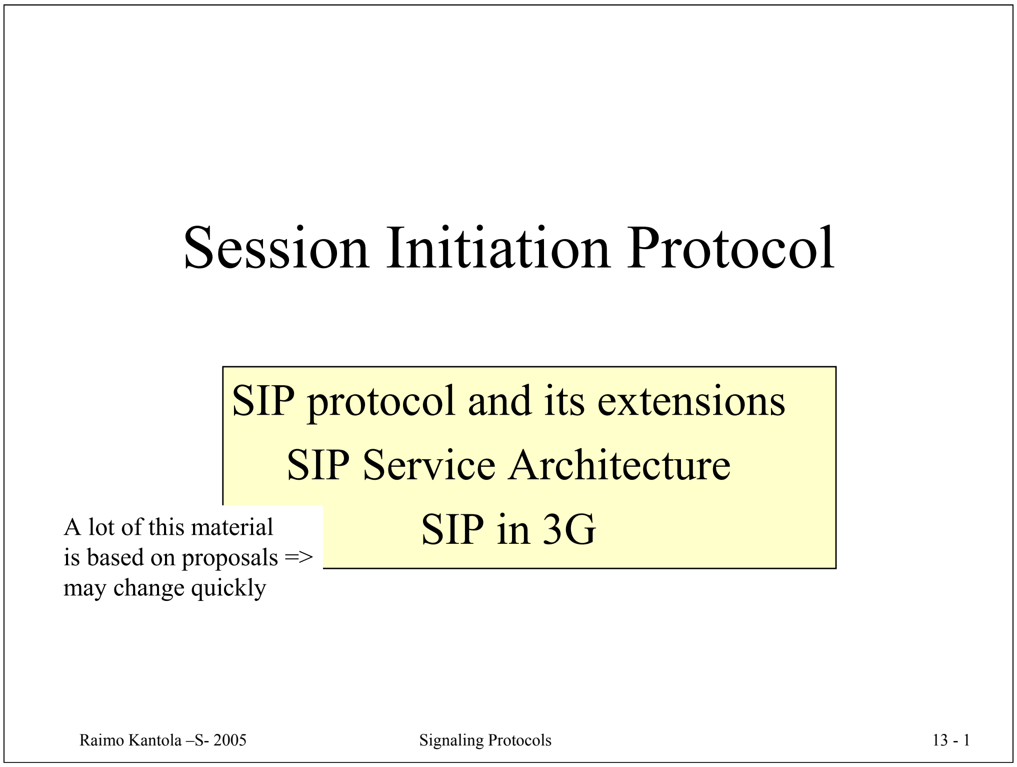


Session Initiation Protocol

SIP protocol and its extensions

SIP Service Architecture

SIP in 3G

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

Sources

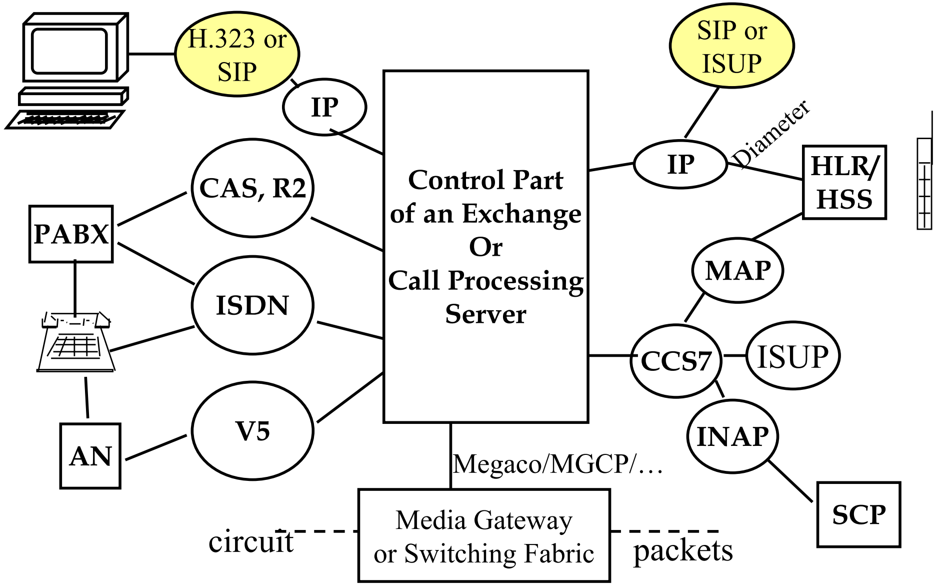
IETF:

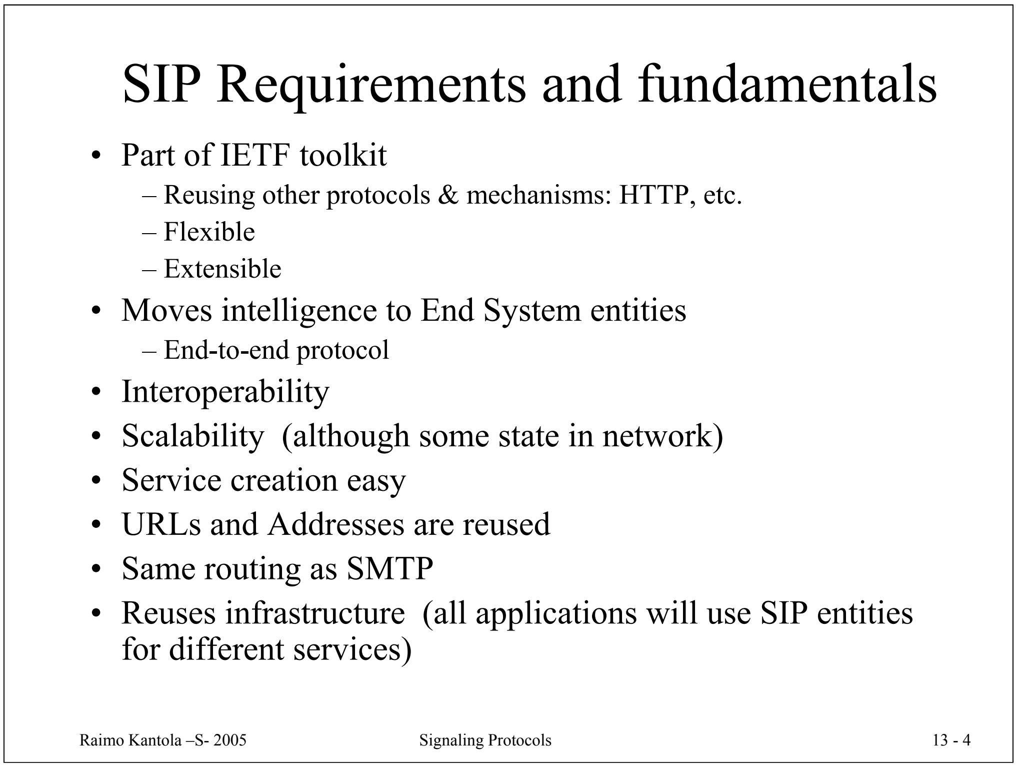
RFC 3261: SIP: Session Initiation Protocol
RFC 3262: Reliability of Provisional Responses in SIP (PRACK)
RFC 3265: SIP Specific Event Notification
RFC 3311: SIP UPDATE method
RFC 3398: ISUP to SIP mapping
RFC 3428: SIP Extension for Instant Messaging
RFC 2327: SDP: Session Description Protocol
RFC 3264: An Offer/Answer Model with Session Description Protocol (SDP)

3G Release 5:

3GPP TS 24.228 v5.2.0 (2002-09) 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) IP multimedia call control protocol based on SIP and SDP, Stage 3 (Release 5)
3GPP TS 29.228 v5.1.0 (2002-09) IMS Cx and Dx interfaces, Signaling flows and message contents; (Release 5)
Etc…

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, proposed also as network to network signaling
- A lot of development during the last 3…4 years!

SIP Implementation Status of 11/2003

There are several single-operator single-vendor islands offering SIP services, but

The inter-operator or multi-vendor solutions are still very rare, and need to be designed case by case.

This despite the Interoperability Bakeoffs organised by vendors!

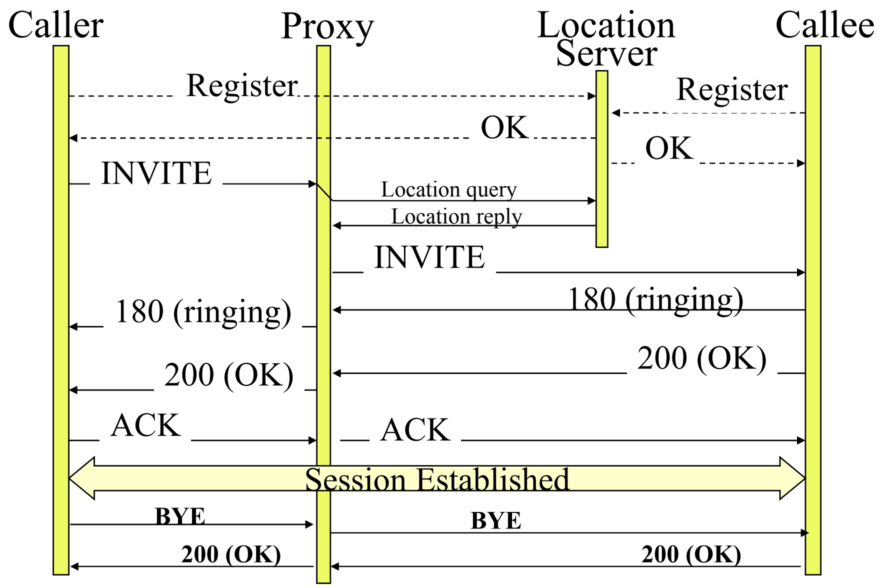
SigComp allows cmpression of Signaling Messages

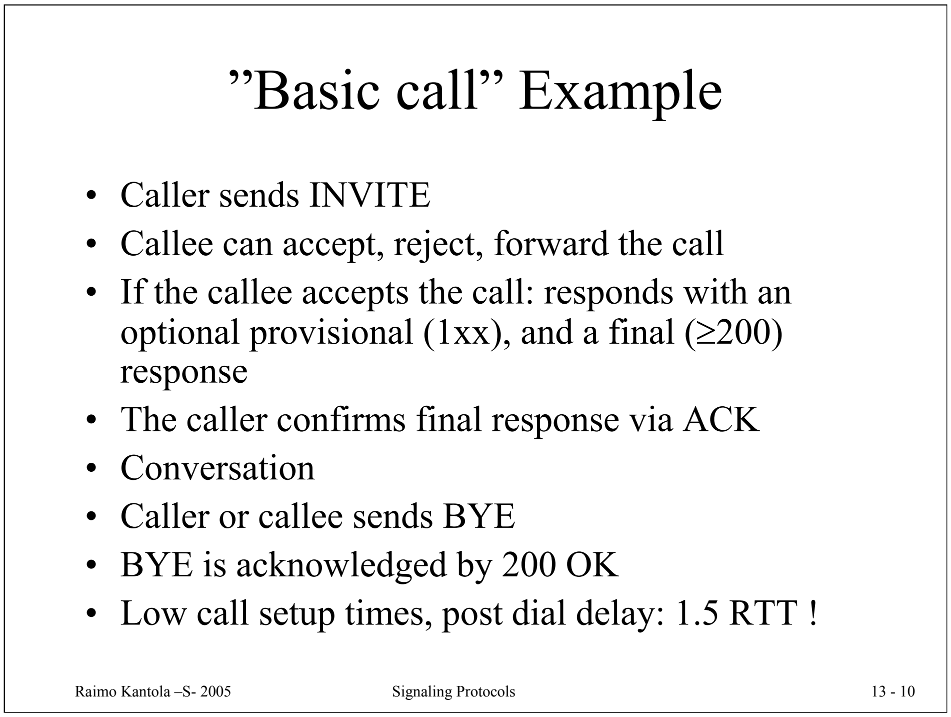
- RFC 3320 and RFC 3321 specify a layer between the signaling transport and the signaling application
- Uses Global and User Specific Dictionaries to store state data over many SIP sessions
- Overall Compression/decompression architecture is based on a bytecode driven Universal Decompression Virtual Machine
  - Bytecode can be sent in SigComp messages by the Compressior
  - 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 <u>not defined by SIP specs</u> (e.g. LDAP)

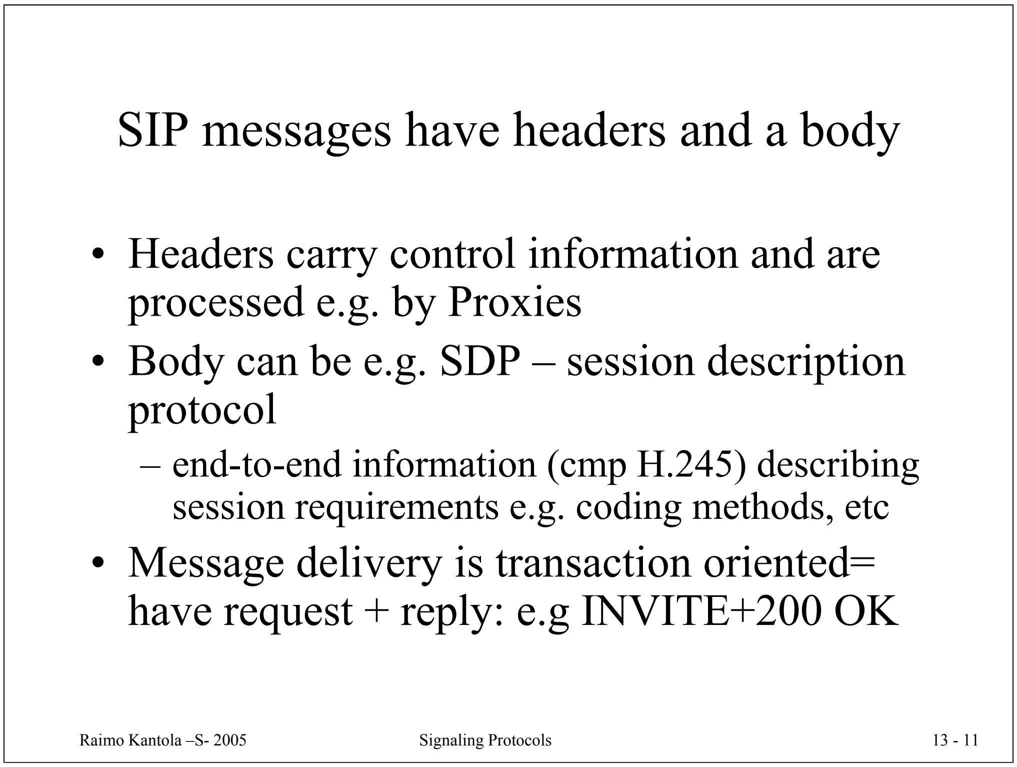
Basic SIP call setup and release





”Basic call” Example

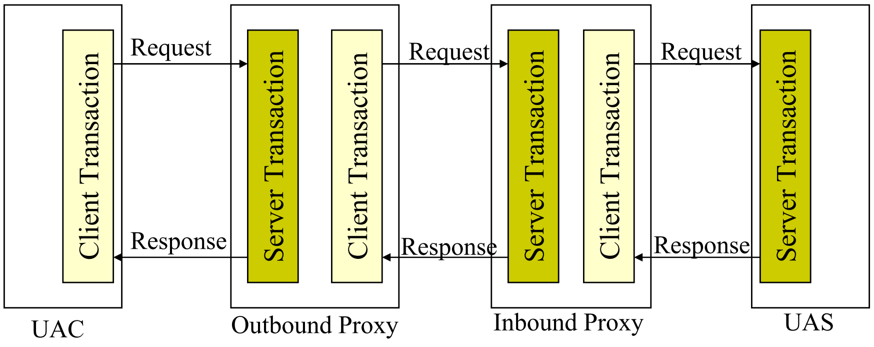
- Caller sends INVITE
- Callee can accept, reject, forward the call
- If the callee accepts the call: responds with an optional provisional (1xx), and a final (≥200) response
- The caller confirms final response via ACK
- Conversation
- Caller or callee sends BYE
- BYE is acknowledged by 200 OK
- Low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



SIP messages have headers and a body

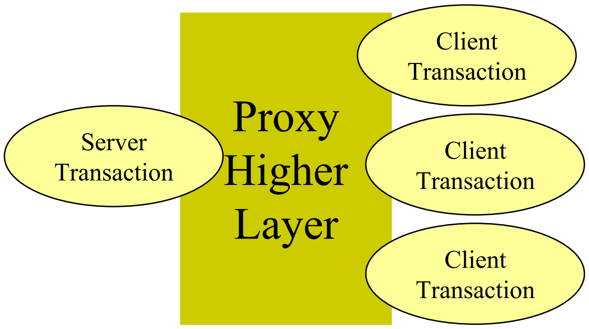
- Headers carry control information and are processed e.g. by Proxies
- Body can be e.g. SDP – session description protocol
  - end-to-end information (cmp H.245) describing session requirements e.g. coding methods, etc
- Message delivery is transaction oriented= have request + reply: e.g INVITE+200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

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



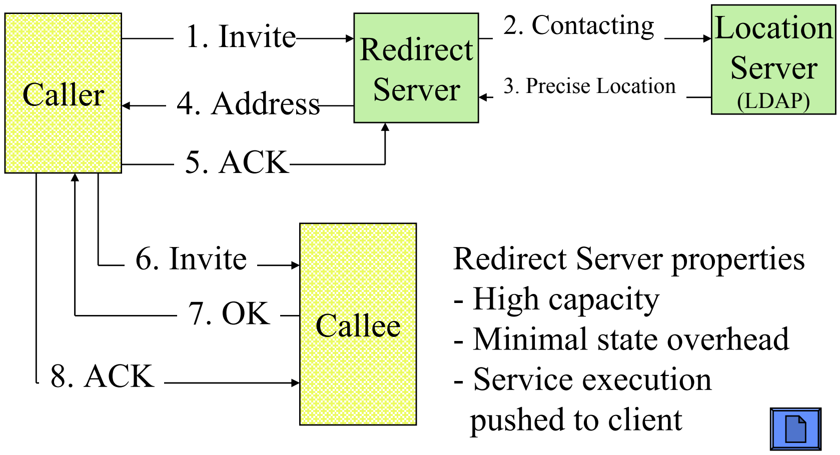
- All Communication follows this transaction model, except 2xx and ACK.
- Server transactions filter incoming requests, absorbing retransmissions
- Only UAS can generate 2xx and only UAC can generate ACK (to success).

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

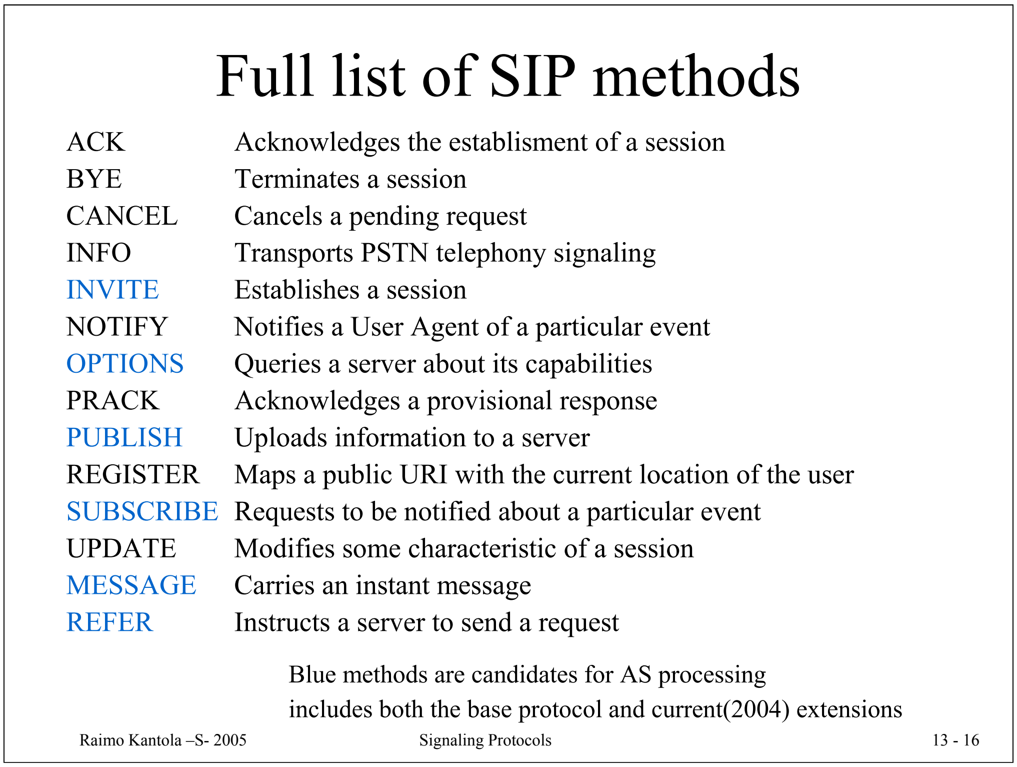
17.3.04

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

- 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
- Also semi-stateful is possible

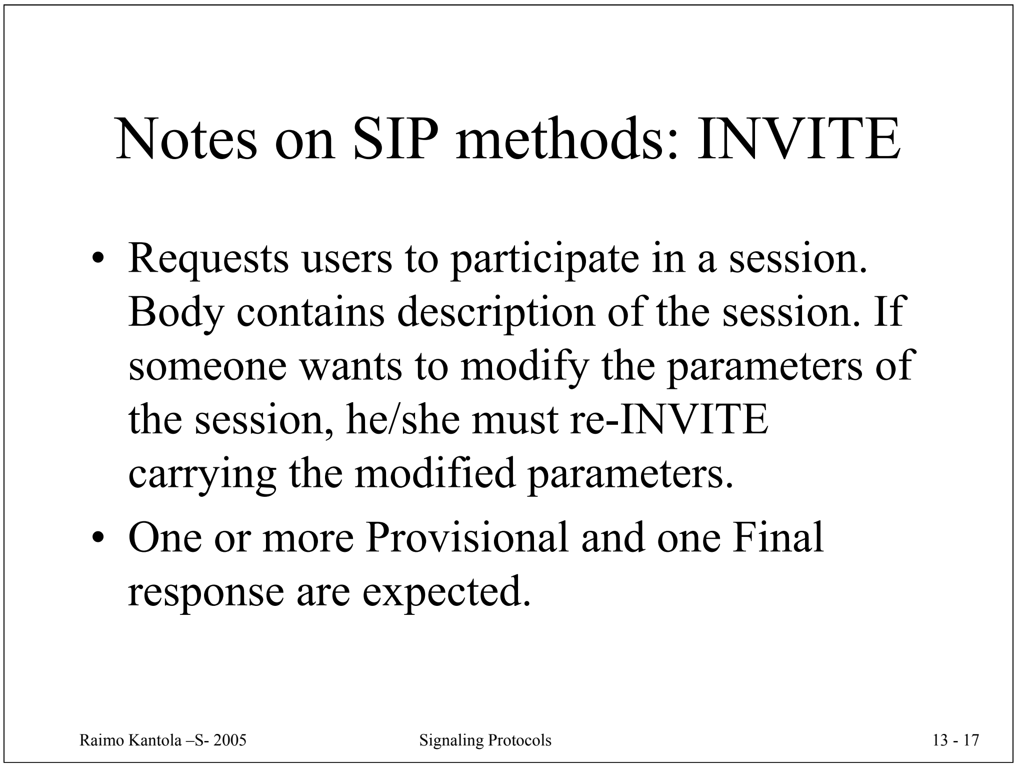
UA = User Agent, UAC = UA Client
UAS = UA Server



