Session Initiation Protocol

SIP protocol and its extensions
SIP Service Architecture
SIP in 3G

A lot of this material
is based on proposals =>
may change quickly

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 3398: ISUP to SIP mapping
- 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 Release 5:
- 3GPP TS 24.228 v5.2.0 (2002-09) (v5.14 of late 2005) Signaling flows for the IP MM call control based on SIP and SDP; stage 3 (Release 5)
- 3GPP TS 24.229 v5.3.0 (2002-12) (v7.2.0 Feb 2006- rel 7) IP multimedia call control protocol based on SIP and SDP, Stage 3 (Release 5)
- 3GPP TS 29.228 v5.1.0 (2002-09) (v700 of 12-2005 for Rel 7) IMS Cx and Dx interfaces, Signaling flows and message contents; (Release 5)
- Etc…

Some SIP WG Documents

Internet-Drafts (total of 13 on Feb 12th 2006) E.g.:
Guidelines for Authors of Extensions to the Session Initiation Protocol (SIP) (58196 bytes)
End-to-middle Security in the Session Initiation Protocol (SIP) (59868 bytes)

Request For Comments (total of 40 on Feb 12th 2006), E.g.:
MIME media types for ISUP and QSIG Objects (RFC 3204) (19712 bytes) updated by RFC 3459
SIP-Specific Event Notification (RFC 3265) (89005 bytes) obsoletes RFC 2543
SIP: Locating SIP Servers (RFC 3263) (42310 bytes) obsoletes RFC 2543
Reliability of Provisional Responses in SIP (RFC 3262) (29643 bytes) obsoletes RFC 2543
SIP: Session Initiation Protocol (RFC 3261) (647976 bytes) obsoletes RFC 2543/ updated by RFC 3853,RFC 4320
The Session Initiation Protocol UPDATE Method (RFC 3311) (28125 bytes)
Integration of Resource Management and SIP (RFC 3312) (65757 bytes) updated by RFC 4032
A Privacy Mechanism for the Session Initiation Protocol (SIP) (RFC 3323) (54116 bytes)
Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks (RFC 3325) (36170 bytes)
Session Initiation Protocol Extension for Instant Messaging (RFC 3428) (41475 bytes)
Security Mechanism Agreement for the Session Initiation Protocol (SIP) Sessions (RFC 3329) (51503 bytes)
Private Session Initiation Protocol (SIP)Extensions for Media Authorization (RFC 3313) (36866 bytes)
Compressing the Session Initiation Protocol (RFC 3486) (24181 bytes)
Caller Preferences for the Session Initiation Protocol (SIP) (RFC 3841) (61382 bytes)
Update to the Session Initiation Protocol (SIP) Preconditions Framework (RFC 4032) (20492 bytes) updates RFC 3312
Session Timers in the Session Initiation Protocol (SIP) (RFC 4028) (65363 bytes)
The Stream Control Transmission Protocol (SCTP) as a Transport for the Session Initiation Protocol (SIP) (RFC 4168) (21079 bytes)

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

Some SIMPLE and XCON WG Documents

SIMPLE Internet-Drafts (Feb 12th 2006 – total of 18 drafts) :
The Message Session Relay Protocol (134595 bytes)
The Extensible Markup Language (XML) Configuration Access Protocol (XCAP) (155849 bytes)
A Data Model for Presence (88211 bytes)

SIMPLE Request For Comments:
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)
Indication of Message Composition for Instant Messaging (RFC 3994) (27472 bytes)

Source: http://www.ietf.org/html.charters/simple-charter.html
NB: Charter has last actions recorded in 2004.

Internet-Drafts XCON WG (Feb 12th, 2006, last action Dec 2005):
Conferencing Scenarios (38834 bytes)
Requirements for Floor Control Protocol (32915 bytes)
The Binary Floor Control Protocol (BFCP) (148929 bytes)
A Framework and Data Model for Centralized Conferencing (131914 bytes)
Connection Establishment in the Binary Floor Control Protocol (BFCP) (30420 bytes)
No Request For Comments
http://www.ietf.org/html.charters/xcon-charter.html

SIPPING WG documents

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

Internet-Drafts(Feb 12th 2006: total of 30) E.g.:
Best Current Practices for NAT Traversal for SIP (111275 bytes)
A Session Initiation Protocol (SIP) Event Package for Conference State (98180 bytes)
Interworking between SIP and QSIG (156122 bytes)
Session Initiation Protocol Call Control - Conferencing for User Agents (98448 bytes)
A Framework for Conferencing with the Session Initiation Protocol (68479 bytes)
Emergency Services URI for the Session Initiation Protocol (23397 bytes)

Request For Comments(Feb 12th 2006 – total of 20) E.g.:
Session Initiation Protocol (SIP) for Telephones (SIP-T): Context and Architectures (RFC 3372) (49893 bytes)
Short Term Requirements for Network Asserted Identity (RFC 3324) (21964 bytes)
Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping (RFC 3398) (166207 bytes)
The Session Initiation Protocol (SIP) and Session Description Protocol (SDP) Static Dictionary for Signaling Compression (SigComp) (RFC 3485) (80195 bytes)
Mapping of of Integrated Services Digital Network (ISUP) Overlap Signalling to the Session Initiation Protocol (SIP) (RFC 3578)
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)
Best Current Practices for Third Party Call Control in the Session Initiation Protocol (RFC 3725) (77308 bytes)
Using E.164 numbers with the Session Initiation Protocol (SIP) (RFC 3824) (36535 bytes)
A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP) (RFC 3842) (36877 bytes)

