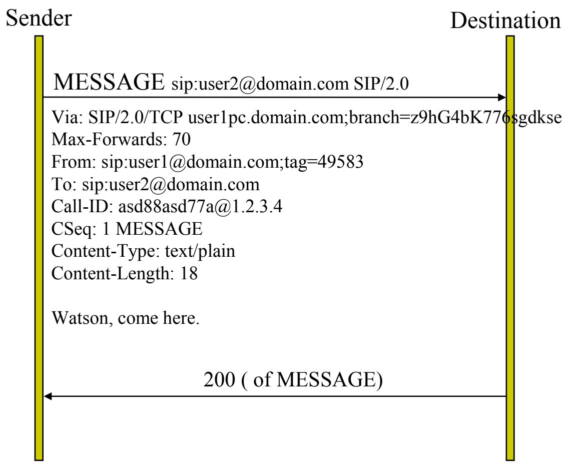


Generate immediate state response

Generate state change event

Use cases: automatic call-back, automatic buddy lists, message waiting indication,
User's own registration status and "IN triggering".

SIP MESSAGE provides Instant Messaging capability in Pager mode



Note:
-body is MIME or S/MIME
- there is no dialog!

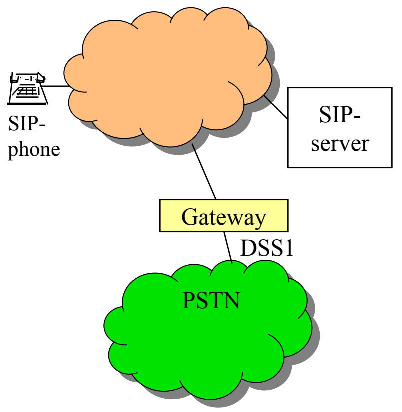
Use example:
-maps to SMS in GSM

Issue:
- authentication of sender and charging! Must use Proper security features

More SIP extensions

- MESSAGE
  - For instant messaging
- INFO
  - To transport mid-session information (very useful in SIP-PSTN gateways to carry all PSTN messages across SIP domains such that do not easily map to any other known SIP message)
- Automatic configuration
  - DHCP or Service Location Protocol (SLP)
- Caller Preferences
  - New headers: Accept-Contact, Reject-Contact, Request-Disposition (e.g. to express a preference for contacting the user at “fixed”, “business” connection).
- REFER
  - For session transfer (Refer-To: and Referred-By:)
- …

Deployment example: Elisa’s experimental service for BB customers



- SIP –server recognizes a numbering block, connects calls directly from IP-phone to IP-phone in the block
- Calls to all other numbers are routed to the gateway
- = SIP-server+Gateway are like a PBX