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

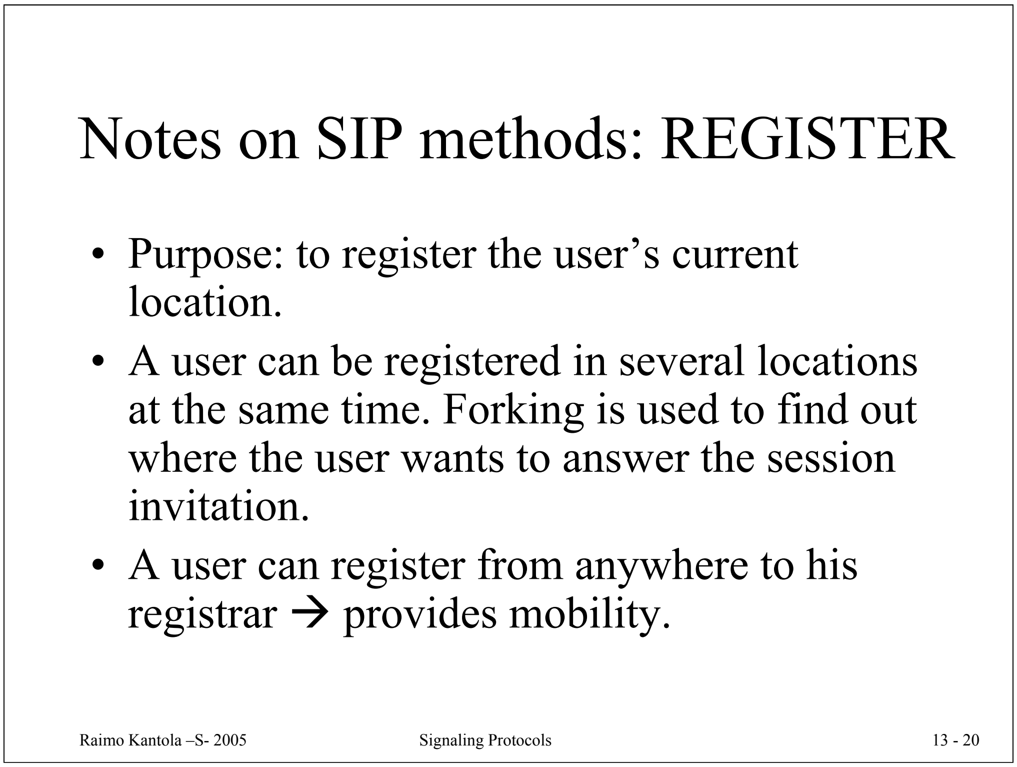
- Requests users to participate in a session. Body contains description of the session. If someone wants to modify the parameters of the session, he/she must re-INVITE carrying the modified parameters.
- One or more Provisional and one Final response are expected.

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

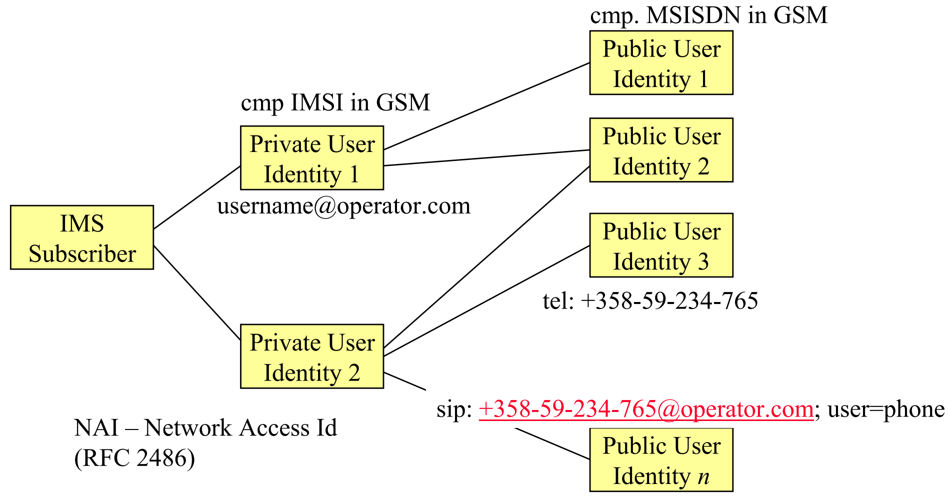
KeepAlive =
re-INVITE mechanism

- Ascii coding increases the signaling overhead 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



cmp. MSISDN in GSM

Public User Identity 1

cmp IMSI in GSM

Private User Identity 1

Public User Identity 2

username@operator.com

IMS Subscriber

Public User Identity 3

tel: +358-59-234-765

Private User Identity 2

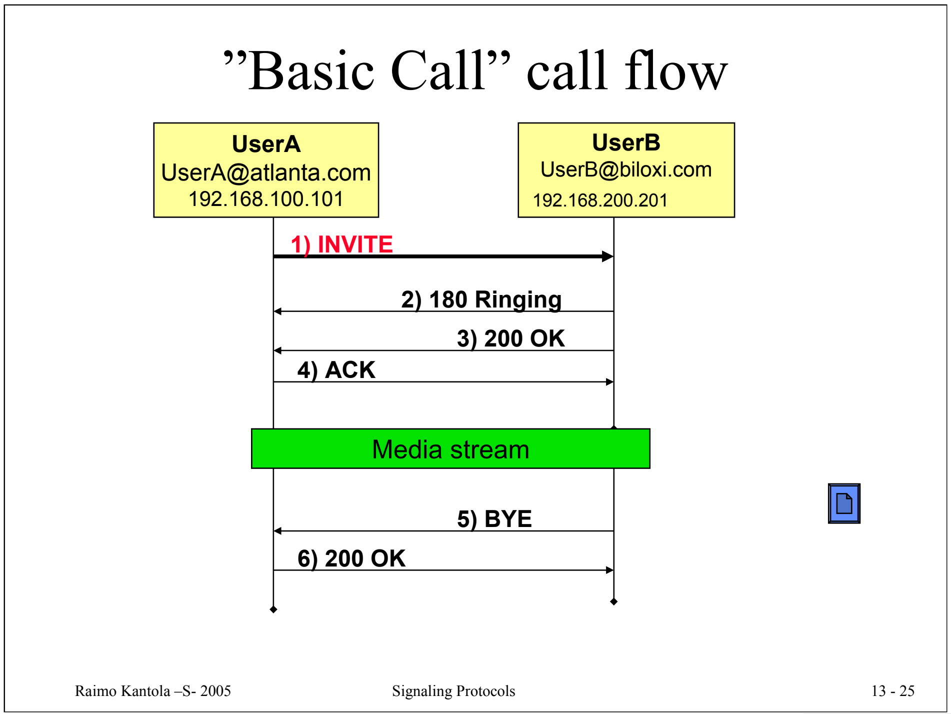
sip: +358-59-234-765@operator.com; user=phone

NAI – Network Access Id (RFC 2486)

Public User Identity *n*

SIP URI (RFC 3261) or
TEL URI (RFC 2806)

In Release 5 only one Private User id



”Basic Call” call flow

UserA
UserA@atlanta.com
192.168.100.101

UserB
UserB@biloxi.com
192.168.200.201

1) INVITE

2) 180 Ringing

3) 200 OK

4) ACK

Media stream

5) BYE

6) 200 OK

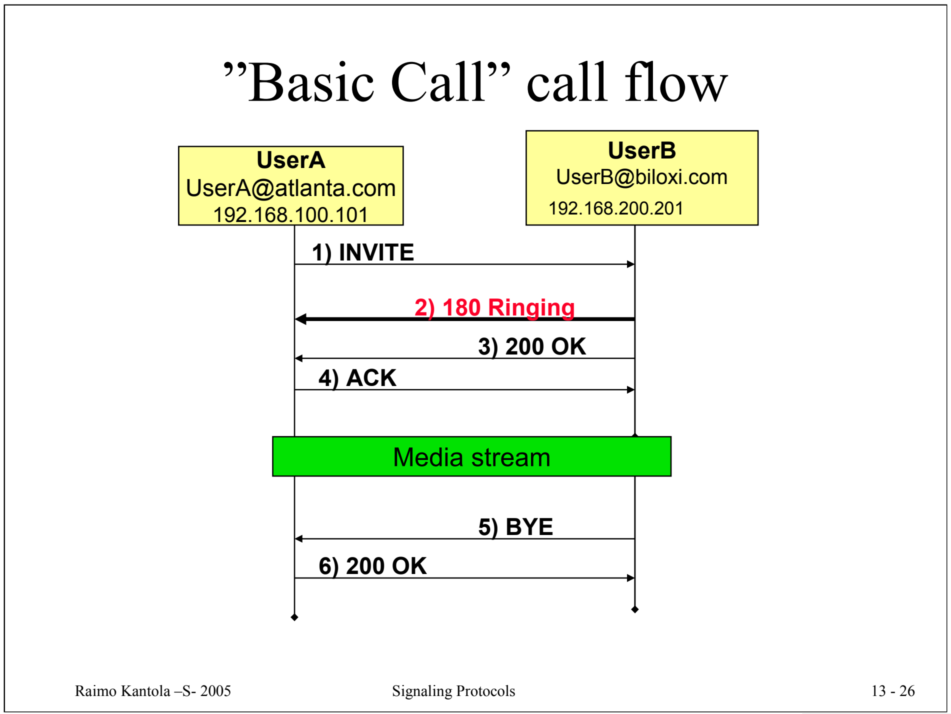
Raimo Kantola –S- 2005 Signaling Protocols 13 - 25

```
    INVITE sip:UserB@biloxi.com SIP/2.0
    Via: SIP/2.0/UDP
client.atlanta.com:5060;branch=z9hG4bK74bf9
    Max-Forwards: 70
    From: BigGuy <sip:UserA@atlanta.com>;tag=9fxced76sl
    To: LittleGuy <sip:UserB@biloxi.com>
    Call-ID: 3848276298220188511@atlanta.com
    CSeq: 1 INVITE
    Contact: <sip:UserA@192.168.100.101>
    Content-Type: application/sdp
    Content-Length: 147

    v=0
    o=UserA 2890844526 2890844526 IN IP4 client.atlanta.com
    s=-
    c=IN IP4 192.168.100.101
    t=0 0
    m=audio 49172 RTP/AVP 0
    a=rtpmap:0 PCMU/8000
```

In this example, UserA sends INVITE to UserB. In the initial INVITE UserA puts Max-Forwards: 70. In Contact: header UserA puts his address where he can be reached in the moment.

In SDP body UserA specifies what kind of session he wants to establish. In this case it would be audio session on port 49172 (RTP over UDP as transport) with PCM µ coding with 8000 samples per second.

Example from: http://www.ietf.org/internet-drafts/draft-ietf-sipping-call-flows-01.txt

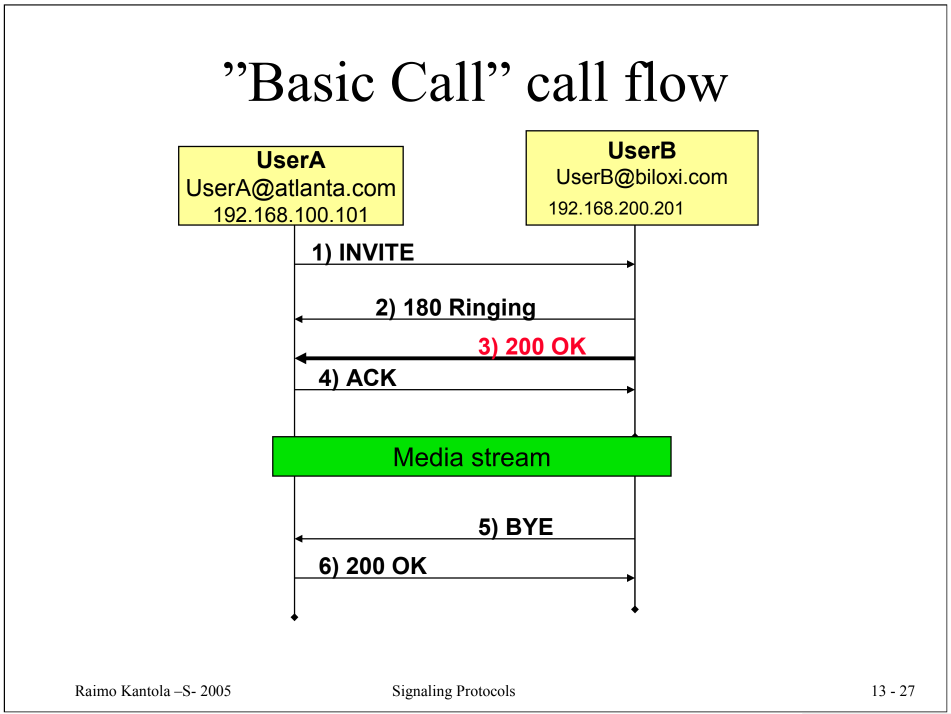


”Basic Call” call flow

```
   SIP/2.0 180 Ringing
   Via: SIP/2.0/UDP
client.atlanta.com:5060;branch=z9hG4bK74bf9
    ;received=192.168.100.101
   From: BigGuy <sip:UserA@atlanta.com>;tag=9fxced76sl
   To: LittleGuy <sip:UserB@biloxi.com>;tag=8321234356
   Call-ID: 3848276298220188511@atlanta.com
   CSeq: 1 INVITE
   Content-Length: 0
```

UserB answers with 180 Ringing provisional answer. UserB also adds tag to To: header, but maintains Call-ID and CSeq

Note that From: and To: fields are not changed. That is because UserB acts as Server (sending responses back), and call is started by UserA (in From: field). Order will be preserved untill transaction is complete



”Basic Call” call flow

UserA
UserA@atlanta.com
192.168.100.101

UserB
UserB@biloxi.com
192.168.200.201

1) INVITE

2) 180 Ringing

3) 200 OK

4) ACK

Media stream

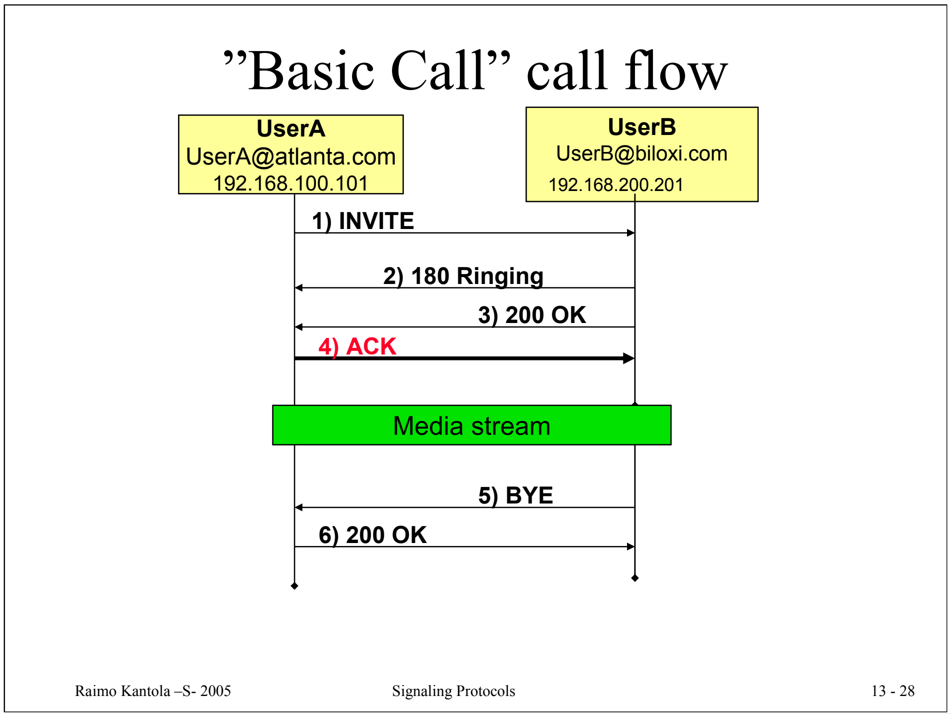
5) BYE

6) 200 OK

```
   SIP/2.0 200 OK
   Via: SIP/2.0/UDP
client.atlanta.com:5060;branch=z9hG4bK74bf9
    ;received=192.168.100.101
   From: BigGuy <sip:UserA@atlanta.com>;tag=9fxced76sl
   To: LittleGuy <sip:UserB@biloxi.com>;tag=8321234356
   Call-ID: 3848276298220188511@atlanta.com
   CSeq: 1 INVITE
   Contact: <sip:UserB@192.168.200.201>
   Content-Type: application/sdp
   Content-Length: 145

   v=0
   o=UserB 2890844527 2890844527 IN IP4 client.biloxi.com
   s=-
   c=IN IP4 192.168.200.201
   t=0 0
   m=audio 3456 RTP/AVP 0
   a=rtpmap:0 PCMU/8000
```

UserB now sends 200 OK final response. He also adds Contact: field (where he could be reached).

In SDP body of the message UserB gives his preferences for session establishment (audio, RTP over UDP, port 3456, PCM μ modulation with 8000 samples per second), and also his IP address in c= field.



”Basic Call” call flow

UserA
UserA@atlanta.com
192.168.100.101

UserB
UserB@biloxi.com
192.168.200.201

1) INVITE

2) 180 Ringing

3) 200 OK

4) ACK

Media stream

5) BYE

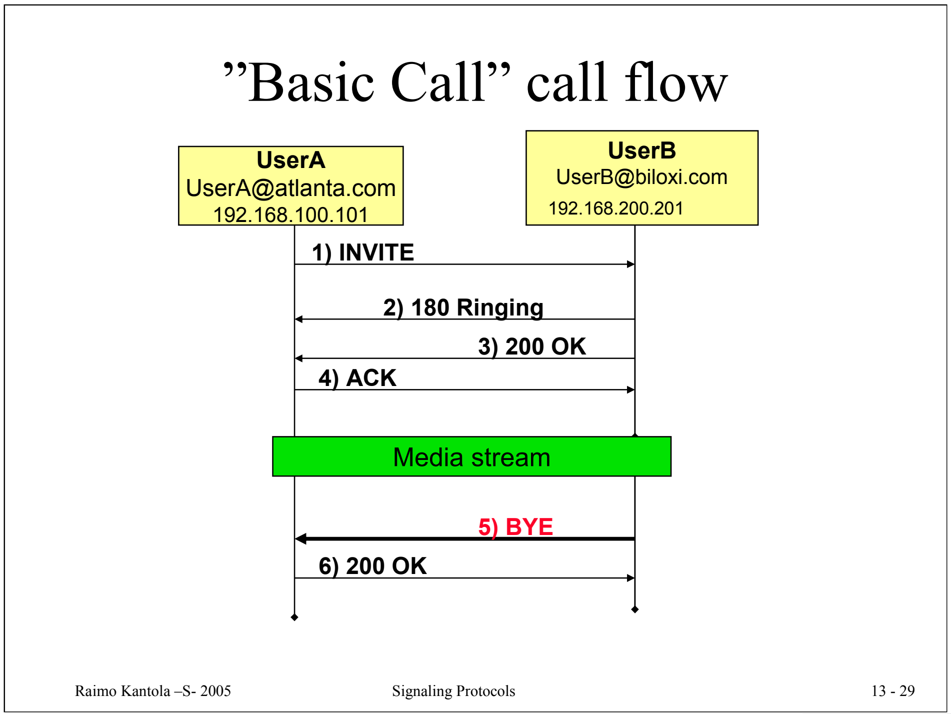
6) 200 OK

Raimo Kantola –S- 2005 Signaling Protocols 13 - 28

```
   ACK sip:UserB@biloxi.com SIP/2.0
   Via: SIP/2.0/UDP
client.atlanta.com:5060;branch=z9hG4bK74bf9
   Max-Forwards: 70
   From: BigGuy <sip:UserA@atlanta.com>;tag=9fxced76sl
   To: LittleGuy <sip:UserB@biloxi.com>;tag=8321234356
   Call-ID: 3848276298220188511@atlanta.com
   CSeq: 1 ACK
   Content-Length: 0
```

UserA acknowledges OK response. After ACK media session can be established, using parameters given in the handshake before (in INVITE and in 200 OK response)

CSeq header value is still 1 (because this is part of the same transaction), but method name has been changed to ACK

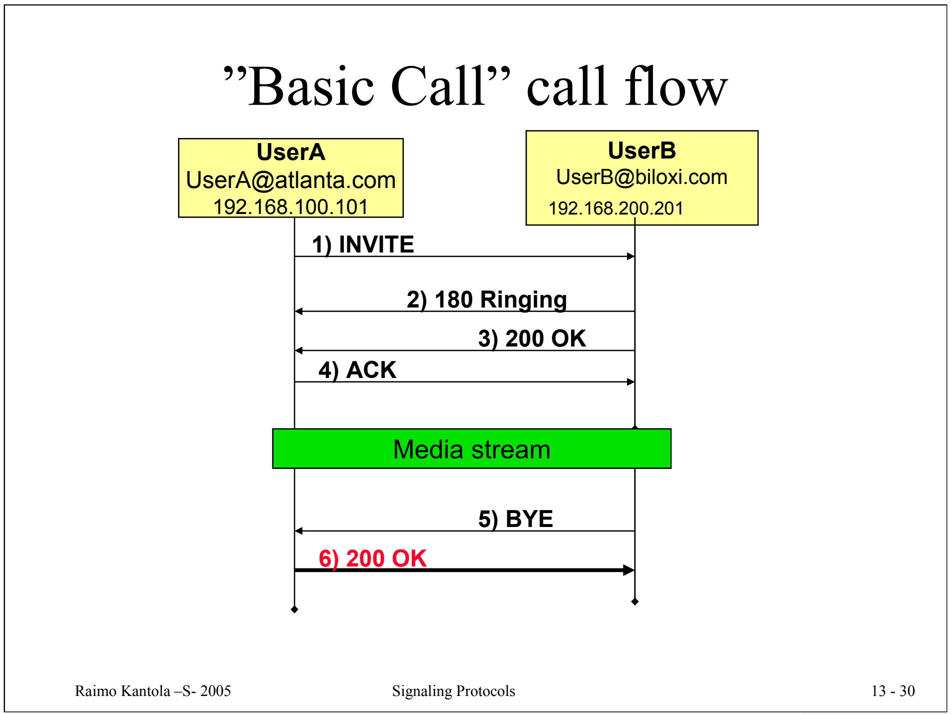


”Basic Call” call flow

```
  BYE sip:UserA@192.168.100.101 SIP/2.0
   Via: SIP/2.0/UDP
client.biloxi.com:5060;branch=z9hG4bKnashds7
   Max-Forwards: 70
   From: LittleGuy <sip:UserB@biloxi.com>;tag=8321234356
   To: BigGuy <sip:UserA@atlanta.com>;tag=9fxced76sl
   Call-ID: 3848276298220188511@atlanta.com
   CSeq: 121 BYE
   Content-Length: 0
```

After the conversation UserB wishes to end the call. It sends BYE request to UserA.