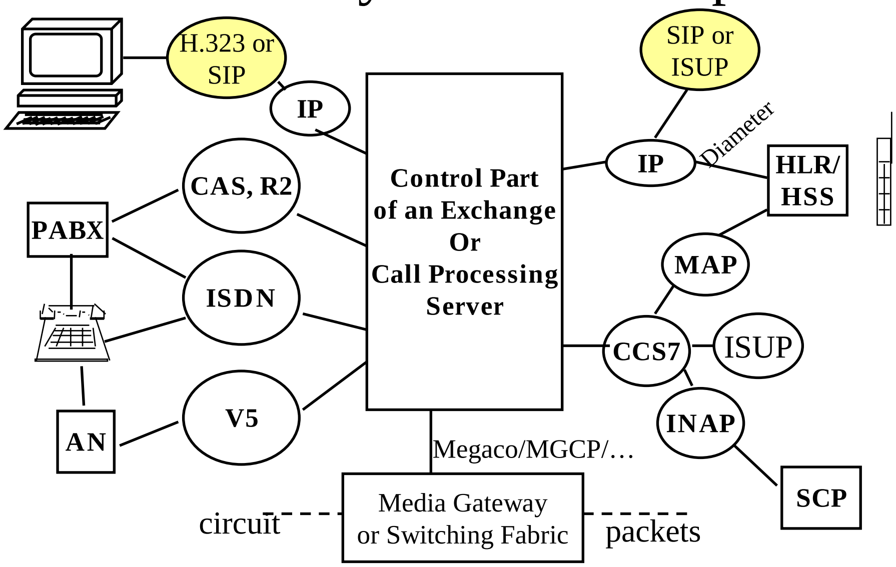
Some mmusic Documents

Internet-Drafts (12th Feb 2006: total of 18)E.g.:
SDP: Session Description Protocol (108592 bytes)
SDPng Transition (40981 bytes)
Real Time Streaming Protocol 2.0 (RTSP) (409743 bytes)
Session Description Protocol Security Descriptions for Media Streams (99284 bytes)
Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols (226395 bytes)
Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams (25758 bytes)
Security Preconditions for Session Description Protocol Media Streams (32863 bytes)
Connectivity Preconditions for Session Description Protocol Media Streams (37979 bytes)
The SDP (Session Description Protocol) Content Attribute (20988 bytes)

Request For Comments (Feb 12th 2006 – total of 15) E.g.:
Real Time Streaming Protocol (RTSP) (RFC 2326) (195011 bytes)
SDP: Session Description Protocol (RFC 2327) (87096 bytes) updated by RFC 3266
Session Announcement Protocol (RFC 2974) (40129 bytes)
An Offer/Answer Model with SDP (RFC 3264) (60854 bytes) obsoletes RFC 2543
Session Description Protocol (SDP) Offer/Answer Examples (RFC 4317) (32262 bytes)

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

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
  - Ascii based - simple tools for development but long messages
  - Lower call setup time than in H.323 (?)
  - 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
- A lot of development during the last 3…5 years!
- Market view: there are 3 important variants of SIP: 3G, IETF and Microsoft Messanger.

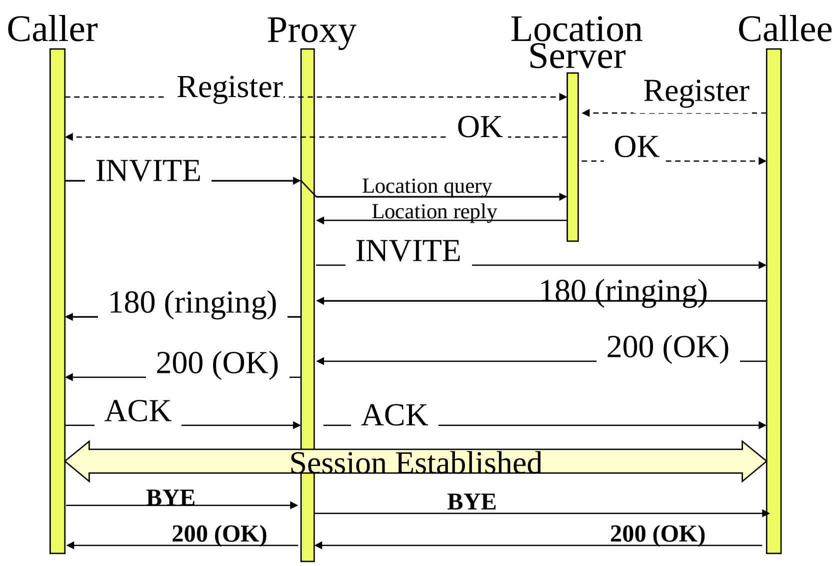
SigComp allows cmpression of Signaling Messages

- 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:
  - 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



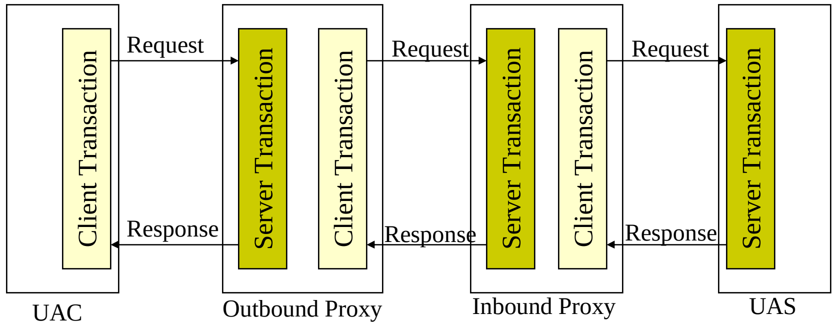
”Basic call” Example (IETF)