Notice that CSeq of UserB has totally different sequence. That is because UserB maintains its own CSeq numbering independently.

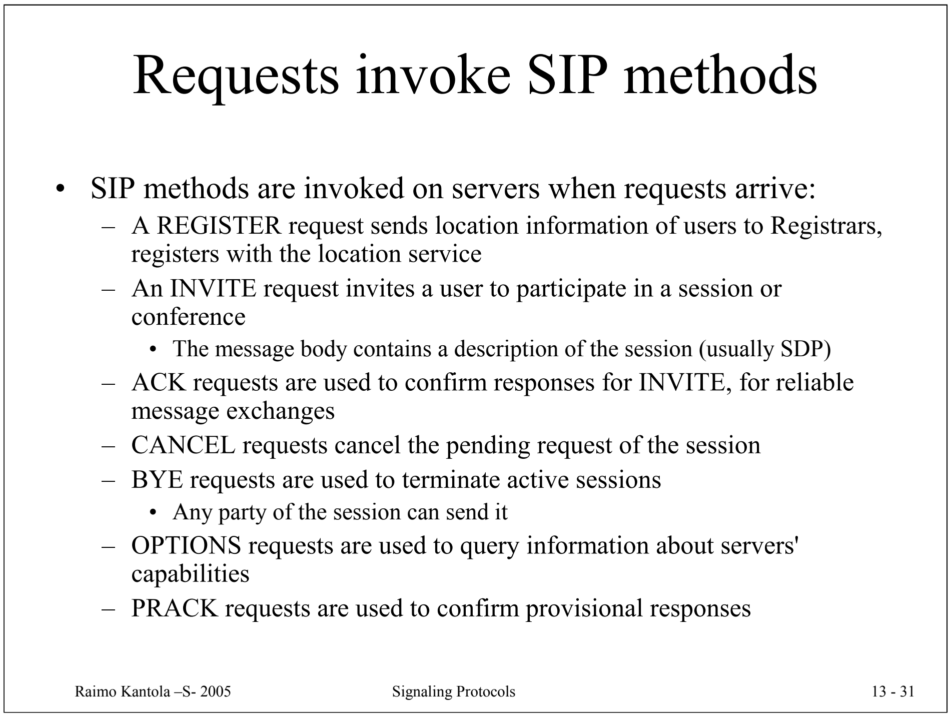


”Basic Call” call flow

Raimo Kantola –S- 2005 Signaling Protocols 13 - 30

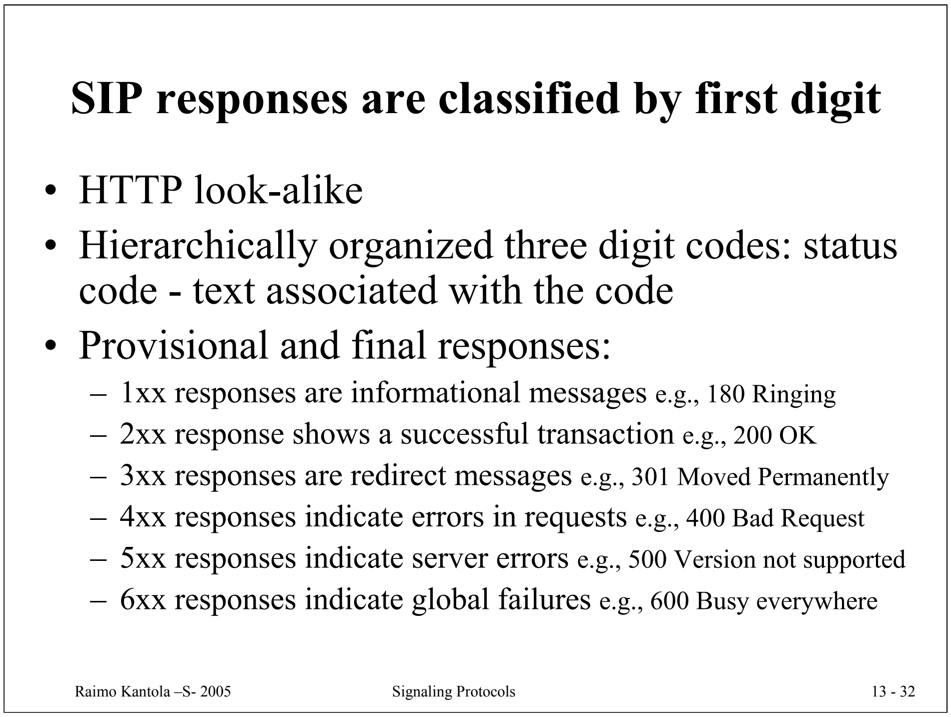
```
   SIP/2.0 200 OK
    Via: SIP/2.0/UDP
client.biloxi.com:5060;branch=z9hG4bKnashds7
     ;received=192.168.200.201
    From: LittleGuy <sip:UserB@biloxi.com>;tag=8321234356
    To: BigGuy <sip:UserA@atlanta.com>;tag=9fxced76sl
    Call-ID: 3848276298220188511@atlanta.com
    CSeq: 121 BYE
    Content-Length: 0
```

UserA sends 200 OK and conversation is ended. Note that method name (BYE) in CSeq header helps UserB to know that UserA sent OK for BYE request.



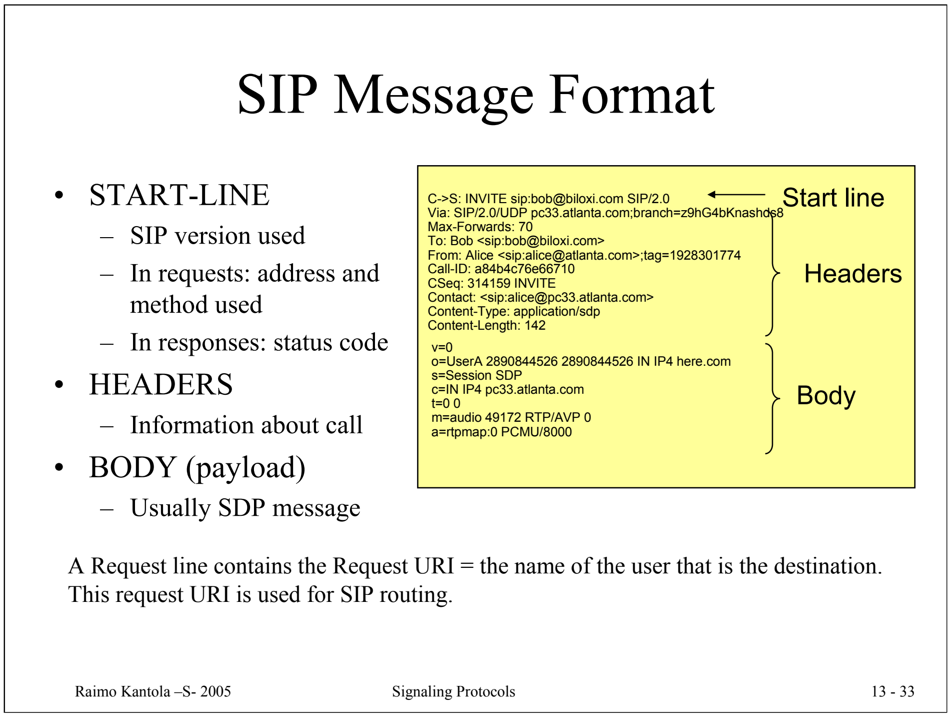
Requests invoke SIP methods

- 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
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
```

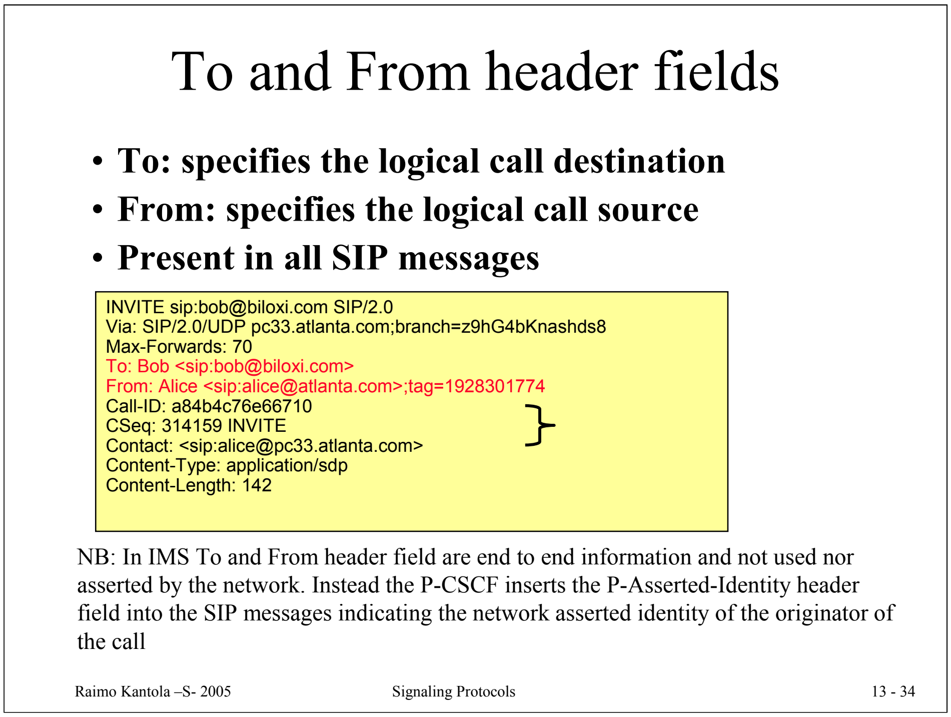
A Request line contains the Request URI = the name of the user that is the destination. This request URI is used for SIP routing.

In this example a typical INVITE message from Alice to Bob is shown. Alice is registered at domain atlanta.com, and has sip address sip:alice@atlanta.com.

Alice tries to invite Bob at his sip address bob@biloxi.com

Headers will be explained later. It is important to know that there are several mandatory headers in every SIP message. Headers always end with Content-Lenght header, that shows how long is the body of the message (in bytes)

In this example, body of the message is SDP (Session Description Protocol). It gives the description of the session that Alice wants to establish (types of codecs, session parameters, etc).



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

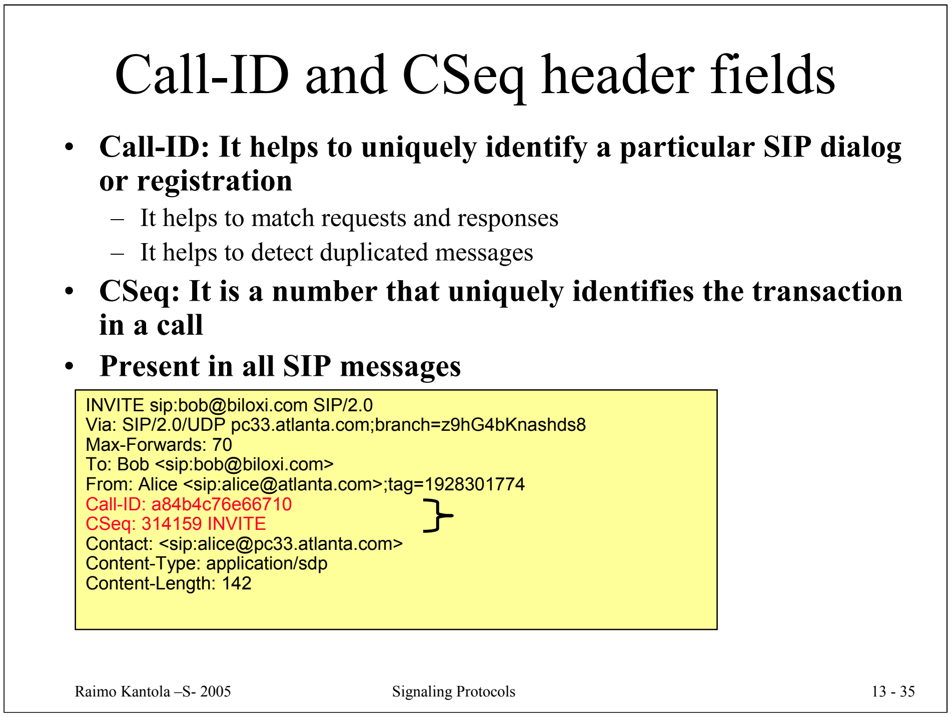
To: header specifies the logical call destination. In other words, this is the recipient’s address. Optional parameter ”display-name” (in this case ”Bob”) is just for human-user interface, while sip address is sip:bob@biloxi.com. The value of To –field is not used for SIP routing, rather it is used for filtering at the destination and for human consumption. A tag –parameter can be appended to distinguish different UAs that are identified by the same URI.

From: header is the logical address of the originator of the request. ”Logical” address means that it does not necessarily mean that it is the current address where Alice is reachable. In this case, Alice registered herself, and is contactable on sip:alice@pc33.atlanta.com. But Alice also has her permanent logical sip address sip:alice@atlanta.com. This value can be used e.g. for filtering purposes by the callee UA. Proxy server will forward request to her logical addresses to her current address.

”tag” parameter is used in ’To’ and ’From’ header fields. Tag is used to identify a dialog* between parties. The dialog is identified by combination of tags from both parties (in ’To’ and in ’From’ heders), plus Call-ID parameter.

In this example, Alice’s User Agent added ’tag=1928301774’ (which is unique value, created by Alice’s User Agent, and it is always different)

Note: In this example there is no tag in To: field. The reason for this is that this message is initial INVITE, and Bob’s User Agent will create tag when receives message.



Call-ID and CSeq header fields

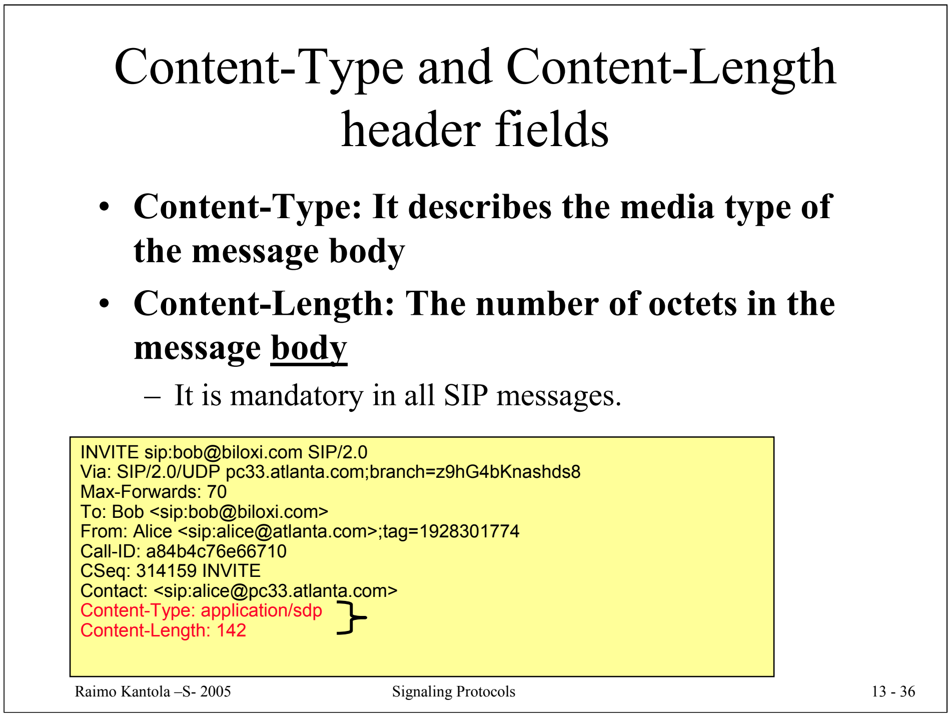
- **Call-ID: It helps to uniquely identify a particular SIP dialog or registration**
  - It helps to match requests and responses
  - It helps to detect duplicated messages
- **CSeq: It is a number that uniquely identifies the transaction in a call**
- **Present in all SIP messages**

```
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
```

Call-ID header consists of locally unique number (generated by User Agent) and @domain parameter, making it globally unique. The algorithm for making locally unique number is implementation issue. There are specifications available how to make algorithms in a way that will give unique numbers at the end.

Call-ID, together with tags in From: and To: headers identifies particular dialog. (see previous slides’s notes for details)

Cseq: header contains single decimal sequence number and the request method. This number helps to order thransactions within one dialog, and to help differentiate between retransmissions and new requests. For example, callee can receive one request with CSeq:31 INVITE. Callee may answer to this request (e.g. with ”200 OK” response). If callee receives again INVITE with same CSeq: 31 INVITE, it will know that its response was lost. It is also sure that this is not reinvitation, because CSeq nuber is the same.



Content-Type and Content-Length header fields

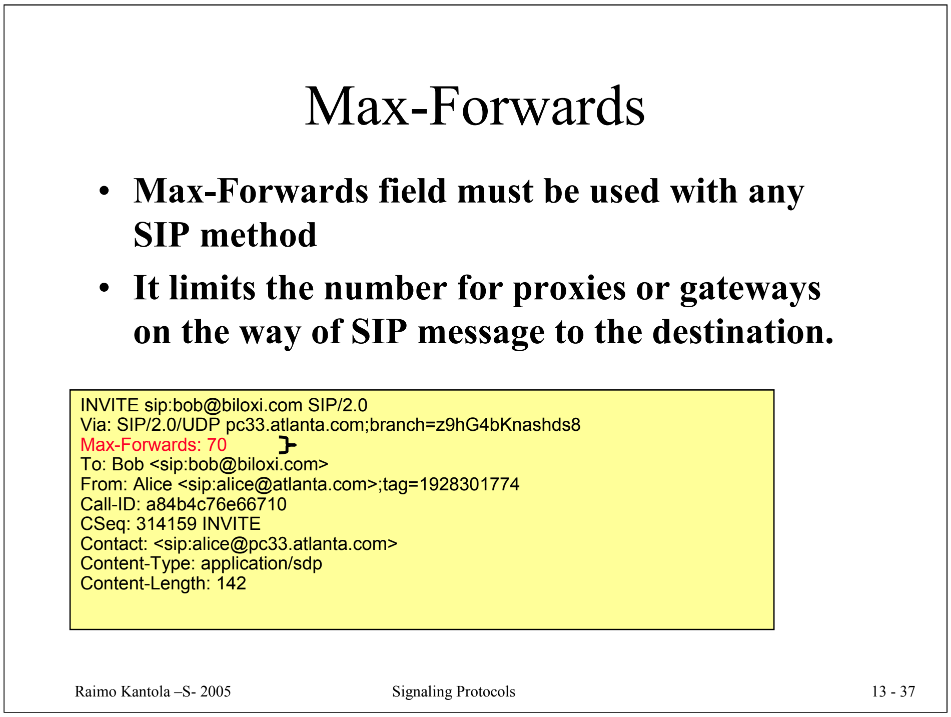
- **Content-Type: It describes the media type of the message body**
- **Content-Length: The number of octets in the message body**
  - It is mandatory in all SIP messages.

```
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
```

Content-Type header gives the media type of the body of the message. It must be present if there is some data in the body. Content-Type is header reused from HTTP and it has same values, e.g. text/html for html body or application/sdp for SDP.

Content-Length gives the size of the body. It does not include CRLF separating header fields and body (headers and body are always separated with one blank line).

If there is no body in the message, then Content-Length is set to 0.



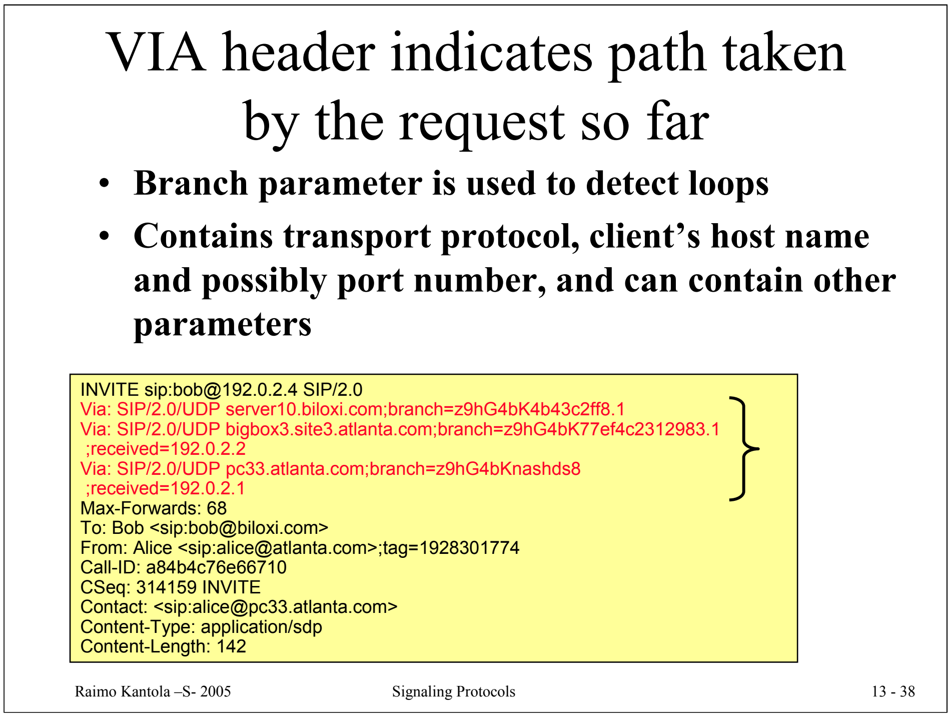
Max-Forwards

- **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
```

This field is used to prevent looping of the message. Every proxy that accepts the message will decrease this field by 1. If it reaches 0, then message is discarded, and message ”483 Too Many Hops” is sent to originator.