- 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

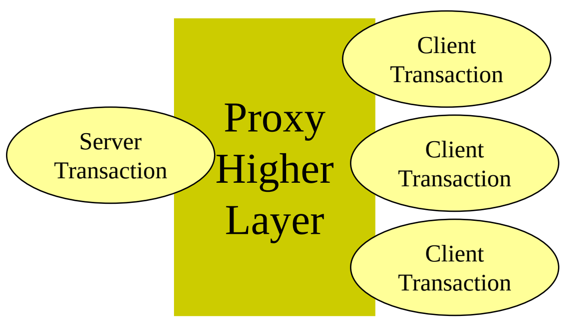
- Headers carry control information and are processed e.g. by Proxies
- Body can be e.g. SDP – session description protocol
  - end-to-end information (cmp H.245) describing session requirements e.g. coding methods, etc
- Message delivery is transaction oriented= have request + reply: e.g INVITE+200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

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



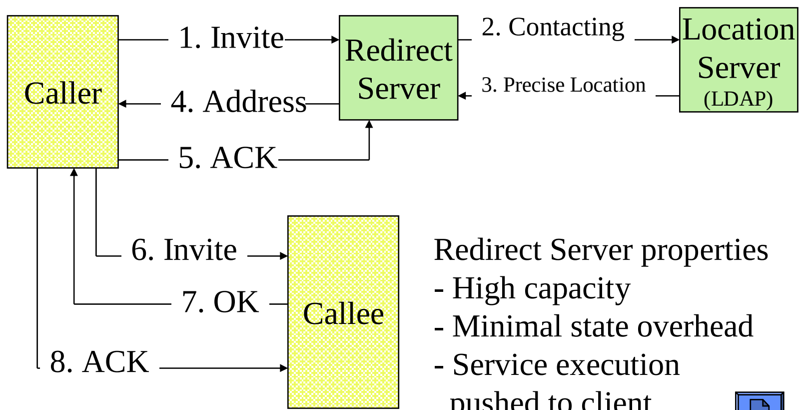
- All Communication follows this transaction model, except 2xx and ACK.
- Server transactions filter incoming requests, absorbing retransmissions
- Only UAS can generate 2xx and only UAC can generate ACK (to success).

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

Full 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 current(2004) extensions

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 calcels (see later for SIP headers).

Notes on SIP methods: REGISTER

- Purpose: to register the user's current location.
- 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

- 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

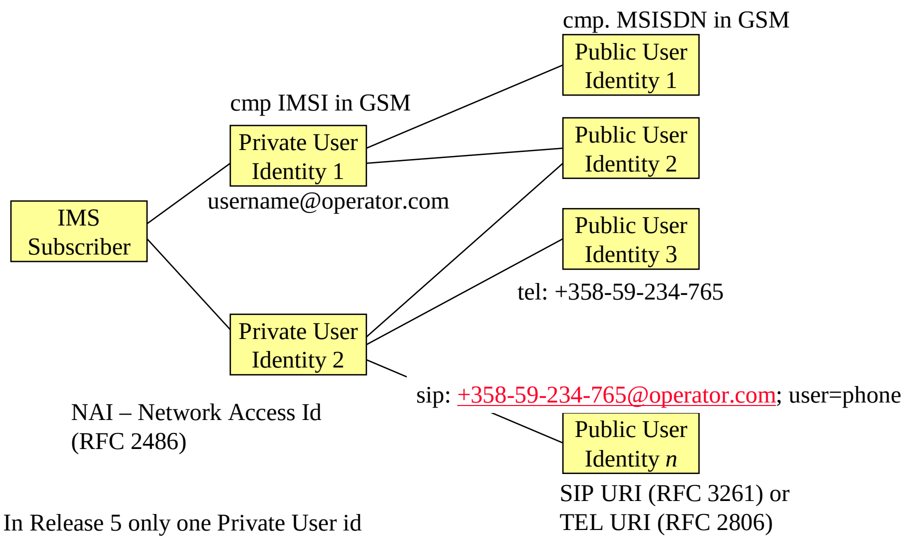
KeepAlive = re-INVITE mechanism

- Text based coding increases the signaling overhead → problem in Radio access

Identification of users

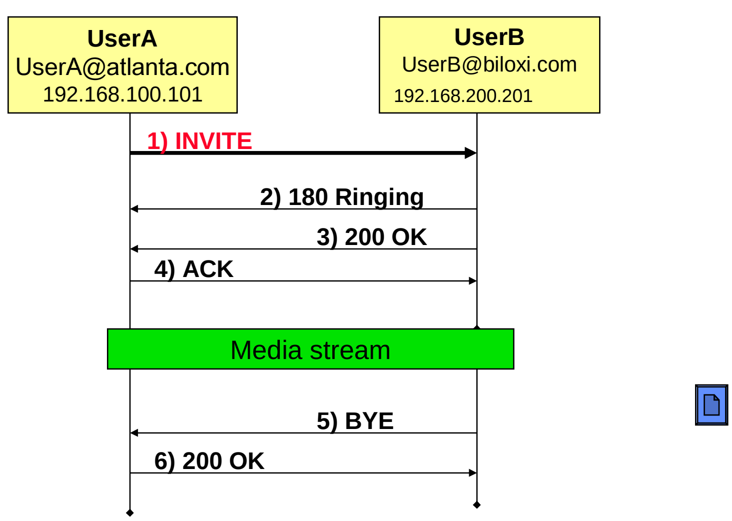
- **sip:user@host[parameters][headers]**
- SIP URIs are like URLs, with prefix sip: which gives schema
  - sip:joe.smith@hut.fi
  - sip:joe.smith@hut.fi?subject=Protocol
  - sip:sales@hotel.xy;geo.position:=48.54_-123.84_120
- Address must include host, other parameters are optional (username, port, 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