Every User Agent Client must insert Max-Forwards header field into each request. Value to be set is recommended to be 70. This number is big enough to prevent message being discarded unnecessarily (if message cannot reach destination in 70 hops, it is assumed that something is not configured properly anyway).



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
```

Via: header is added by every entity that the request passes. It also determines the path for the response.

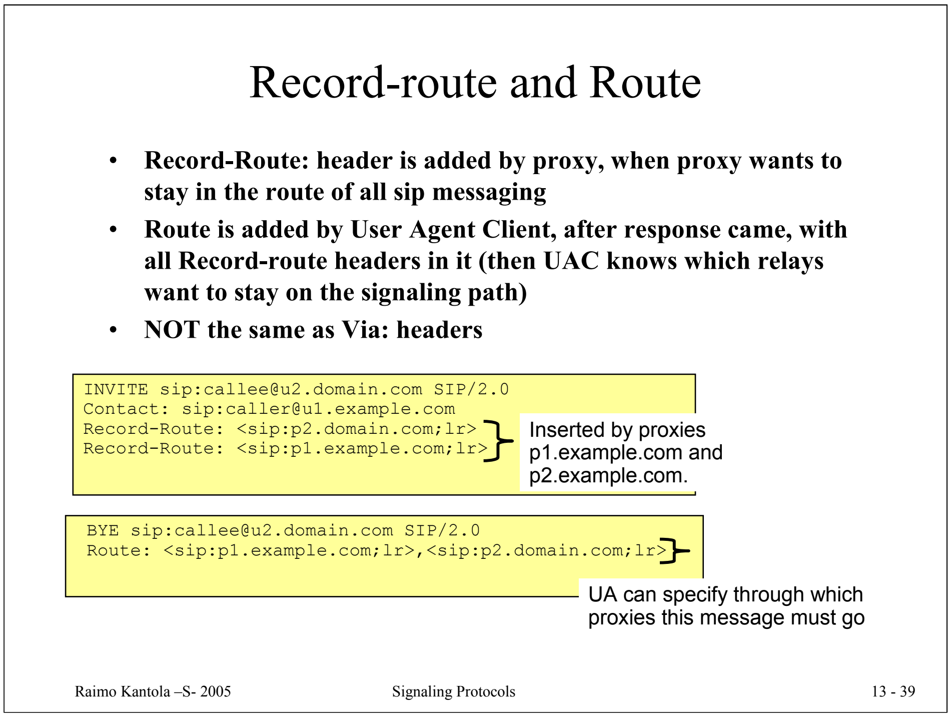
Every relay on the path of the request will put its address in the topmost Via: header.

In the response, when a relay receives the message, it will check if the topmost Via: header is its address. It can determine what is particular call by branch parameter.

It will then examine where to send response back (topmost Via: header value, after removing Via: with its own address)

Generally, for the requests every proxy on the way adds one Via: header with its address as value.

In sending response back, Via: header value is used for routing the response back.



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

(explanation from RFC 3261) U1 sends:

```
      INVITE sip:callee@domain.com SIP/2.0
      Contact: sip:caller@u1.example.com
```

to P1. P1 is an outbound proxy. P1 is not responsible for domain.com, so it looks it up in DNS and sends it there. It also adds a Record-Route header field value:

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

P2 gets this. It is responsible for domain.com so it runs a location service and rewrites the Request-URI. It also adds a Record-Route header field value. There is no Route header field, so it resolves the new Request-URI to determine where to send the request:

```
      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>
```

The callee at u2.domain.com gets this and responds with a 200 OK:

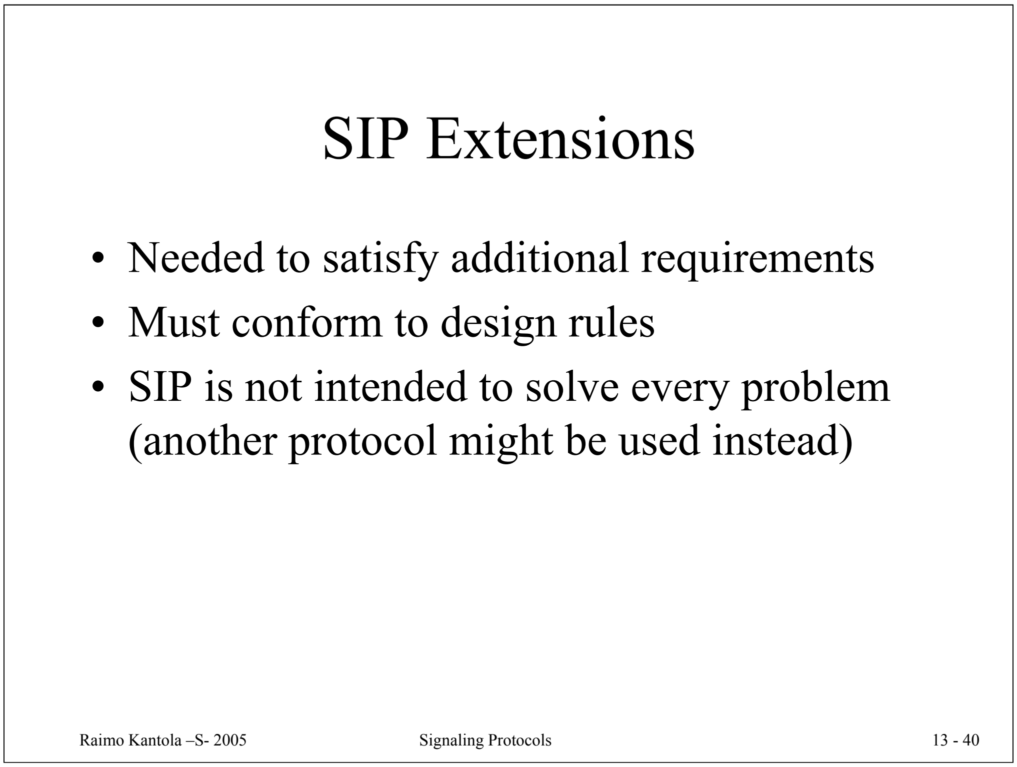
```
      SIP/2.0 200 OK
      Contact: sip:callee@u2.domain.com
      Record-Route: <sip:p2.domain.com;lr>
      Record-Route: <sip:p1.example.com;lr>
```

The callee at u2 also sets its dialog state's remote target URI to sip:caller@u1.example.com and its route set to:

```
      (<sip:p2.domain.com;lr>,<sip:p1.example.com;lr>)
```

This is forwarded by P2 to P1 to U1 as normal. Now, U1 sets its dialog state's remote target URI to sip:callee@u2.domain.com and its route set to:

```
      (<sip:p1.example.com;lr>,<sip:p2.domain.com;lr>)
```



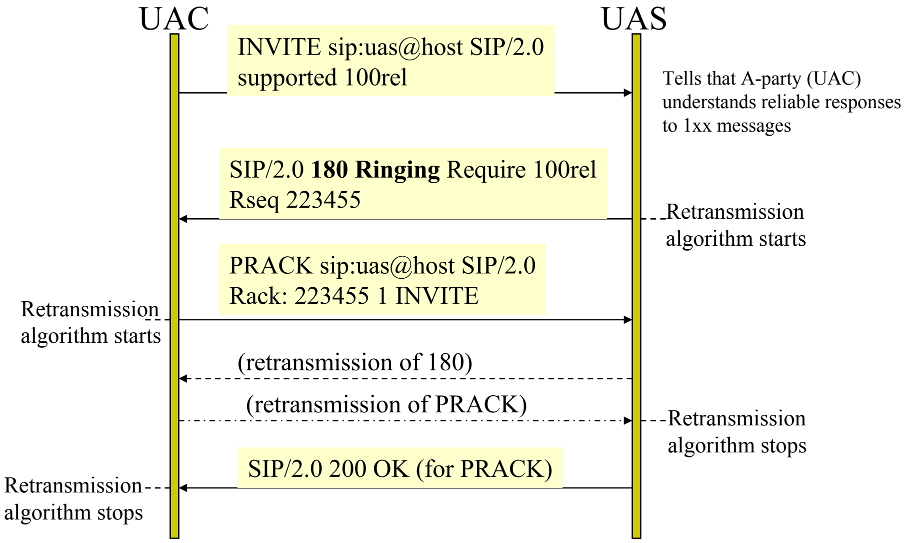
SIP Extensions

- 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



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

Tells that A-party (UAC) understands reliable responses to 1xx messages

SIP/2.0 **180 Ringing** Require 100rel
Rseq 223455

Retransmission algorithm starts

PRACK sip:uas@host SIP/2.0
Rack: 223455 1 INVITE

Retransmission algorithm starts

(retransmission of 180)

(retransmission of PRACK)

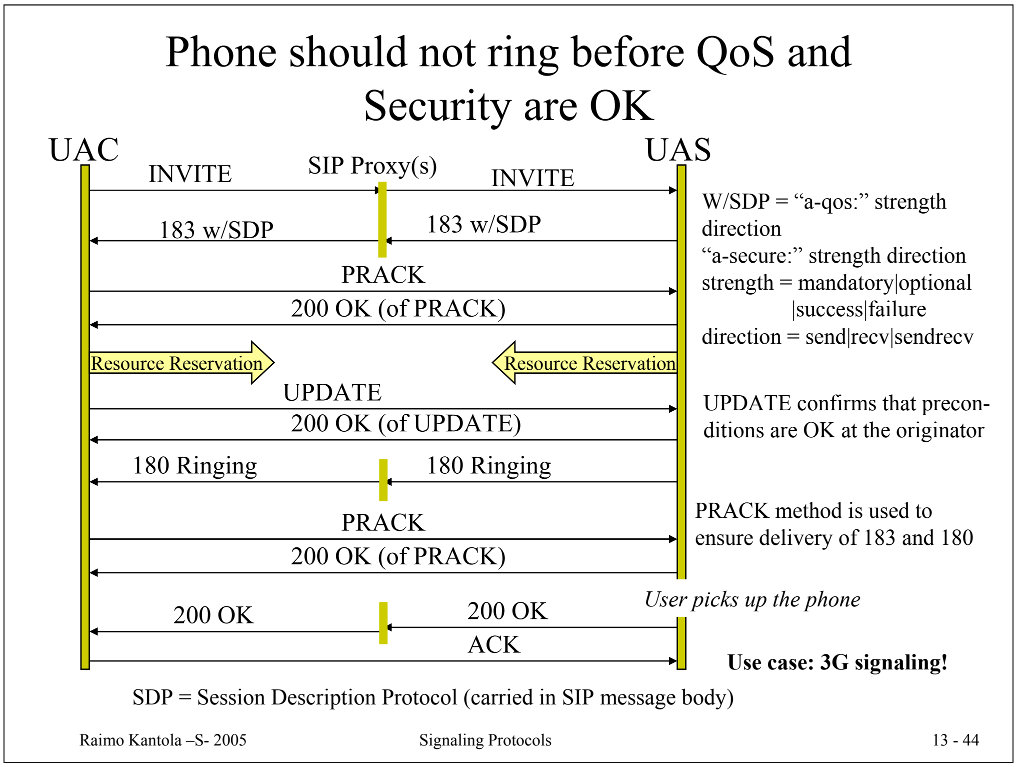
Retransmission algorithm stops

SIP/2.0 200 OK (for PRACK)

Retransmission algorithm stops

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

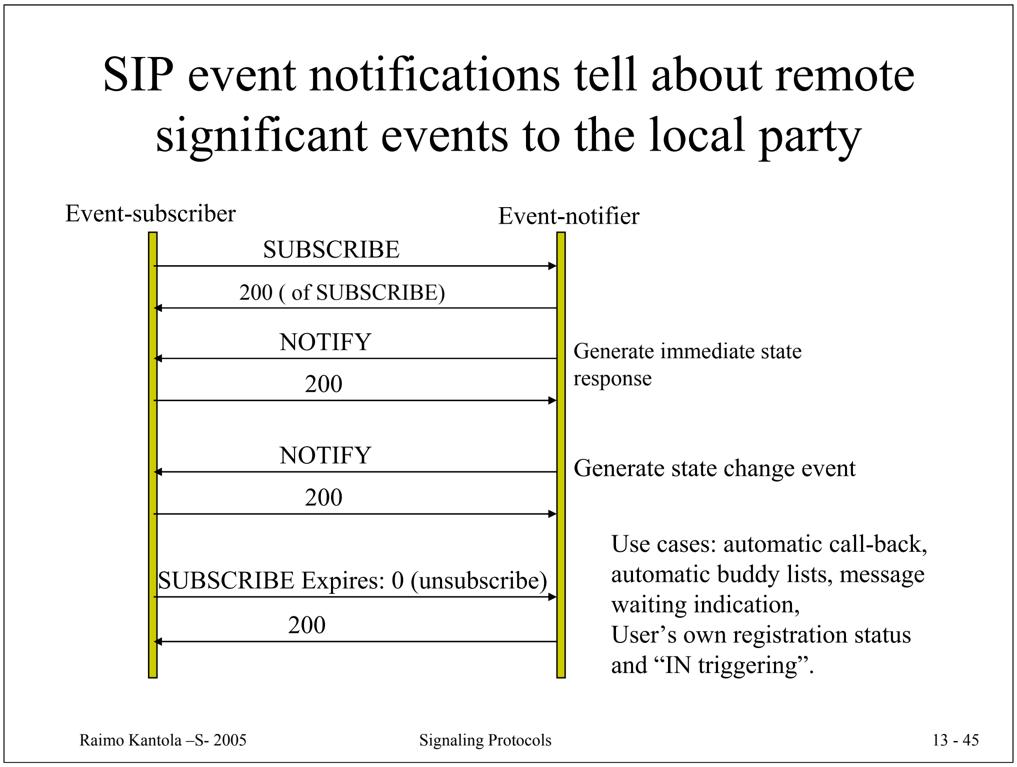
User picks up the phone

Use case: 3G signaling!

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

For IMS the this use case is similar to what is found in RFC 3312 ”Integration of Resource management and SIP.

This approach allows finding out whether the callee is reachable before any resources are dedicated for the call and thus saving radio resources at the originating end for a period that can not be billed. When ringing is sent by the callee, preconditions are OK also at the callee, i.e. the radio resources for the media plane have been reserved.



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

Event-subscriber

Event-notifier

SUBSCRIBE

200 (of SUBSCRIBE)

NOTIFY

200

Generate immediate state response

NOTIFY

200

Generate state change event

SUBSCRIBE Expires: 0 (unsubscribe)

200

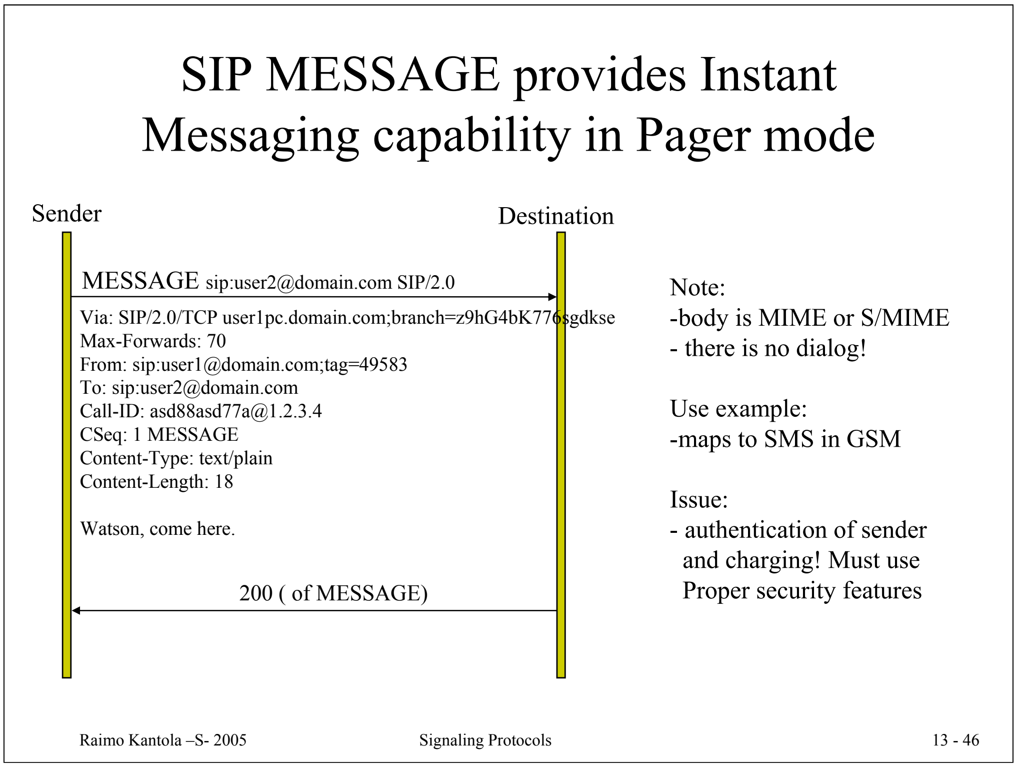
Use cases: automatic call-back, automatic buddy lists, message waiting indication,
User's own registration status and "IN triggering".

Automatic call-back = call completion on busy. A calls to a busy subscriber, subscribes to the state change of the callee. When callee becomes free, the caller is notified of the the state change. Caller INVITEs the callee again, now callee is likely to be free and the call can be established. It is important that the call is always established from the paying customer to the other party.

Message wayting indication: a user subscribes to the state of his voice mail or e-mail account. When a message arrives, the user is immediately notified that messages are waiting and the user can retrieve them at a convenient time.

A use case: A mobile customer wants to see whether she is registered in IMS in real time (cmp whether GSM phone is connected to a network). The registration may be terminated for administrative reasons at any time, a network node may die and break the registration. To have up to date information on her registration status, the UA needs to subscribe to the reg –event package at the S-CSCF that also acts as the registrar in IMS. To do this, always after the registration at S-CSCF of the home network, the UA will send the Subscribe message to the S-CSCF. Moreover, also the P-CSCF wants to have the same information. Therefore also the P-CSCF will subscribe to the same event package for all of the visiting users.

Idea of IN triggering: A service node can subscribe to events that trigger services. On occurrence of such an event the service node is notified and it can do what it needs to do. In this case a bit more is needed: the notifier needs to stop and wait for more instructions until it can proceed.



SIP MESSAGE provides Instant Messaging capability in Pager mode

Sender

Destination

MESSAGE sip:user2@domain.com SIP/2.0

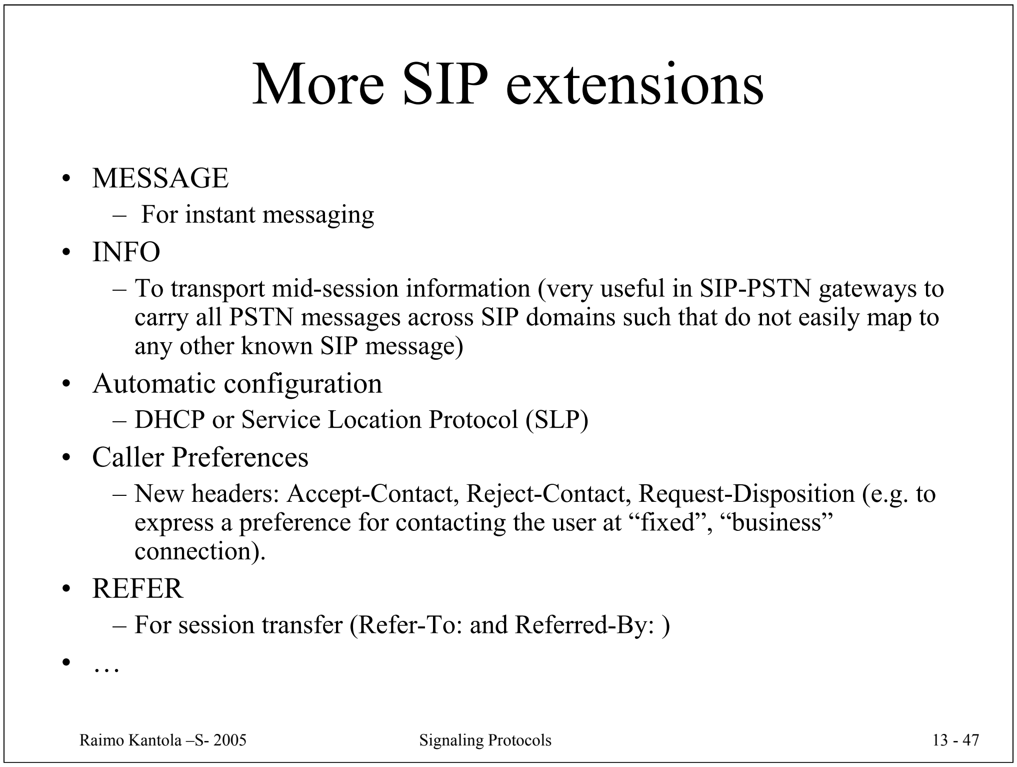
```
Via: SIP/2.0/TCP user1pc.domain.com;branch=z9hG4bK776sgdkse
Max-Forwards: 70
From: sip:user1@domain.com;tag=49583
To: sip:user2@domain.com
Call-ID: asd88asd77a@1.2.3.4
CSeq: 1 MESSAGE
Content-Type: text/plain
Content-Length: 18