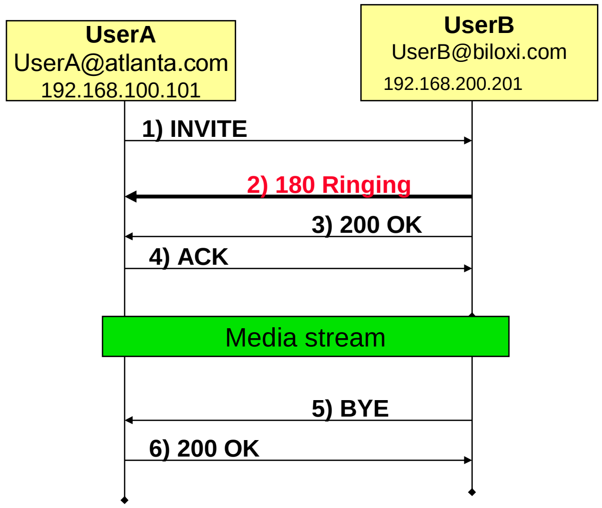
Raimo Kantola –S- 2006 Signaling Protocols 13 - 27

”Basic Call” call flow



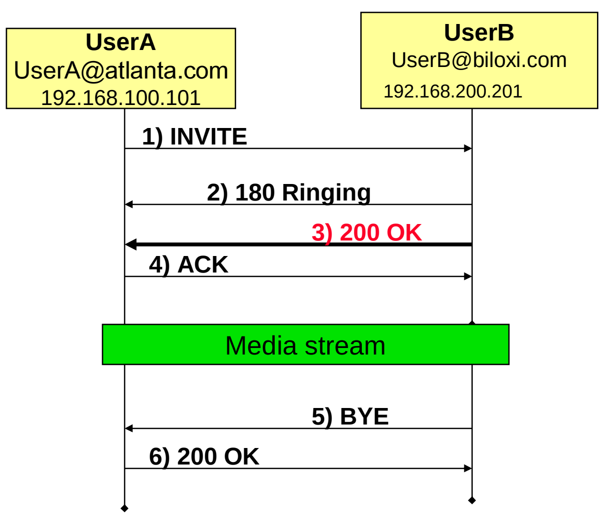
Raimo Kantola –S- 2006 Signaling Protocols 13 - 28

”Basic Call” call flow



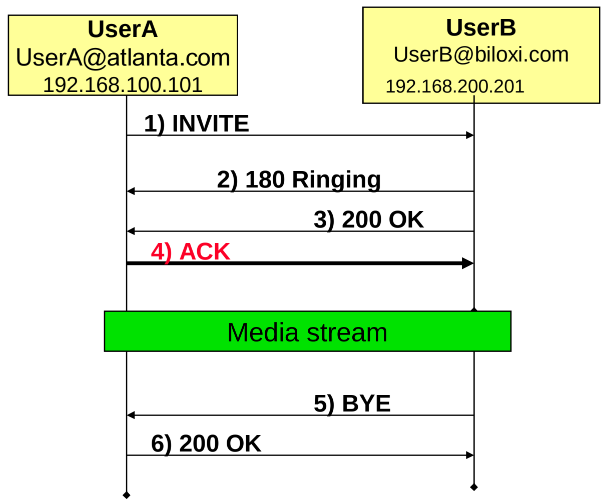
Raimo Kantola –S- 2006 Signaling Protocols 13 - 29

”Basic Call” call flow



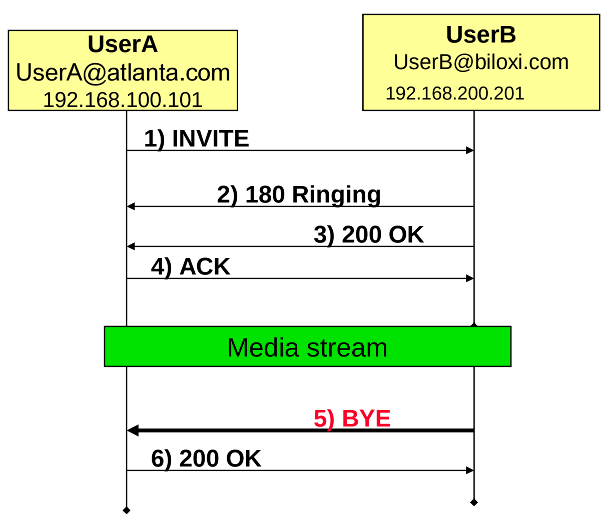
Raimo Kantola –S- 2006 Signaling Protocols 13 - 30

”Basic Call” call flow



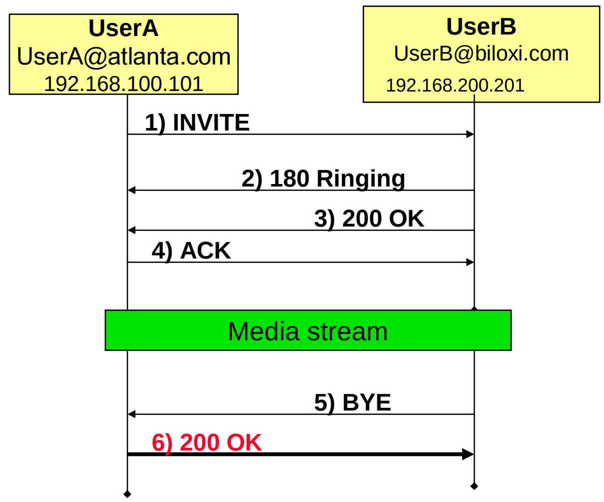
Raimo Kantola –S- 2006 Signaling Protocols 13 - 31

”Basic Call” call flow



Raimo Kantola –S- 2006 Signaling Protocols 13 - 32

”Basic Call” call flow



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.