Watson, come here.
```

200 (of MESSAGE)

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

Use example:
-maps to SMS in GSM

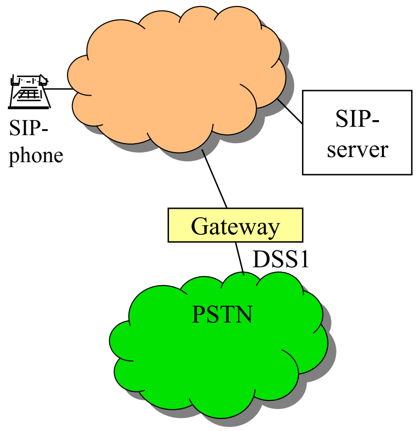
Issue:
- authentication of sender and charging! Must use Proper security features



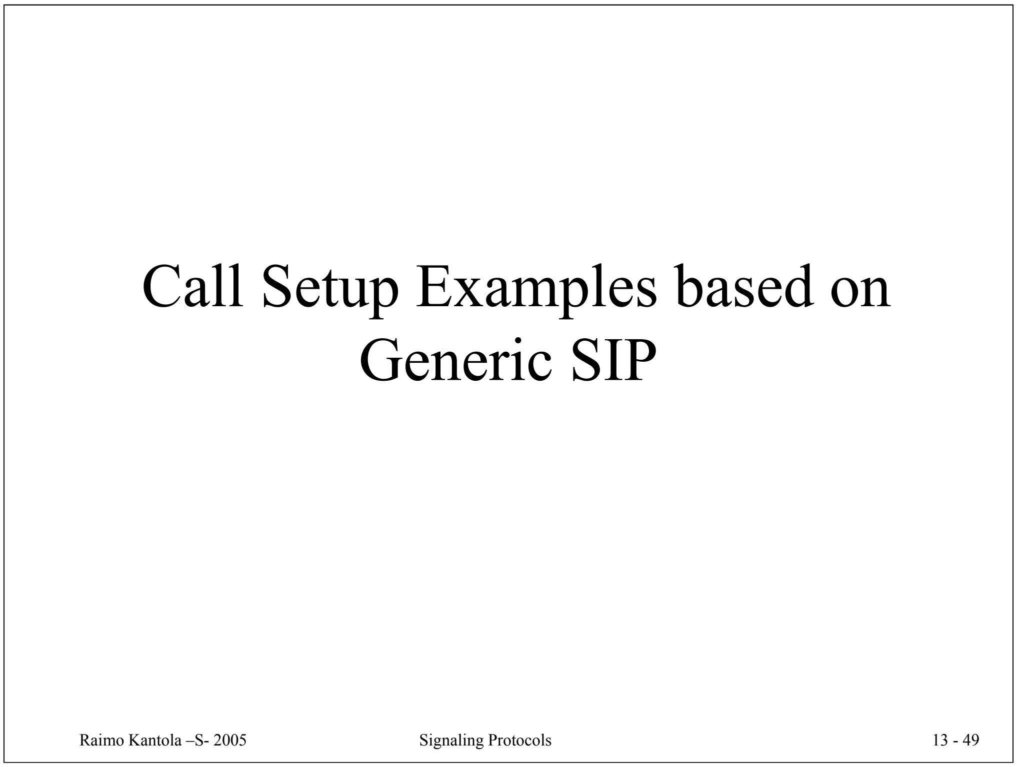
More SIP extensions

- 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:)
- …

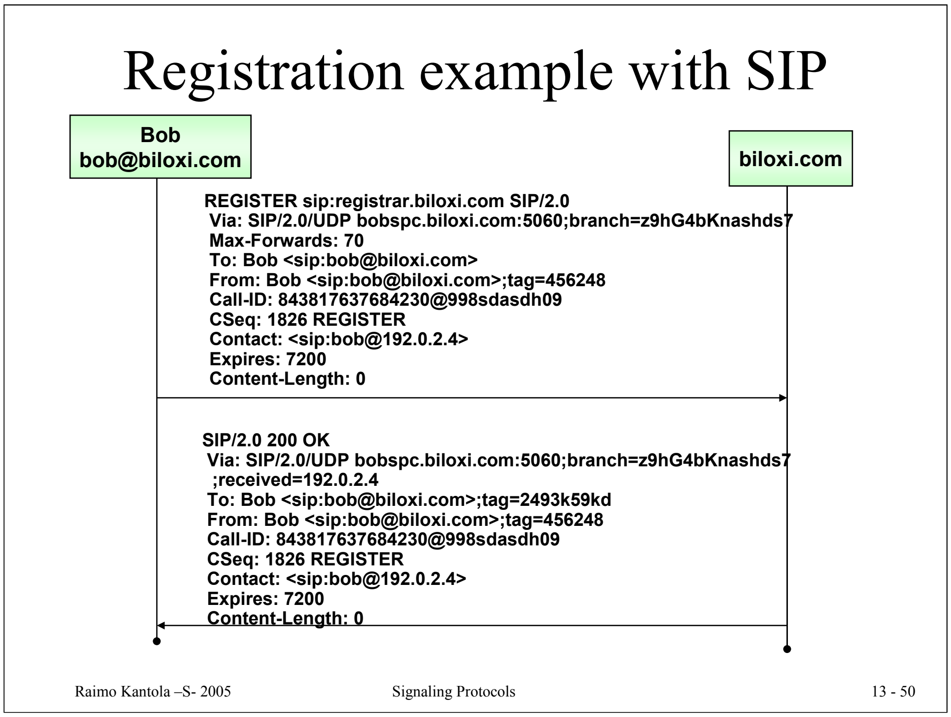
Deployment example: Elisa's experimental service for BB customers



- 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

```
REGISTER sip:registrar.biloxi.com SIP/2.0
 Via: SIP/2.0/UDP bobspc.biloxi.com:5060;branch=z9hG4bKnashds7
 Max-Forwards: 70
 To: Bob <sip:bob@biloxi.com>
 From: Bob <sip:bob@biloxi.com>;tag=456248
 Call-ID: 843817637684230@998sdasdh09
 CSeq: 1826 REGISTER
 Contact: <sip:bob@192.0.2.4>
 Expires: 7200
 Content-Length: 0
```

```
SIP/2.0 200 OK
 Via: SIP/2.0/UDP bobspc.biloxi.com:5060;branch=z9hG4bKnashds7
  ;received=192.0.2.4
 To: Bob <sip:bob@biloxi.com>;tag=2493k59kd
 From: Bob <sip:bob@biloxi.com>;tag=456248
 Call-ID: 843817637684230@998sdasdh09
 CSeq: 1826 REGISTER
 Contact: <sip:bob@192.0.2.4>
 Expires: 7200
 Content-Length: 0
```

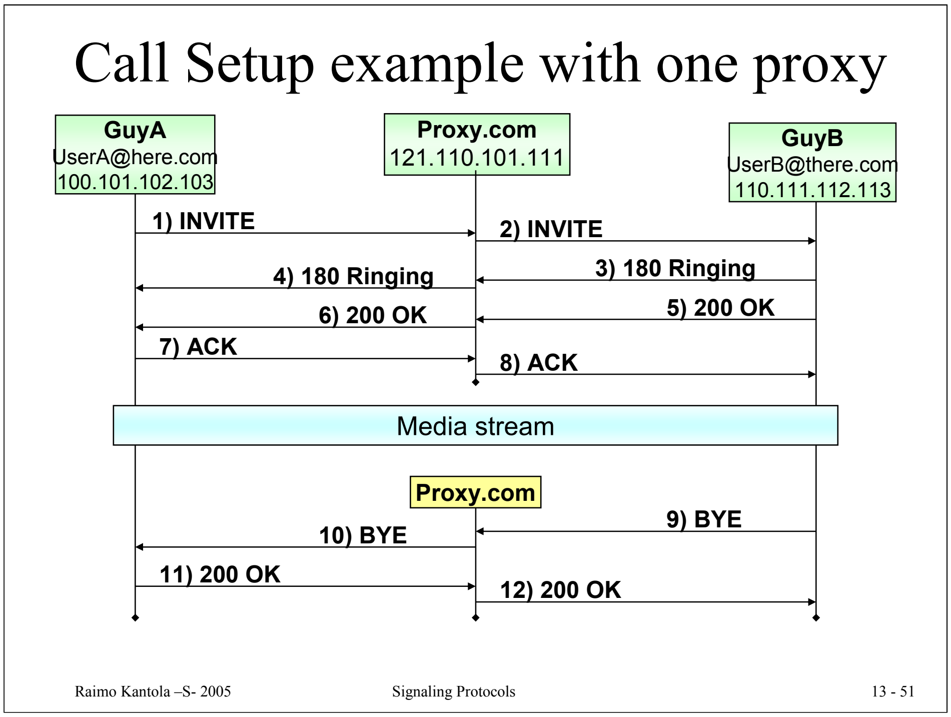
Raimo Kantola –S- 2005 Signaling Protocols 13 - 50

In this example Bob’s User Agent performs a successful registration to a registrar whose domain name is biloxi.com.

Registration is set to expire after two hours (7200 seconds). It may be seen in header Expires:

Registrar answers with 200 OK response, meaning that registration was succesfull.

(example from RFC3261)



Call Setup example with one proxy

GuyA
UserA@here.com
100.101.102.103

Proxy.com
121.110.101.111

GuyB
UserB@there.com
110.111.112.113

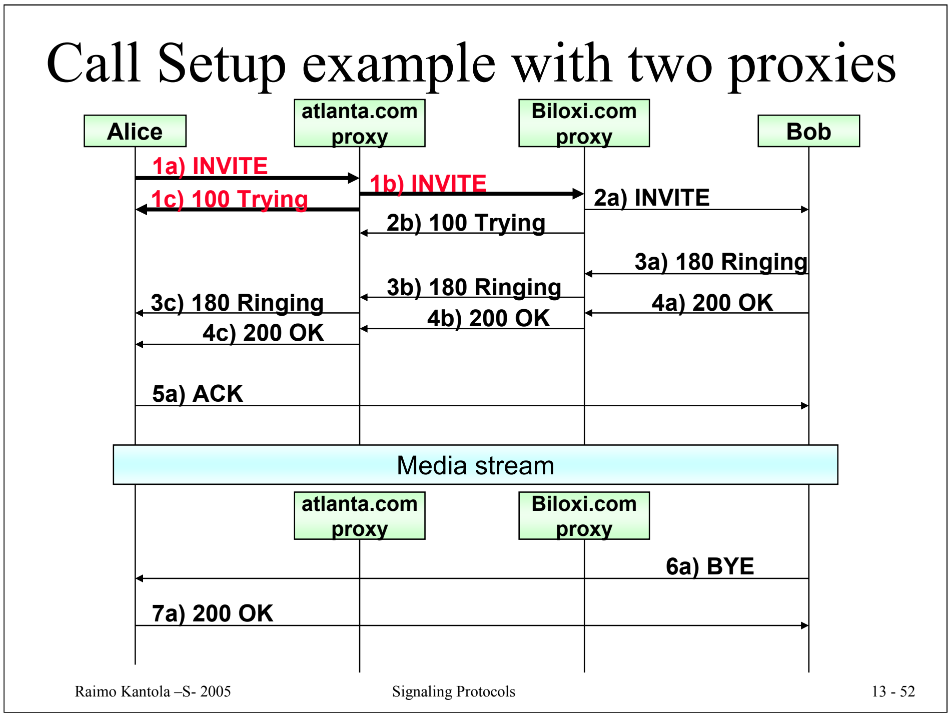
1) INVITE
2) INVITE
3) 180 Ringing
4) 180 Ringing
5) 200 OK
6) 200 OK
7) ACK
8) ACK

Media stream

Proxy.com

9) BYE
10) BYE
11) 200 OK
12) 200 OK

In this example GuyA (UserA@here.com at 100.101.102.103) performs a successful call (session) initiation to GuyB (UserB@there.com at 100.101.102.103). For the sake of simplicity in the example both party is registered the same SIP proxy (proxy.com).



Call Setup example with two proxies

Raimo Kantola –S- 2005 Signaling Protocols 13 - 52

In this example Alice, who is registered to atlanta.com, performs a successful call (session) initiation to Bob, who is registered to biloxy.com.

1a) Alice -> atlanta.com proxy

```
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
```

(Alice's SDP not shown)

1b) INVITE atlanta.com proxy -> biloxi.com proxy

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com
;branch=z9hG4bK77ef4c2312983.1
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
 ;received=192.0.2.1
Max-Forwards: 69
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
```

(Alice's SDP not shown)

1c) 100 Trying atlanta.com proxy -> Alice

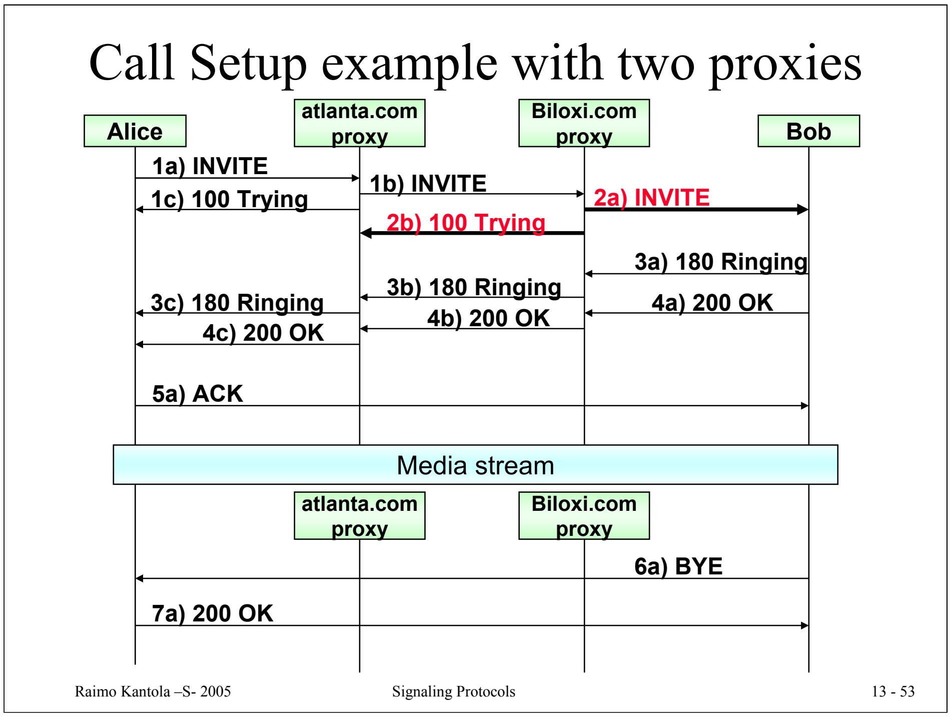
```
SIP/2.0 100 Trying
```

Alice's UA sets Max-Forwards to 70. INVITE is addressed to logical SIP address, and Alice relies on proxy to find Bob. Alice is currently at pc33.atlanta.com and she puts current address in header Contact:

Atlanta.com proxy decreases Max-forwards by 1. Adds Via: header (puts its own address and branch there)

Atlanta.com leaves all other headers unchanged and resolves Bob's domain proxy. It then sends INVITE to biloxi.com proxy

After atlanta.com has forwarded INVITE to biloxi.com proxy, it sends 100 Trying to Alice. It will make her User Agent aware that call establishment is in the process (everything goes OK). It will also prevent Alice from retransmitting INVITE



Call Setup example with two proxies

Raimo Kantola –S- 2005 Signaling Protocols 13 - 53

In this example Alice, who is registered to atlanta.com, performs a successful call (session) initiation to Bob, who is registered to biloxy.com.

2a) biloxi.com proxy -> Bob

```
INVITE sip:bob@192.0.2.4 SIP/2.0
Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.
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
```

(Alice's SDP not shown)

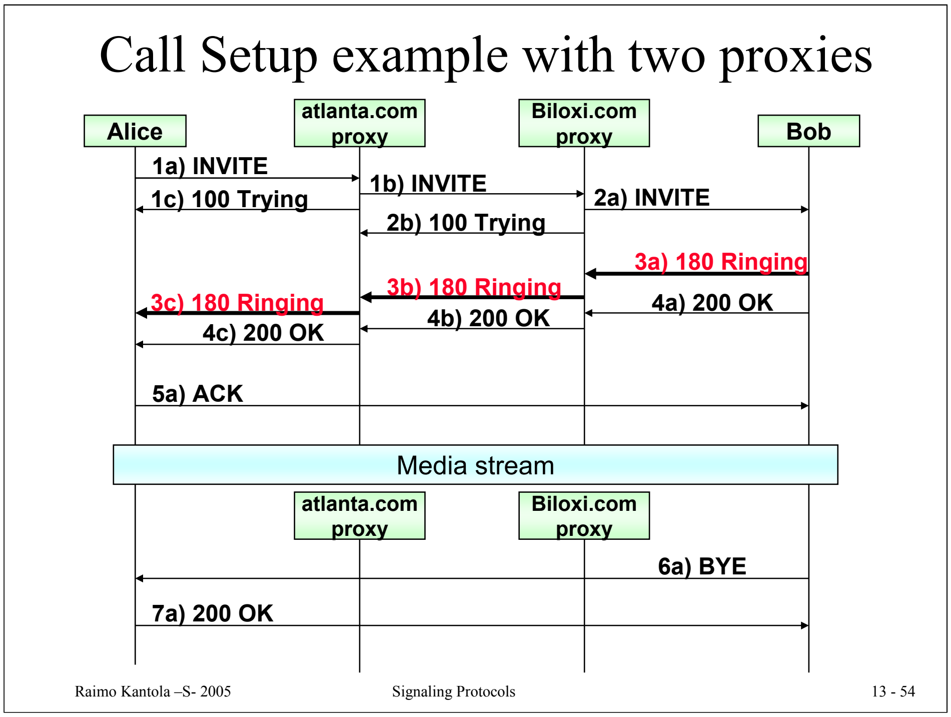
Biloxi.com proxy decreases max-forwards by 1. It also adds its address in Via: header, and assigns branch to it.

Note that in first line there is Bob's current address (sip:bob@192.0.2.4). It has been obtained from Location server.

2b) TRYING biloxi.com proxy -> atlanta.com proxy

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com
;branch=z9hG4bK77ef4c2312983.1
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
 ;received=192.0.2.1
Max-Forwards: 69
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
```

After biloxi.com has forwarded INVITE to Bob, it sends 100 Trying to átlanta.com proxy.



Call Setup example with two proxies

Raimo Kantola –S- 2005 Signaling Protocols 13 - 54

In this example Alice, who is registered to atlanta.com, performs a successful call (session) initiation to Bob, who is registered to biloxy.com.

3a) Bob -> biloxi.com proxy

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP server10.biloxi.com
;branch=z9hG4bK4b43c2ff8.1;received=192.0.2.3
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
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
Contact: <sip:bob@192.0.2.4>
CSeq: 314159 INVITE
Content-Length: 0
```

Bob's UA sends 180 Ringing provisional response. It will send it back via the same route (usin Via: headers). Response will go to the topmost Via: address.

Also, Bob's UA adds tag to To: header. Now dialog is completely determined (with Call-ID, From: and To: tags)

3b) biloxi.com proxy -> atlanta.com proxy

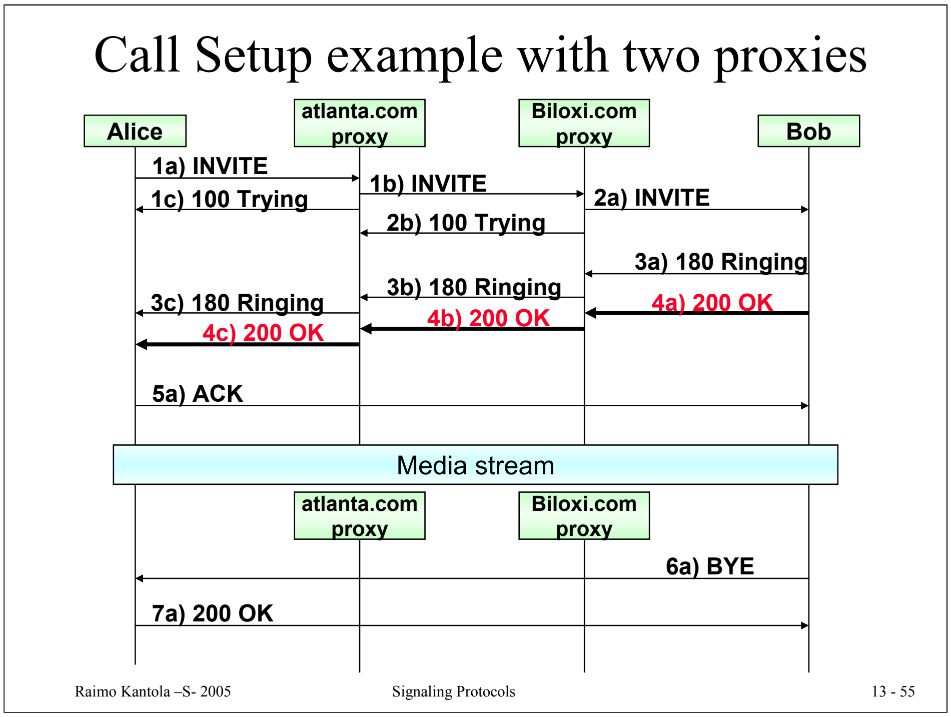
```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com
;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP pc33.atlanta.com
;branch=z9hG4bKnashds8 ;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
Contact: <sip:bob@192.0.2.4>
CSeq: 314159 INVITE
Content-Length: 0
```

From biloxi.com proxy to atlanta.com proxy.