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 body**
  - 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

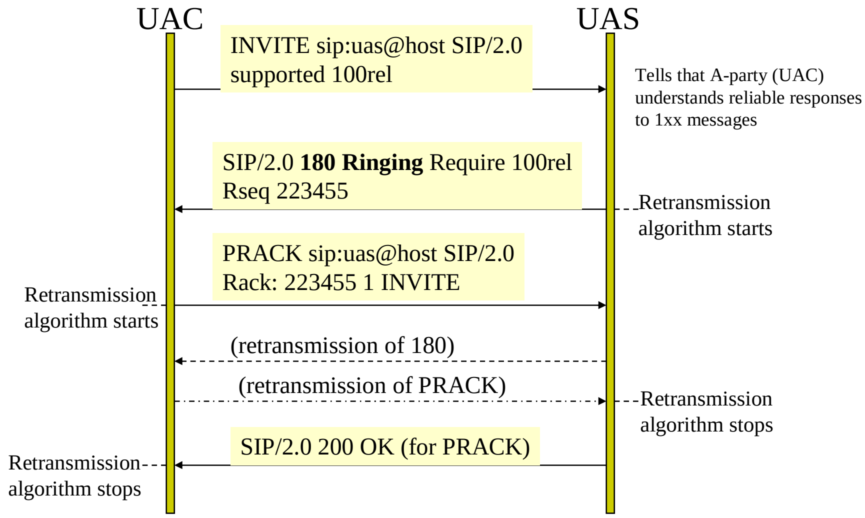
SIP Extensions

- 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 request and response
  - **Supported** UAC and UAS tell features they support
- *Required* features can be specified in request and response
  - **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

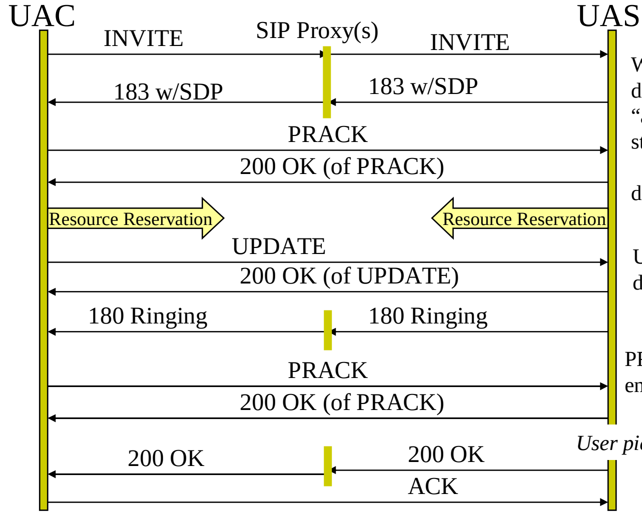
Reliable Provisional response in SIP



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 precon-ditions are OK at the originator

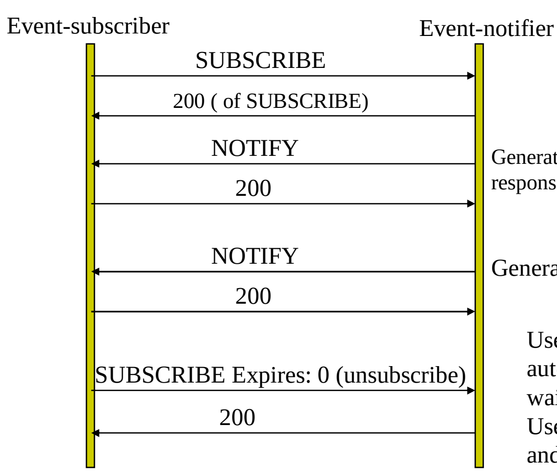
PRACK method is used to ensure delivery of 183 and 180

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

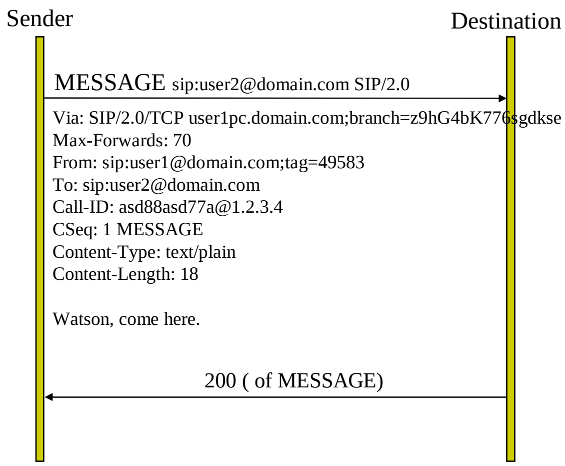


Generate immediate state response

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!

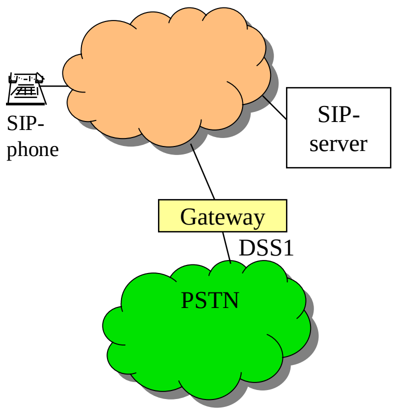
Use example:
-maps to SMS in GSM

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:)
- …

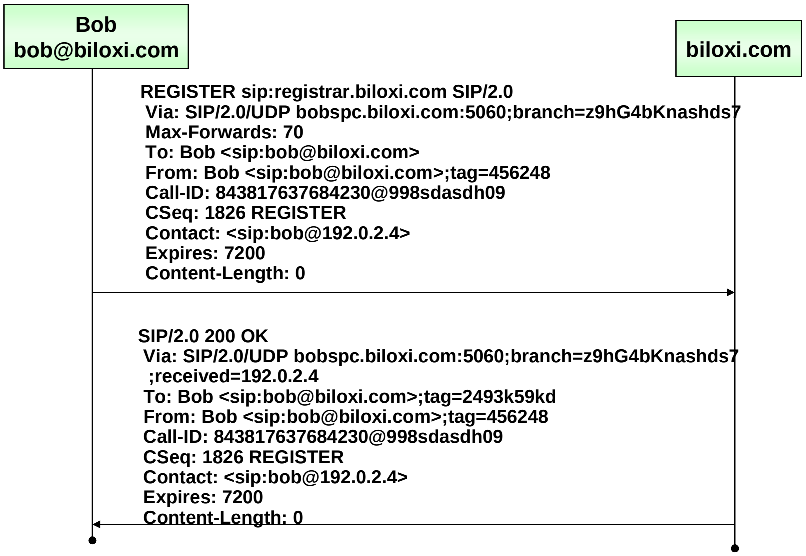
Deployment example: Elisa's experimental service for BB customers