3c) atlanta.com proxy -> Alice

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
 ;received=192.0.2.1
```

Alice receives 180 Ringing response. Now Alice's UA knows that Bob has been alerted.



Call Setup example with two proxies

Raimo Kantola –S- 2005 Signaling Protocols 13 - 55

4a) Bob -> biloxi.com proxy

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP server10.biloxi.com
;branch=z9hG4bK4b43c2ff8.1;received=192.0.2.3
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
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:bob@192.0.2.4>
Content-Type: application/sdp
Content-Length: 131
```

(Bob's SDP not shown)

4b) biloxi.com proxy -> atlanta.com proxy

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com;branch=z9hG4bK77ef
 ;received=192.0.2.2
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
 ;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:bob@192.0.2.4>
Content-Type: application/sdp
Content-Length: 131
```

(Bob's SDP not shown)

4c) 100 Trying atlanta.com proxy -> Alice

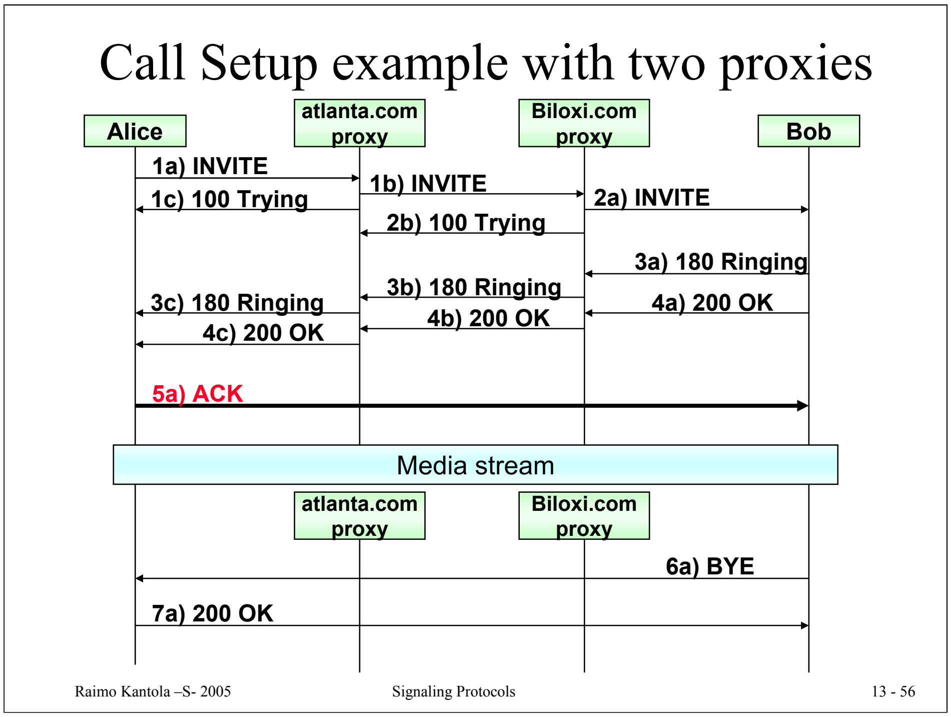
```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
 ;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
```

Bob accepts invitation and sends 200 OK final response. It will go through the same set of proxies (determined by Via: headers)

Bob puts his current addres in Contact: header. Alice is now able to contact him directly. Subsequent SIP messages may go directly to Bob, and not through proxies

Biloxi.com proxy forwards response to topmost via: header address. (it removed its own address in Via: header previously)

Alice receives 200 OK with Bob's session parameters in the message body (not shown here). If Alice can accept it, she will send ACK message back to Bob directly. Now Alice has Bob's address where he is contactable in Contact: header.



Call Setup example with two proxies

Alice

atlanta.com proxy

Biloxi.com proxy

Bob

1a) INVITE

1b) INVITE

1c) 100 Trying

2a) INVITE

2b) 100 Trying

3a) 180 Ringing

3b) 180 Ringing

3c) 180 Ringing

4a) 200 OK

4b) 200 OK

4c) 200 OK

5a) ACK

Media stream

atlanta.com proxy

Biloxi.com proxy

6a) BYE

7a) 200 OK

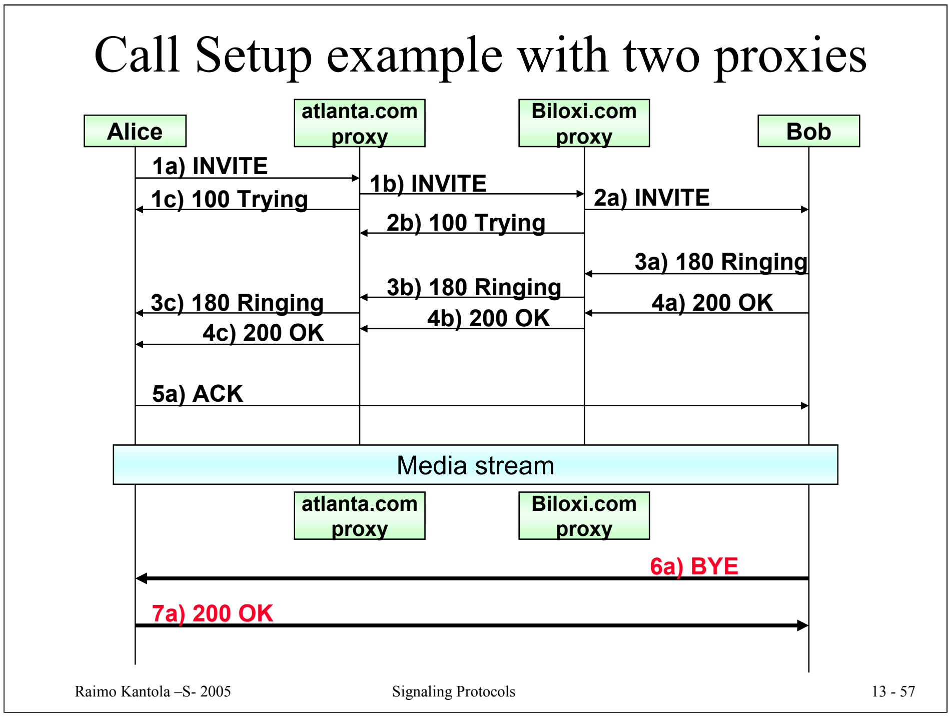
Raimo Kantola –S- 2005 Signaling Protocols 13 - 56

In this example Alice, who is registered to atlanta.com, performs a successful call (session) initiation to Bob, who is registered to biloxy.com.

5a) Alice -> Bob

```
ACK sip:bob@192.0.2.4 SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds9
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 ACK
Content-Length: 0
```

The media session between Alice and Bob is now established. They agreed on session parameters (described in SDP body)



Call Setup example with two proxies

Alice

atlanta.com proxy

Biloxi.com proxy

Bob

1a) INVITE

1b) INVITE

2a) INVITE

1c) 100 Trying

2b) 100 Trying

3a) 180 Ringing

3b) 180 Ringing

3c) 180 Ringing

4a) 200 OK

4b) 200 OK

4c) 200 OK

5a) ACK

Media stream

atlanta.com proxy

Biloxi.com proxy

6a) BYE

7a) 200 OK

Raimo Kantola –S- 2005 Signaling Protocols 13 - 57

6a) Bob -> Alice

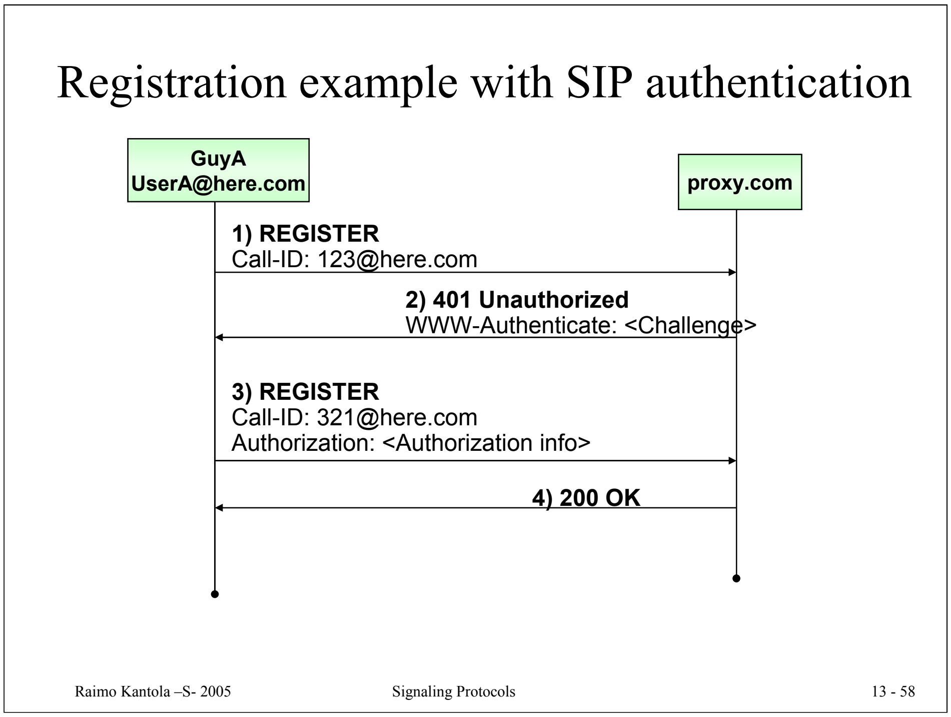
```
BYE sip:alice@pc33.atlanta.com SIP/2.0
Via: SIP/2.0/UDP 192.0.2.4;branch=z9hG4bKnashds10
Max-Forwards: 70
From: Bob <sip:bob@biloxi.com>;tag=a6c85cf
To: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 231 BYE
Content-Length: 0
```

Bob after a while decides to disconnect. Bob's UA has its own CSeq sequencing (note) and From: and To: fields are swapped, because Bob is originating a request. But Bob still refers to the same dialog (can be seen by Call-ID and tags)

7a) Bob -> Alice

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.0.2.4;branch=z9hG4bKnashds10
From: Bob <sip:bob@biloxi.com>;tag=a6c85cf
To: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 231 BYE
Content-Length: 0
```

Alice acknowledges BYE, and call is over. This 200 OK will refer to the BYE request, and that can be seen from CSeq field, carrying BYE method name



Registration example with SIP authentication

GuyA
UserA@here.com

proxy.com

1) REGISTER
Call-ID: 123@here.com

2) 401 Unauthorized
WWW-Authenticate: <Challenge>

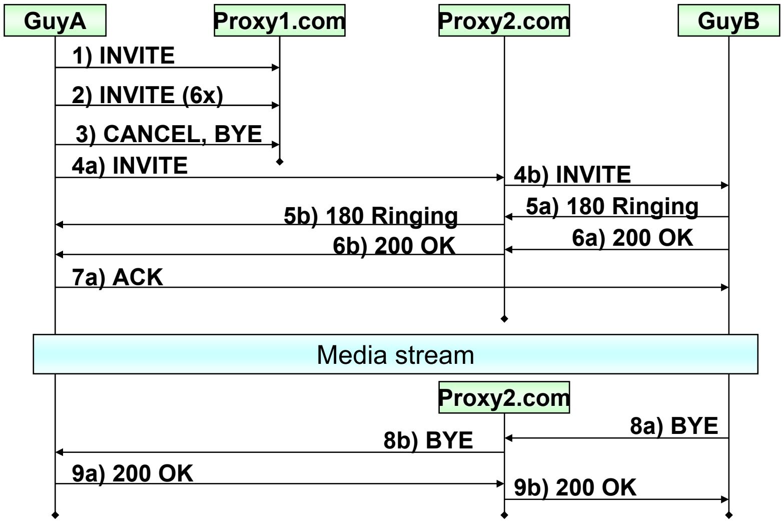
3) REGISTER
Call-ID: 321@here.com
Authorization: <Authorization info>

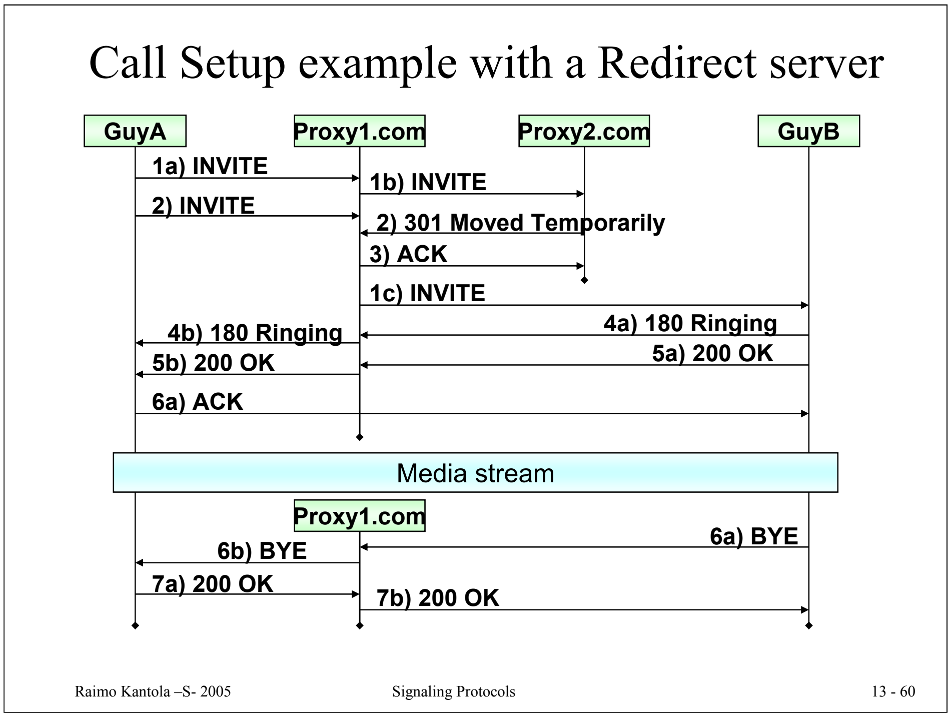
4) 200 OK

Raimo Kantola –S- 2005 Signaling Protocols 13 - 58

In this example GuyA (UserA@here.com at 100.101.102.103) performs a successful registration to a proxy whose domain name is proxy.com.

Call Setup example with a non-working proxy





Call Setup example with a Redirect server

GuyA
Proxy1.com
Proxy2.com
GuyB

1a) INVITE
1b) INVITE
2) INVITE
2) 301 Moved Temporarily
3) ACK
1c) INVITE
4a) 180 Ringing
4b) 180 Ringing
5a) 200 OK
5b) 200 OK
6a) ACK

Media stream

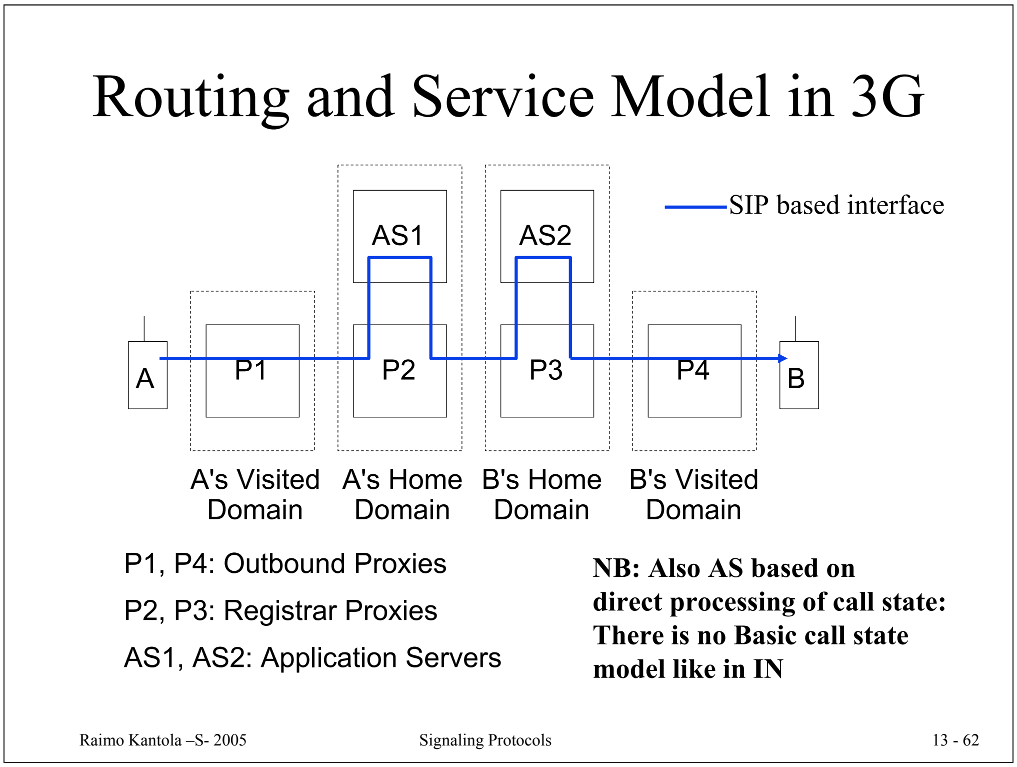
Proxy1.com
6a) BYE
6b) BYE
7a) 200 OK
7b) 200 OK

Raimo Kantola –S- 2005 Signaling Protocols 13 - 60

In this example GuyA, who is registered to proxy1.com, performs a successful call (session) initiation to GuyB, who is registered to proxy2.com.

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.



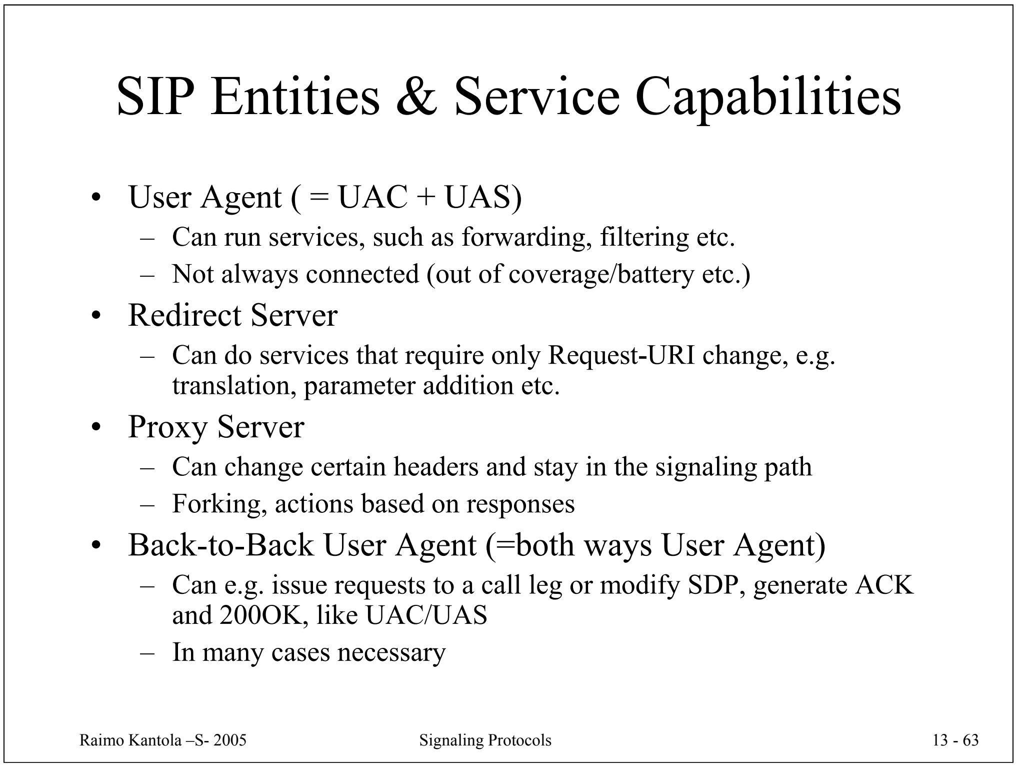
Routing and Service Model in 3G

P1, P4: Outbound Proxies

P2, P3: Registrar Proxies

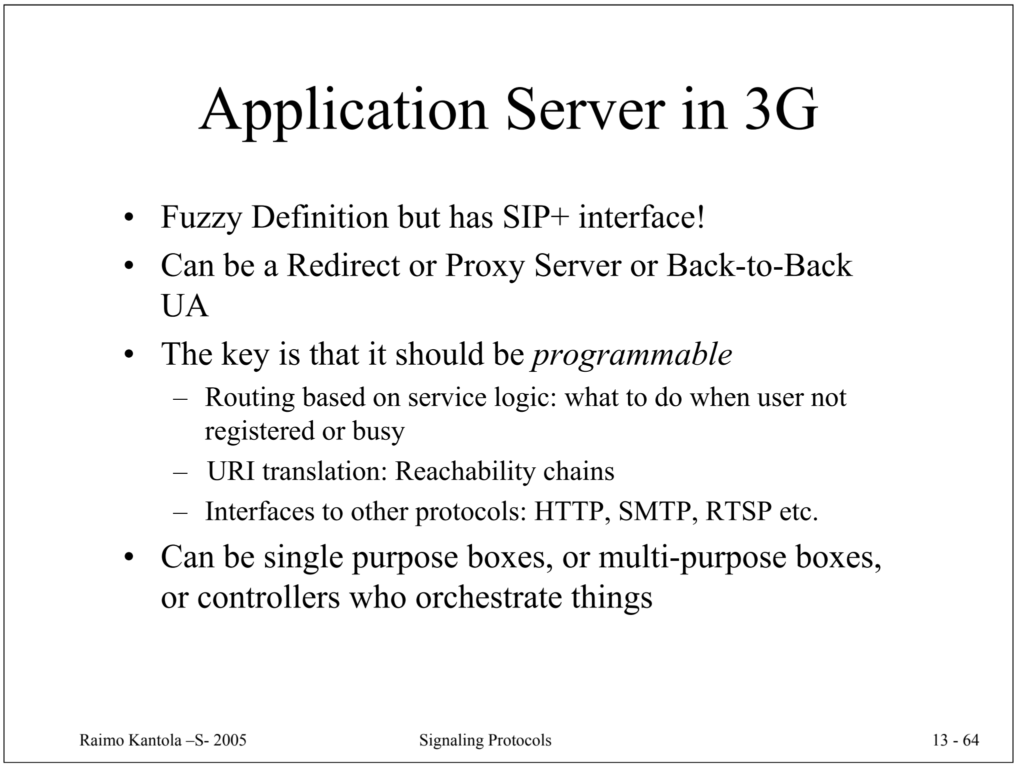
AS1, AS2: Application Servers

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



SIP Entities & Service Capabilities

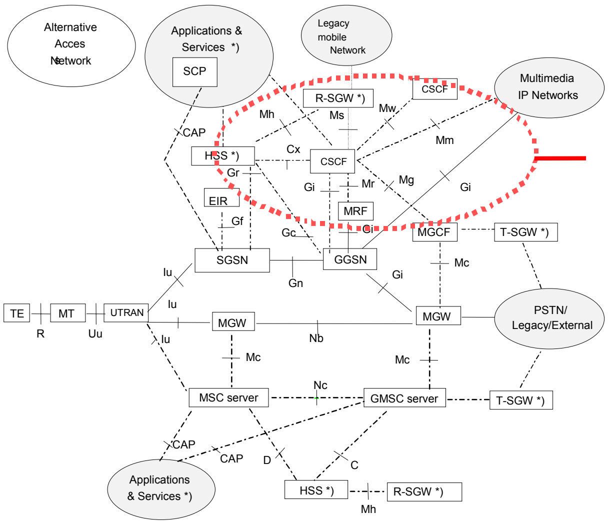
- User Agent (= UAC + UAS)
  - Can run services, such as forwarding, filtering etc.
  - Not always connected (out of coverage/battery etc.)
- Redirect Server
  - Can do services that require only Request-URI change, e.g. translation, parameter addition etc.
- Proxy Server
  - Can change certain headers 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

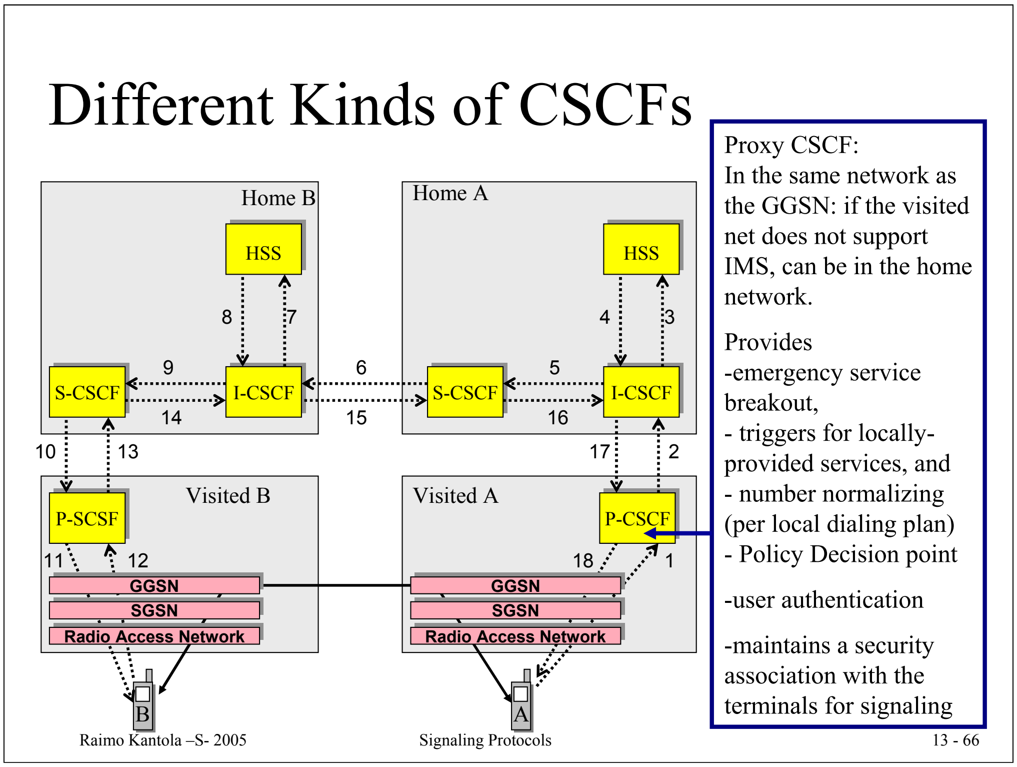
- Fuzzy Definition but has SIP+ interface!
- Can be a Redirect or Proxy Server or Back-to-Back UA
- The key is that it should be *programmable*
  - Routing based on service logic: what to do when user not registered or busy
  - URI translation: Reachability chains
  - Interfaces to other protocols: HTTP,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

3GPP Network Model (preliminary: …



CSCF (Call/Session Control Function) is the primary SIP node in the network.

(from www.sipforum.org)



Different Kinds of CSCFs

Proxy CSCF:
In the same network as the GGSN: if the visited net does not support IMS, can be in the home network.

Provides
-emergency service breakout,
- triggers for locally-provided services, and
- number normalizing (per local dialing plan)
- Policy Decision point

-user authentication

-maintains a security association with the terminals for signaling

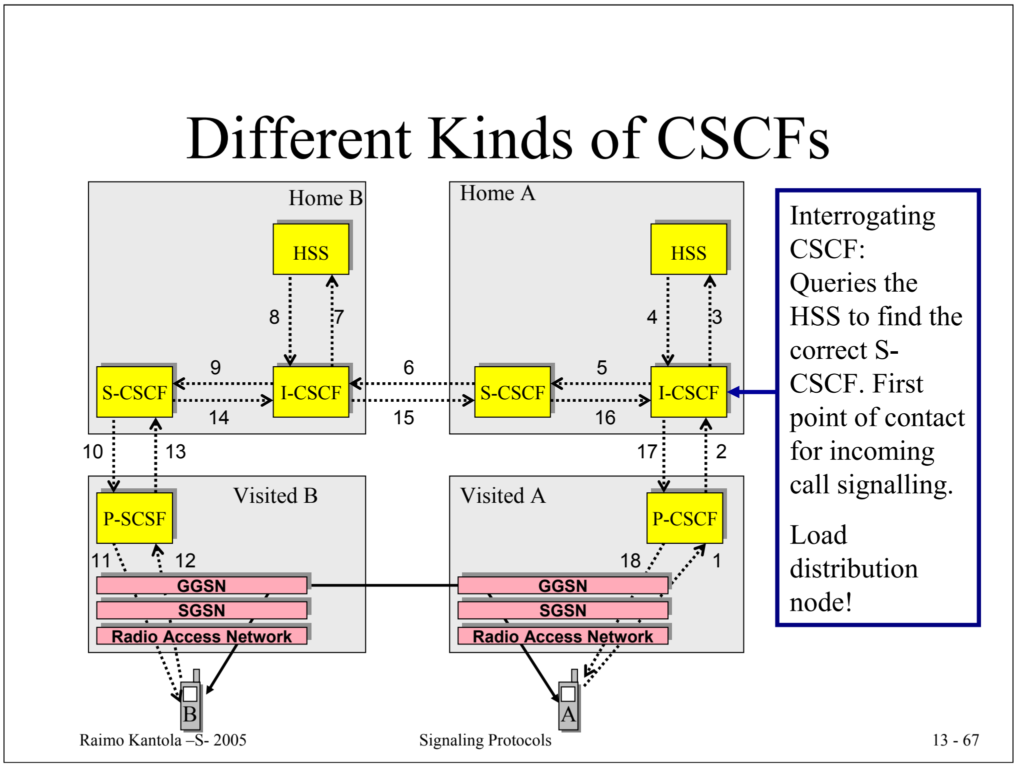
Raimo Kantola –S- 2005 Signaling Protocols 13 - 66

Currently, 3GPP has defined three different functional behaviors which the CSCF will exhibit.

The Proxy CSCF (P-CSCF) provides a first point of contact for the handset. All signaling to and from the handset goes through the P-CSCF. In terms of SIP, it behaves as an outbound proxy.

The main purpose for this node is to provide emergency service breakout and to do some basic message manipulation to enable the visited domain operator to provide locally sensitive services (e.g. traffic reports, directory services, etc). It also does simple number internationalization (which allows the support of local dialing plans).

It will probably also play a role in quality of service reservations.



Different Kinds of CSCFs

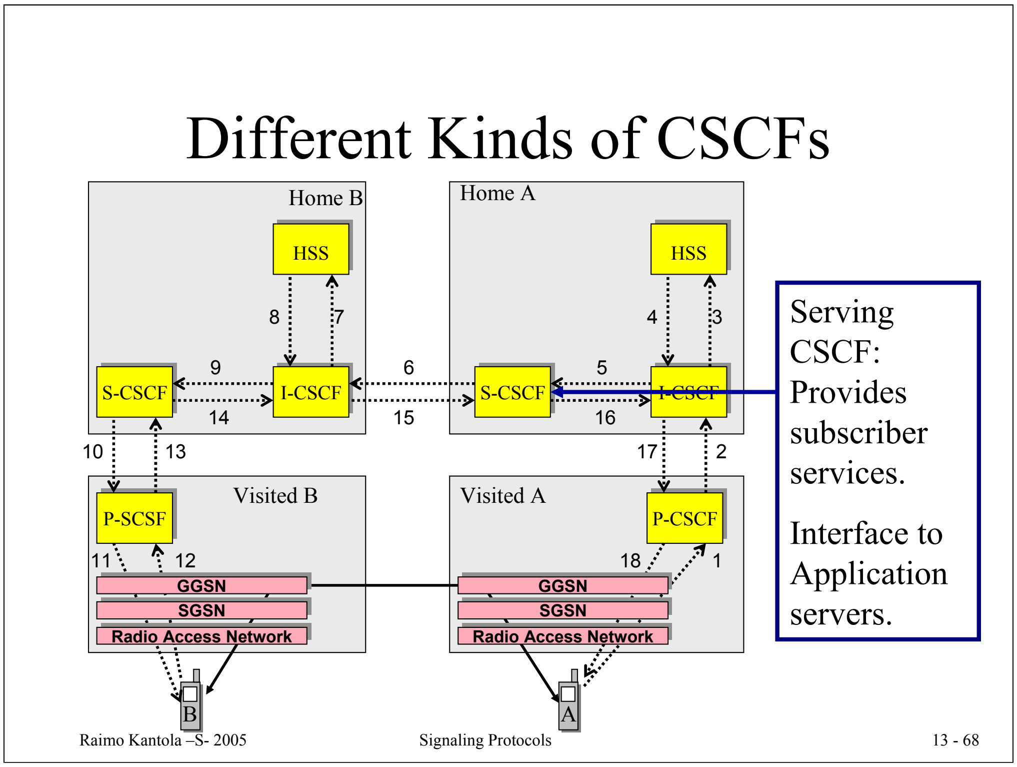
Interrogating CSCF: Queries the HSS to find the correct S-CSCF. First point of contact for incoming call signalling.

Load distribution node!

The Interrogating CSCF (I-CSCF) is mostly a load distribution node. Since DNS allows us simple statistical distribution among identical nodes, distributing load among the I-CSCFs is quite simple. But if all we relied on was statistical distribution, we wouldn’t be able to allocate subscriptions on appropriate serving nodes according to their capabilities, nor would we be assured of the ability to keep call state information between transactions.

So, the I-CSCF, in conjunction with the HSS, allocates subscription information onto appropriate Serving CSCFs. The HSS keeps track of this information so that all transactions and all calls for the same user go through the same service node.

The HSS stores user profile information; it’s somewhat similar to the HLR found in today’s cellular networks.



Different Kinds of CSCFs

Serving CSCF: Provides subscriber services.

Interface to Application servers.

Raimo Kantola –S- 2005 Signaling Protocols 13 - 68

The Serving CSCF (S-CSCF), quite simply, provides users services.

Of course, SIP allows the terminal to provide many services itself. The S-CSCF will be useful in providing, for example: call forwarding when the terminal is not available, call barring, centralized speed dial lists, VPN services, etc.

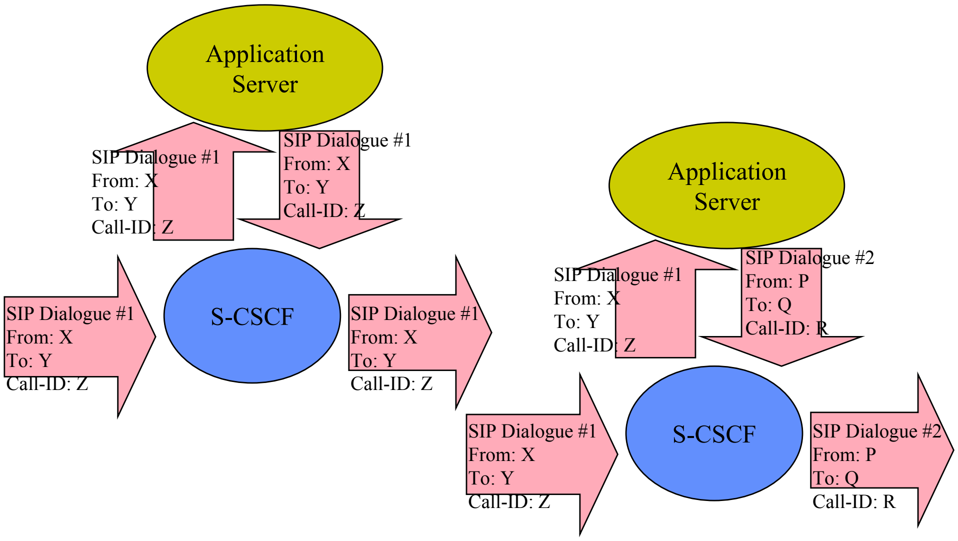
The interface to AS is based on iFC (initial Filter Criteria) and SIP. The interface also has a historic label ISC (IMS Service Control) and ”possibly SIP with some extensions”. In practice the protocol for AS communication is pure SIP. When a filter matches, the system finds the address of the AS that needs to get involved in proving the service.

The AS can a UA, a SIP proxy, a SIP redirect Server or a B2BUA = a collection of UAs with some service logic binding them together.

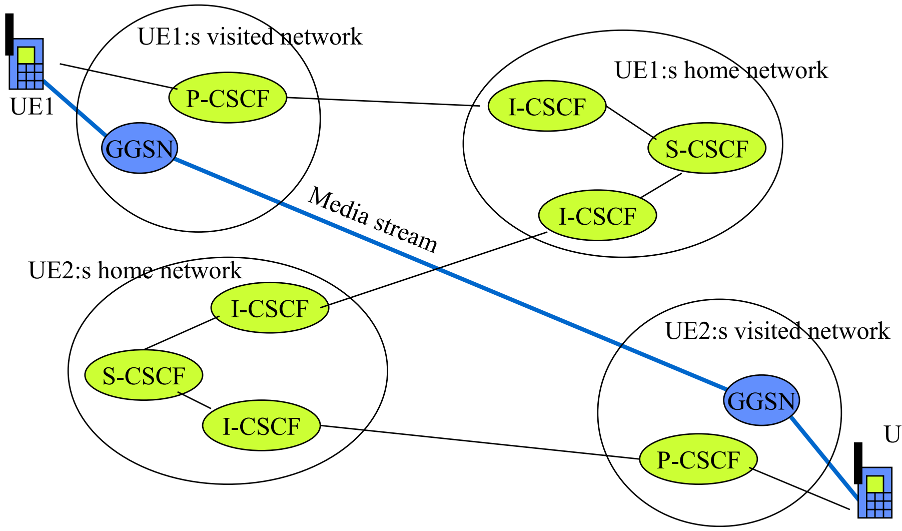
To route the call to the AS the S-CSCF creates a ROUTE header with two entries (or adds two new entries into the ROUTE header) containing the SIP URI of the AS and its own SIP URI in the second place. Based on the latter the AS will know that it needs to route the request back to the S-CSCF. The own SIP URI also contains some state info in the username part of the URI. When the request returns to the S-CSCF, it uses this state info to figure out where to continue the call processing.

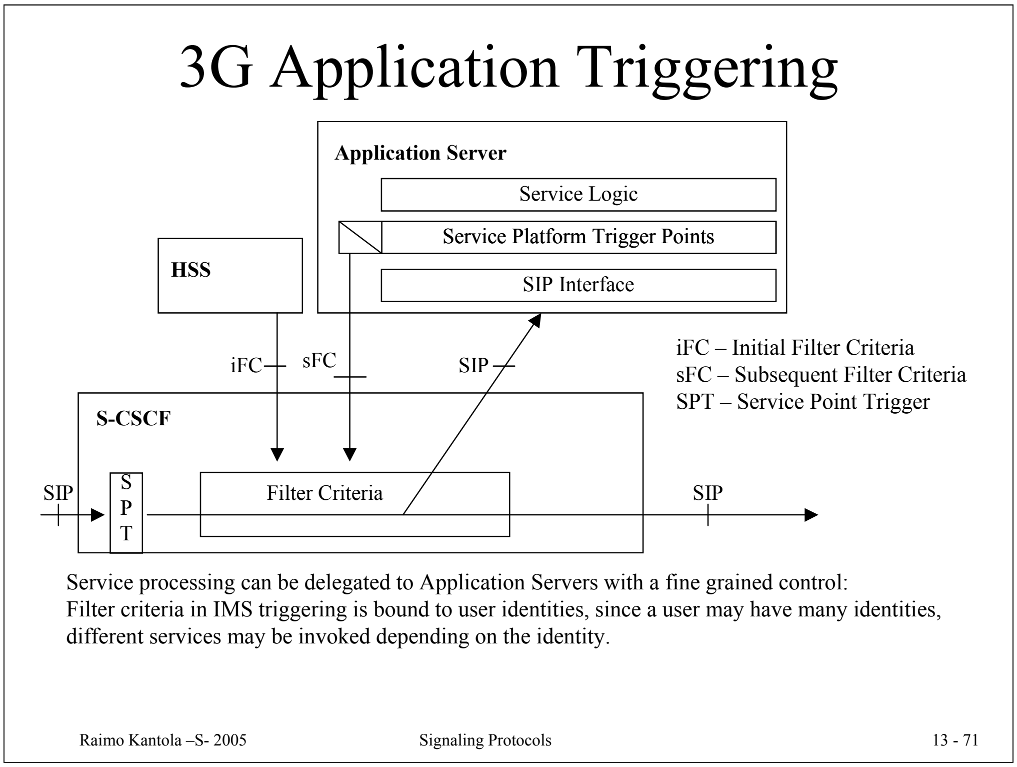
The AS may or may not decide to stay on the signaling path. To stay on the path, the AS places its SIP URI in the RECORD ROUTE header.

SIP Proxy vs B2BUA



Overview of routing between two mobile terminals





3G Application Triggering

Application Server

Service Logic

Service Platform Trigger Points

SIP Interface

HSS

iFC

sFC

SIP

S-CSCF

SIP

S P T

Filter Criteria

SIP

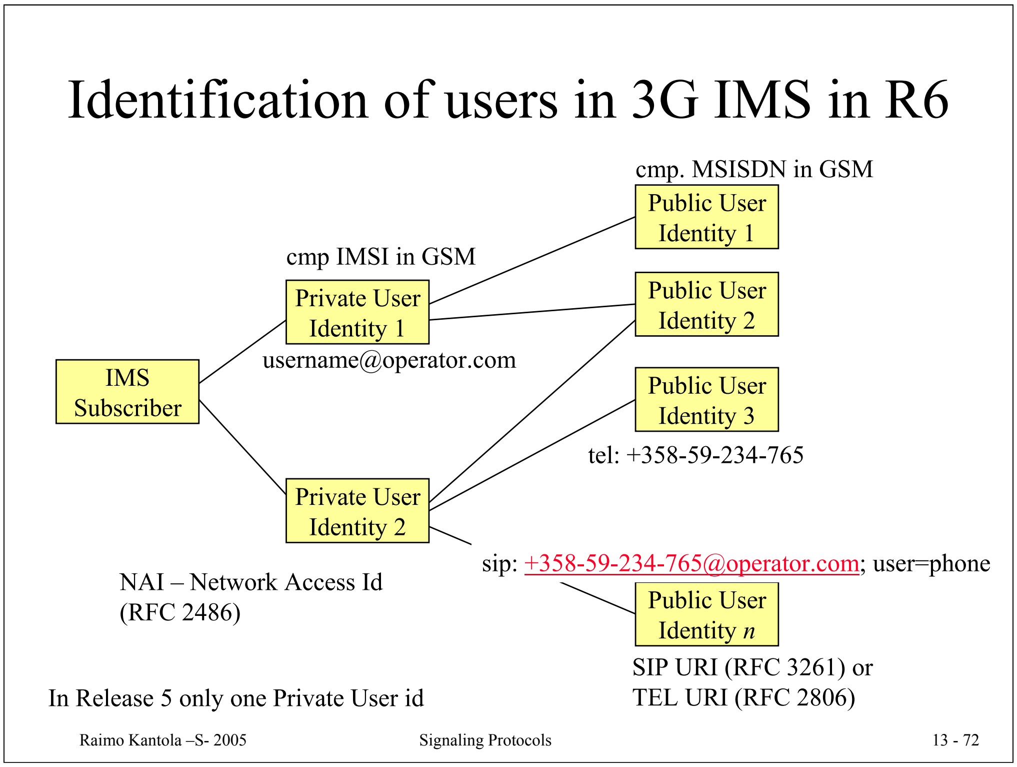
iFC – Initial Filter Criteria
sFC – Subsequent Filter Criteria
SPT – Service Point Trigger

Service processing can be delegated to Application Servers with a fine grained control:
Filter criteria in IMS triggering is bound to user identities, since a user may have many identities, different services may be invoked depending on the identity.

Raimo Kantola –S- 2005 Signaling Protocols 13 - 71

The originating IMS terminal sets the Preferred Identity for application triggering in the P-Preferred-Identity header field. The P-CSCF verifies that this a legal identity for the particular user within the current security association, changes the header field to P-Asserted-Identity with the value from the P-Preffered-Identity field. If the verification fails, the P-CSCF chooses to forward the default user identity in the P-Asserted-Identity field. If there was no P-Preferred-Identity in the INVITE, P-CSCF will insert the default user id into the P-Asserted-Identity header field.

Another factor that may serve as criteria for certain services is the type of access network (ADSL, WLAN, GERAN, UTRAN etc) in the P-Access-Network-Info. This information is carried only until the calling user’s home network and never forwarded into the callee’s home network for privacy reasons. The type of the access network gives an idea about the available capacity and pricing for the capacity.



Identification of users in 3G IMS in R6

IMS Subscriber

cmp IMSI in GSM

Private User Identity 1

username@operator.com

Private User Identity 2

cmp. MSISDN in GSM

Public User Identity 1

Public User Identity 2

Public User Identity 3

tel: +358-59-234-765

sip: +358-59-234-765@operator.com; user=phone

Public User Identity *n*

NAI – Network Access Id (RFC 2486)

SIP URI (RFC 3261) or TEL URI (RFC 2806)

In Release 5 only one Private User id

How to Program Services

- Call Processing Language
- SIP CGI
- SIP Servlets
- SIP JAIN (JSLEE – Jain Serv Logic Exec Env)
- Soft SSF and INAP/CAP
- Parlay
- OSA

=> Whatever… Different abstraction levels

==>
There will be many
competing ways to
implement services!