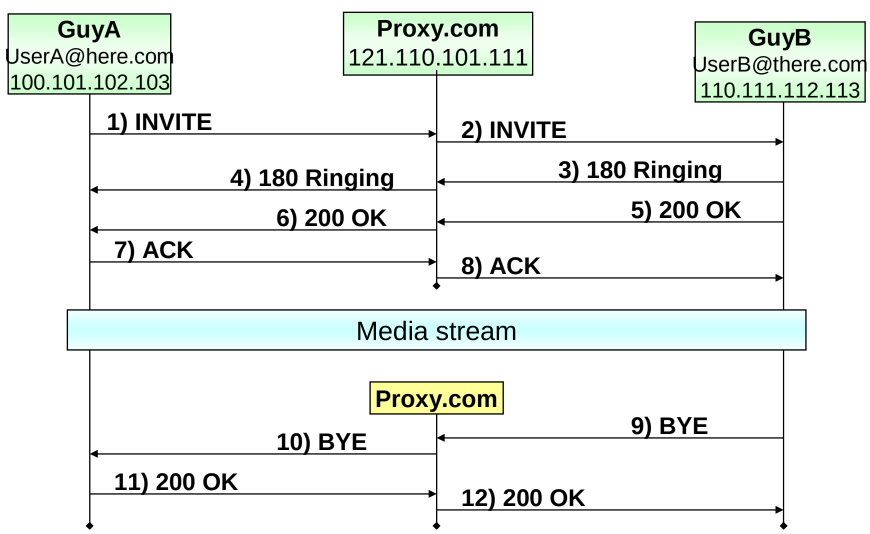
- SIP –server recognizes a numbering block, connects calls directly from IP-phone to IP-phone in the block
- Calls to all other numbers are routed to the gateway
- = SIP-server+Gateway are like a PBX

Call Setup Examples based on Generic SIP

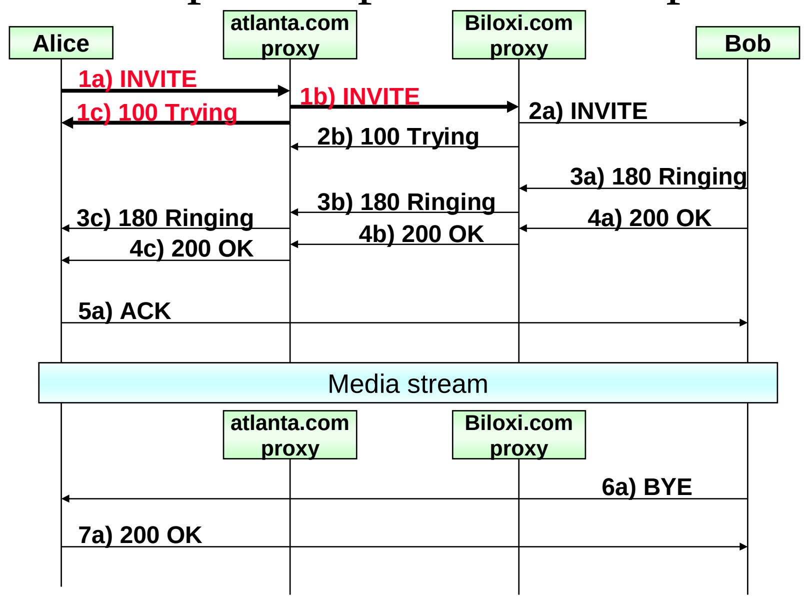
Registration example with SIP



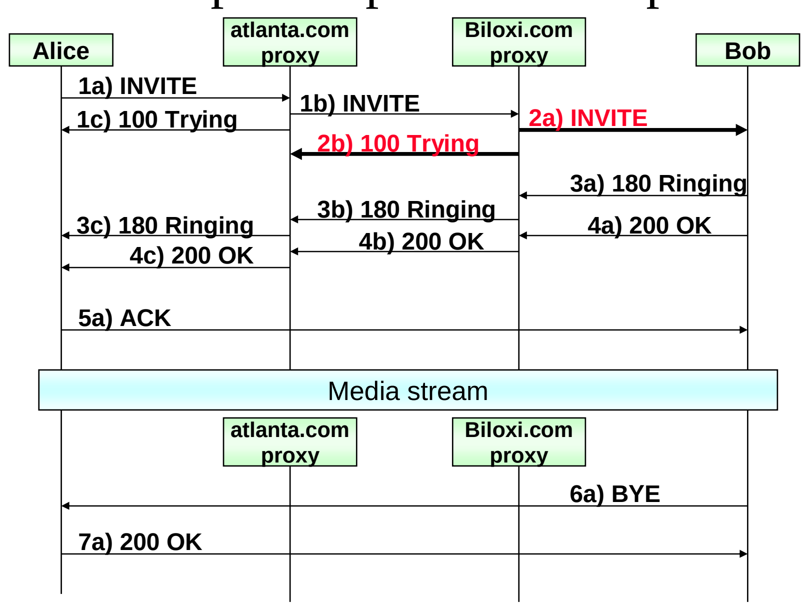
Call Setup example with one proxy



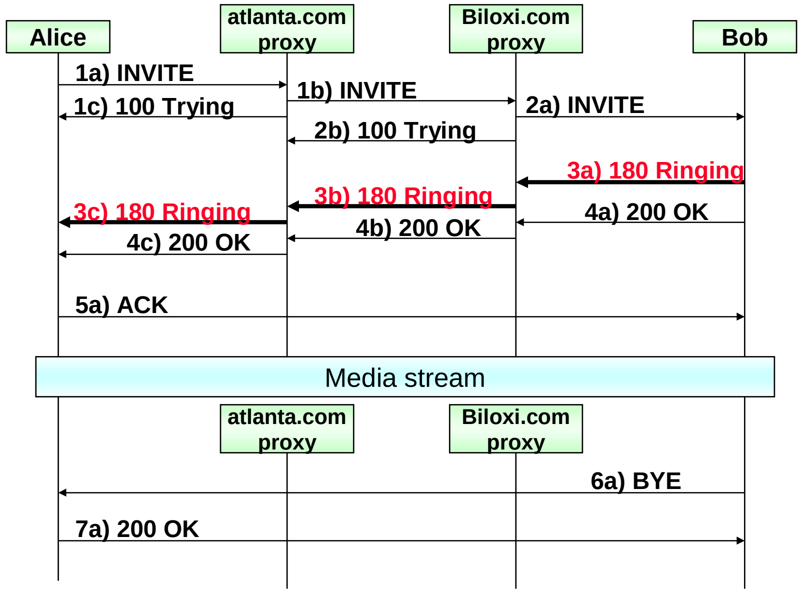
Call Setup example with two proxies



Call Setup example with two proxies

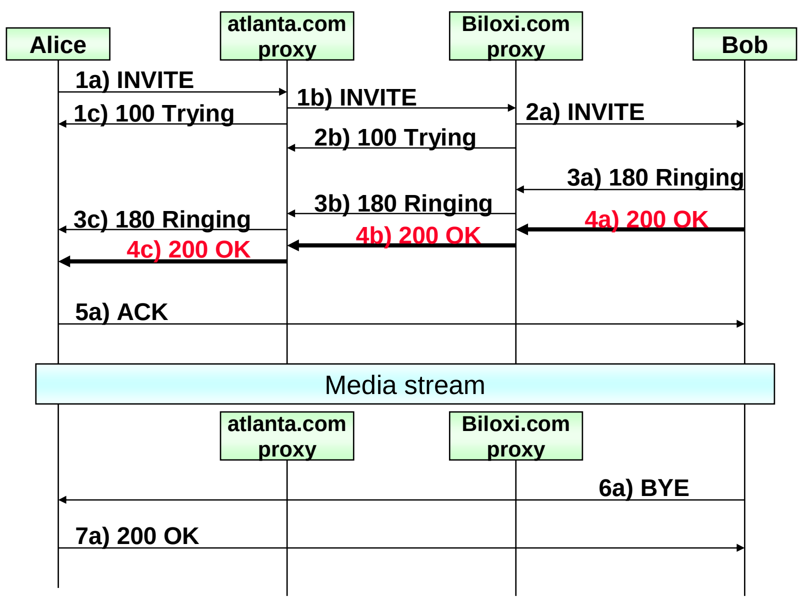


Call Setup example with two proxies



Raimo Kantola –S- 2006 Signaling Protocols 13 - 57

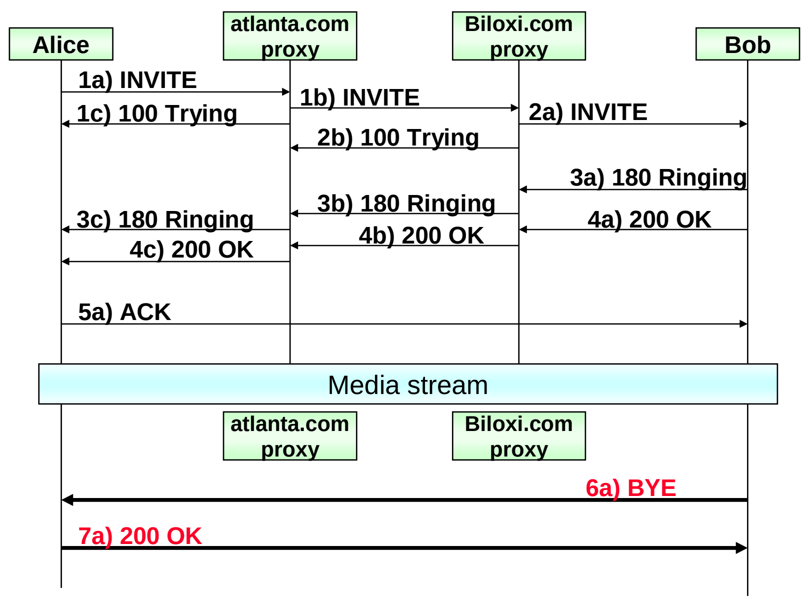
Call Setup example with two proxies



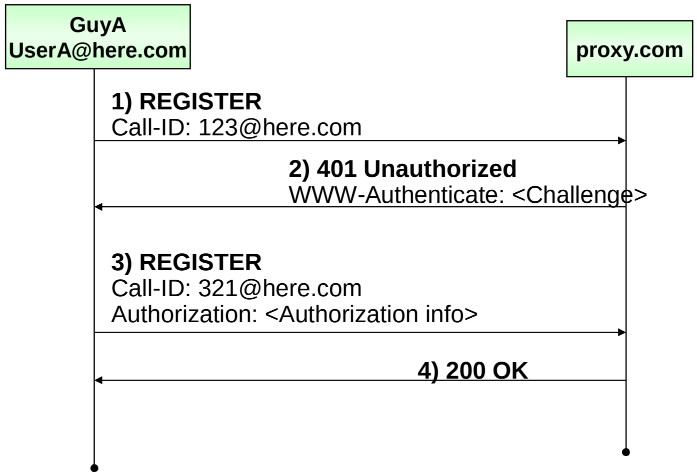
Raimo Kantola –S- 2006 Signaling Protocols 13 - 58

Call Setup example with two proxies

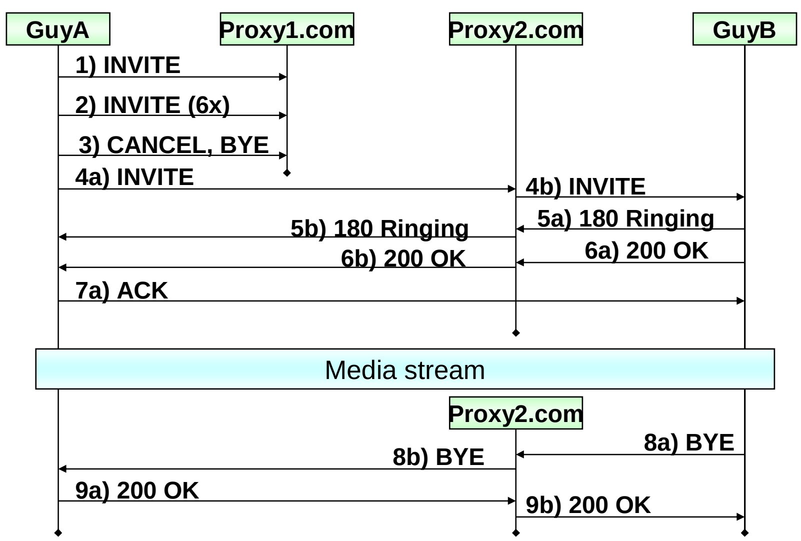
Call Setup example with two proxies



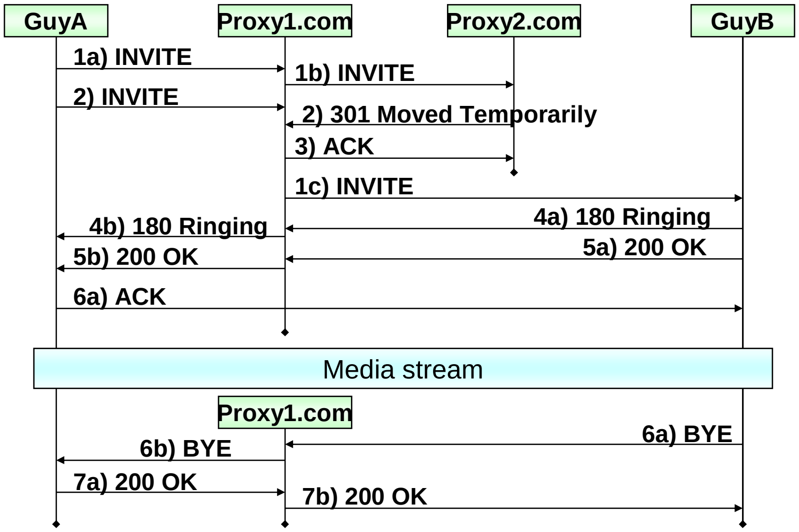
Registration example with SIP authentication



Call Setup example with a non-working proxy



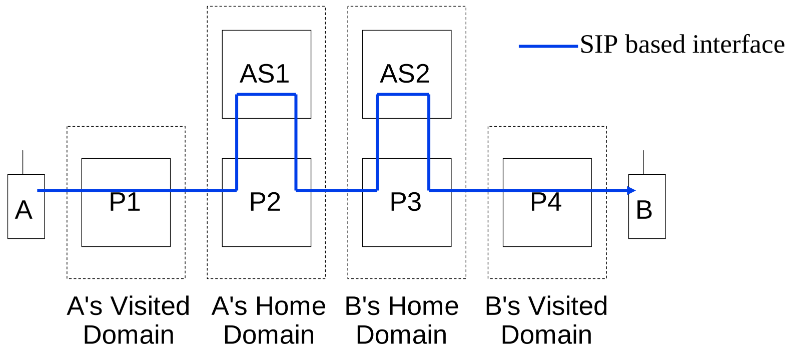
Call Setup example with a Redirect server



Services use many protocols

- New services and more flexible service creation should differentiate IP Communications Network from PSTN
- Services should combine different forms of communication, thus multiple protocols are needed:
  - SIP for media sessions and session related services, subscriptions and notifications?, messaging?
  - HTTP for web and transactions
  - SMTP for e-mail
  - RTSP for media streaming
- The use of these protocols is orchestrated by the service logic: context is set up using SIP.

Routing and Service Model in 3G



P1, P4: Outbound Proxies

P2, P3: Registrar Proxies

AS1, AS2: Application Servers

NB: Also AS based on direct processing of call state: There is no Basic call state model like in IN

SIP Entities & Service Capabilities

- User Agent (= UAC + UAS)
  - Can run services, such as forwarding, filtering etc.
  - Not always connected (out of coverage/battery etc.)
- Redirect Server
  - Can do services that require only Request-URI change, e.g. translation, parameter addition etc.
- Proxy Server
  - Can change certain headers and stay in the signaling path
  - Forking, actions based on responses
- Back-to-Back User Agent (=both ways User Agent)
  - Can e.g. issue requests to a call leg or modify SDP, generate ACK and 200OK, like UAC/UAS
  - In many cases necessary

Application Server in 3G

- Fuzzy Definition but has SIP+ interface!
- Can be a Redirect or Proxy Server or Back-to-Back UA
- The key is that it should be *programmable*
  - Routing based on service logic: what to do when user not registered or busy
  - URI translation: Reachability chains
  - Interfaces to other protocols: HTTP, SMTP, RTSP etc.
- Can be single purpose boxes, or multi-purpose boxes, or controllers who orchestrate things

3GPP Network Model (preliminary: …