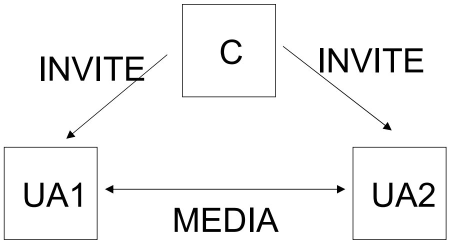
The claim is that it should be as open as flexible as creating services in the web these days

Server types for different services

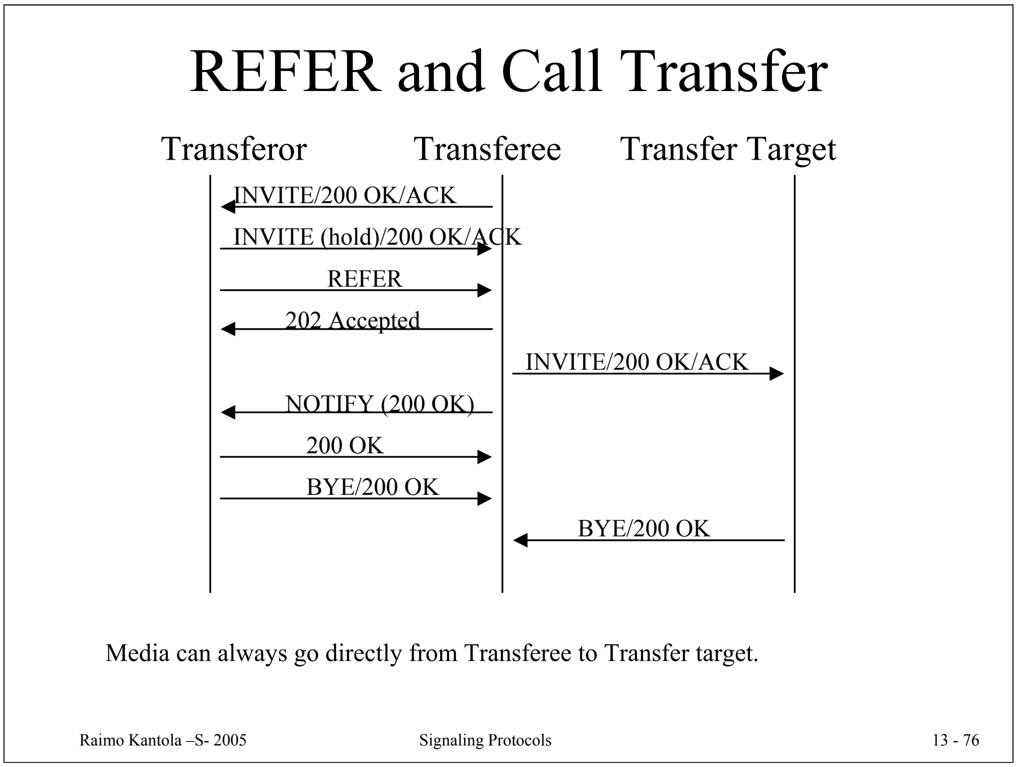
- Media Server (SIP, RTSP, HTTP)
  - Announcements, IVR, Voicemail, Media on demand
- Conferencing Server (SIP)
  - Media mixer
- Presence Server (SIP)
  - Users status info, capabilities, willingness to communicate
- Web Server (HTTP), E-mail Server (SMTP), Messaging Server (SIP?), Text-to-Speech Server etc.
- Controller Server
  - Co-ordinates the overall service

=> Server resources can be addressed by URLs, no need for tight coupling a la MGCP/Megaco

Third Party Call Control is based on SIP



- Details are still to be solved in the IETF
- Powerful tool e.g. for inviting users to centralized conferences or sessions with a Media Server
- In principle third party call control that has never been properly implemented in CSN, is as natural in SIP as first party call control because SIP is used also on the the interface to Application servers.



REFER and Call Transfer

Transferor Transferee Transfer Target

INVITE/200 OK/ACK

INVITE (hold)/200 OK/ACK

REFER

202 Accepted

INVITE/200 OK/ACK

NOTIFY (200 OK)

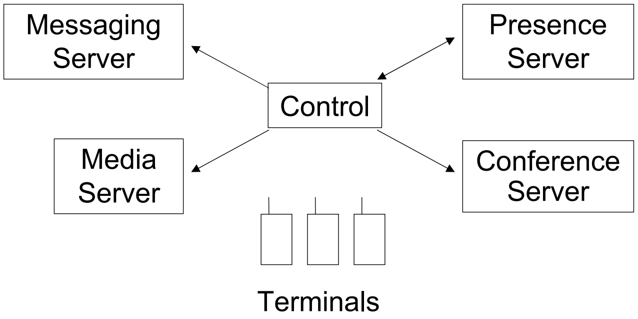
200 OK

BYE/200 OK

BYE/200 OK

Media can always go directly from Transferee to Transfer target.

Auto-conferencing Service Example

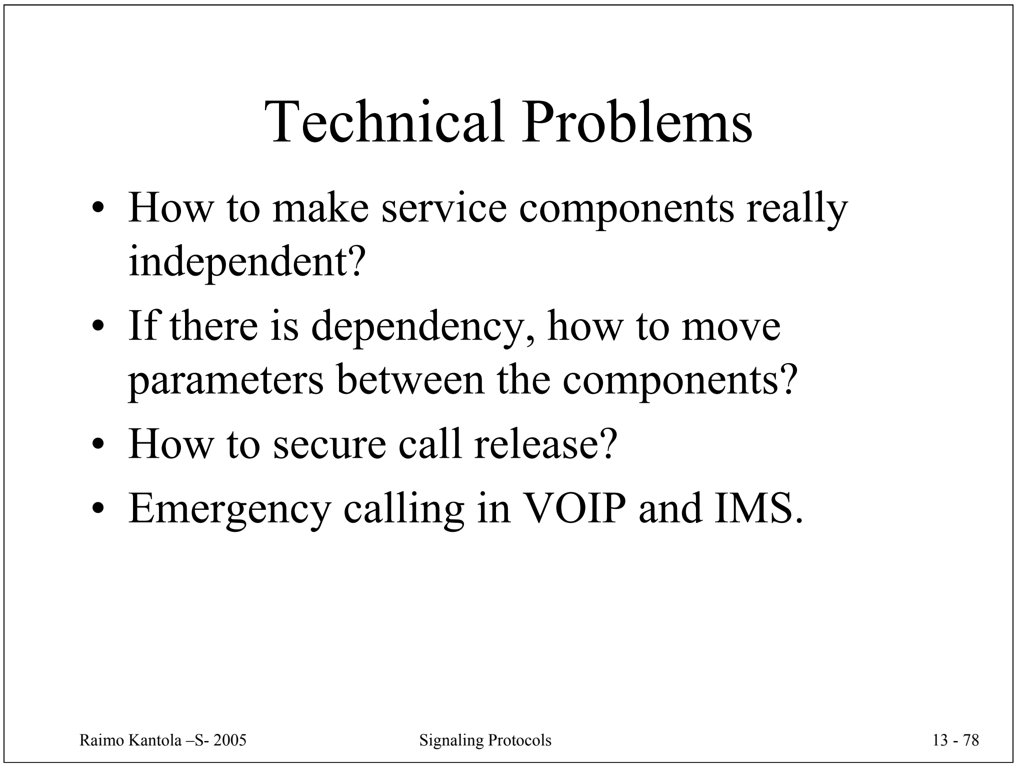


1. One user orders the conference by filling a web form

2. Controller subscribes to each participants presence

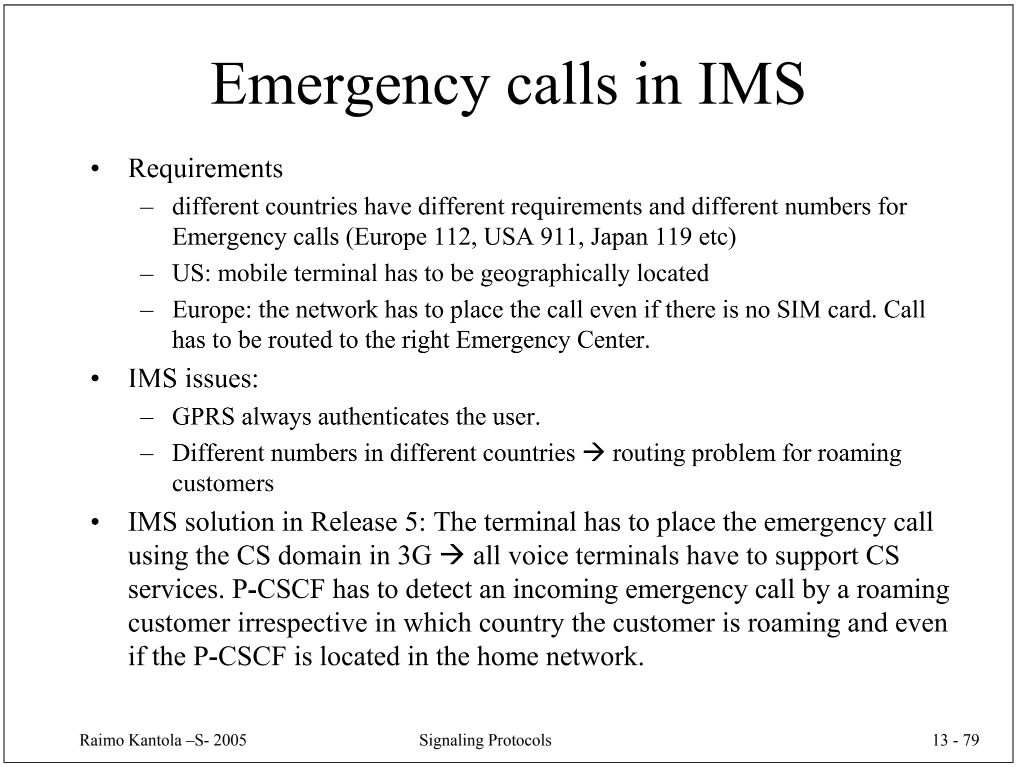
3. When all available, send message or start IVR session to each participant to confirm willingness

4. Connect each participant to conference server. Play announcements to conference from media server when new parties join



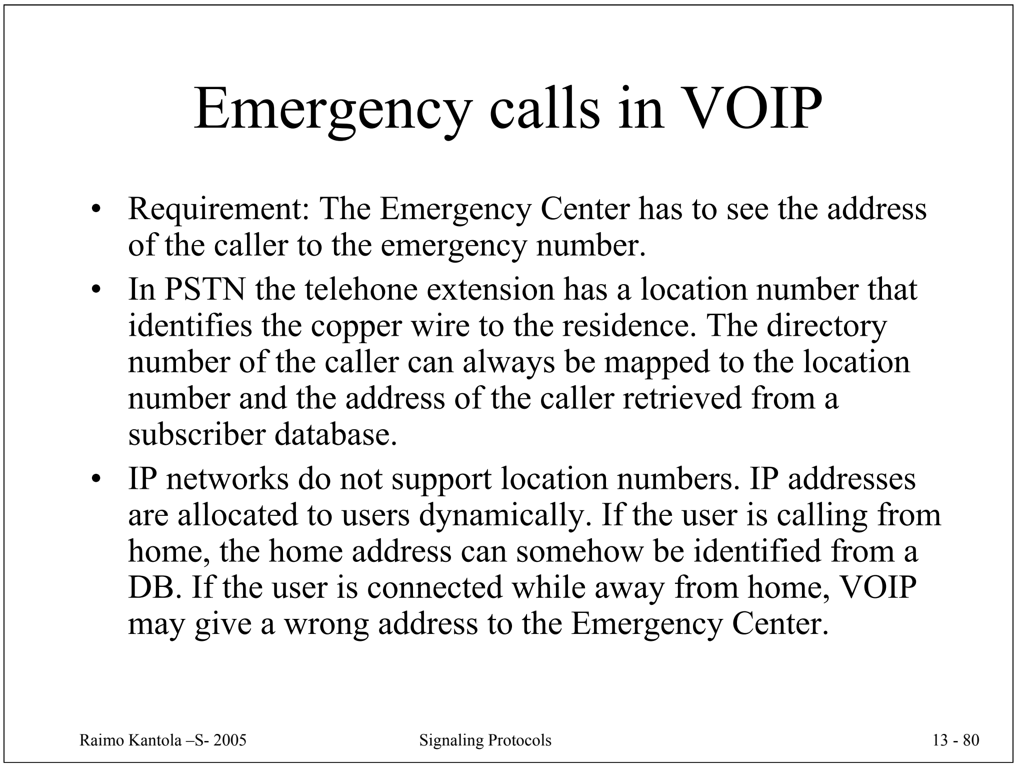
Technical Problems

- How to make service components really independent?
- If there is dependency, how to move parameters between the components?
- How to secure call release?
- Emergency calling in VOIP and IMS.



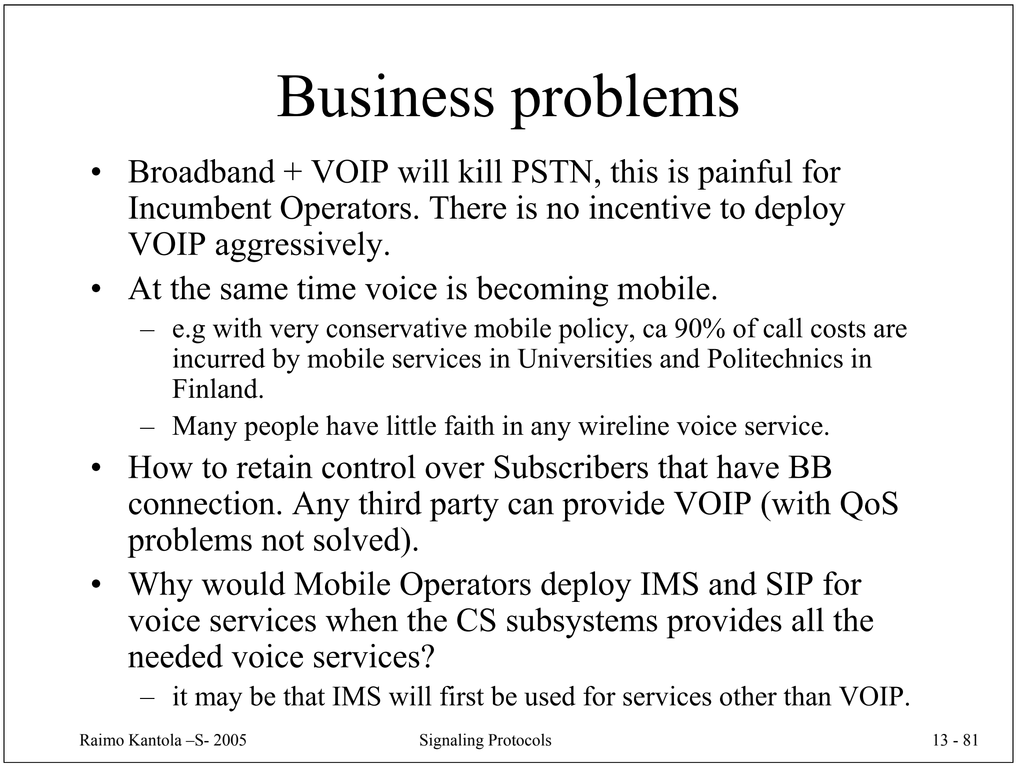
Emergency calls in IMS

- Requirements
  - different countries have different requirements and different numbers for Emergency calls (Europe 112, USA 911, Japan 119 etc)
  - US: mobile terminal has to be geographically located
  - Europe: the network has to place the call even if there is no SIM card. Call has to be routed to the right Emergency Center.
- IMS issues:
  - GPRS always authenticates the user.
  - Different numbers in different countries → routing problem for roaming customers
- IMS solution in Release 5: The terminal has to place the emergency call using the CS domain in 3G → all voice terminals have to support CS services. P-CSCF has to detect an incoming emergency call by a roaming customer irrespective in which country the customer is roaming and even if the P-CSCF is located in the home network.



Emergency calls in VOIP

- Requirement: The Emergency Center has to see the address of the caller to the emergency number.
- In PSTN the telehone extension has a location number that identifies the copper wire to the residence. The directory number of the caller can always be mapped to the location number and the address of the caller retrieved from a subscriber database.
- IP networks do not support location numbers. IP addresses are allocated to users dynamically. If the user is calling from home, the home address can somehow be identified from a DB. If the user is connected while away from home, VOIP may give a wrong address to the Emergency Center.



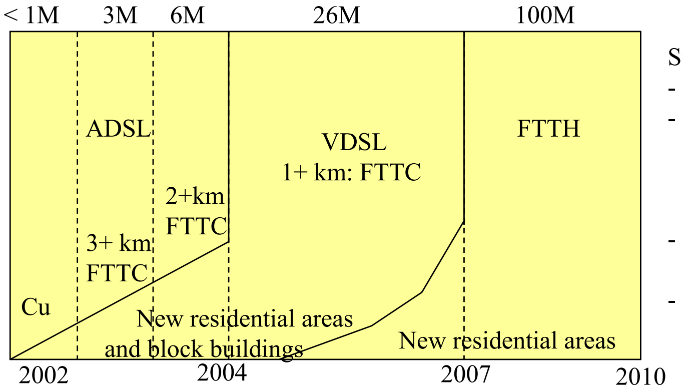
Business problems

- Broadband + VOIP will kill PSTN, this is painful for Incumbent Operators. There is no incentive to deploy VOIP aggressively.
- At the same time voice is becoming mobile.
  - e.g with very conservative mobile policy, ca 90% of call costs are incurred by mobile services in Universities and Politechnics in Finland.
  - Many people have little faith in any wireline voice service.
- How to retain control over Subscribers that have BB connection. Any third party can provide VOIP (with QoS problems not solved).
- Why would Mobile Operators deploy IMS and SIP for voice services when the CS subsystems provides all the needed voice services?
  - it may be that IMS will first be used for services other than VOIP.

Voting for VOIP

- Vendors have stopped developing CS telephony.
- BB deployment is proceeding: Examples of South-Korea, US.
- With wide spread BB, if operators do not deploy VOIP, someone will (e.g. SKYPE).

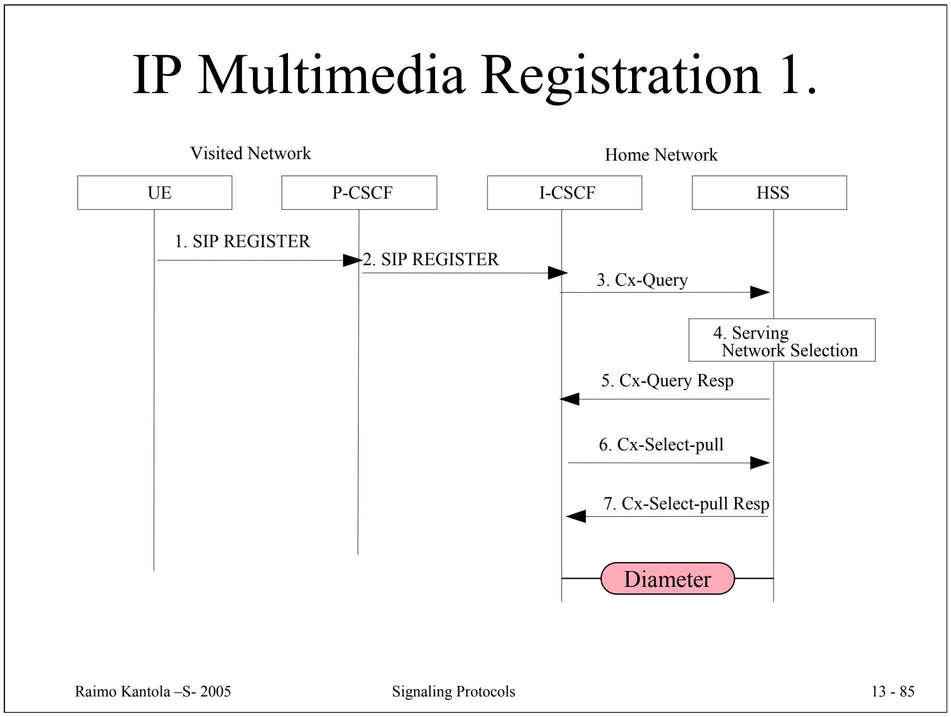
Broadband in South-Korea



Stats: 12/2002:
- > 70% homes have BB
- > 10M xDSL + CATV+ homes, containing
  6M xDSL + 3,7M CATV
  0,8 M VDSL
- 10% retail business in Internet
- 50% subscribers < 2km loop
  80% < 3 km, avg = 2,5km.

Source: Korea Telecom authors in IEEE Communications Magazine, Dec 2003

Appendix B – 3GPP IMS call flows



IP Multimedia Registration 1.

Visited Network

Home Network

UE

P-CSCF

I-CSCF

HSS

1. SIP REGISTER

2. SIP REGISTER

3. Cx-Query

4. Serving Network Selection

5. Cx-Query Resp

6. Cx-Select-pull

7. Cx-Select-pull Resp

Diameter

Raimo Kantola –S- 2005 Signaling Protocols 13 - 85

1. The UE has obtained a signalling channel through the access network, UE sends the Register information flow to the proxy (subscriber identity, home networks domain name).

2. The P-CSCF examines the “home domain name” to discover the entry point to the home network (I-CSCF).

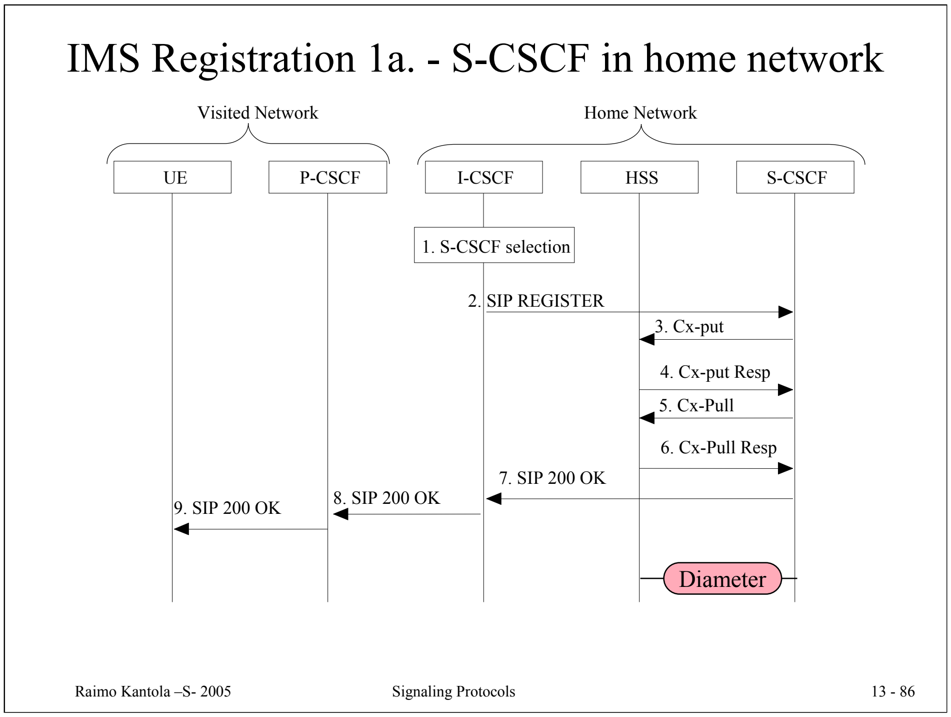
3. The I-CSCF shall send the Cx-Query information flow to the HSS (P-CSCF name, subscriber identity, home domain name, visited network capabilities, visited network contact name).

4. HSS selects whether the serving network is in the home network or the visited network

5. Indication of serving network selection is sent from the HSS to the I-CSCF

6. The I-CSCF sends Cx-Select-Pull (serving network indication, subscriber identity) to the HSS to request the information related to the required S-CSCF capabilities

7. The HSS sends Cx-Select-Pull Resp (required S-CSCF capbilities) to the I-CSCF.



IMS Registration 1a. - S-CSCF in home network

1. The I-CSCF determines the name of an appropriate S-CSCF.

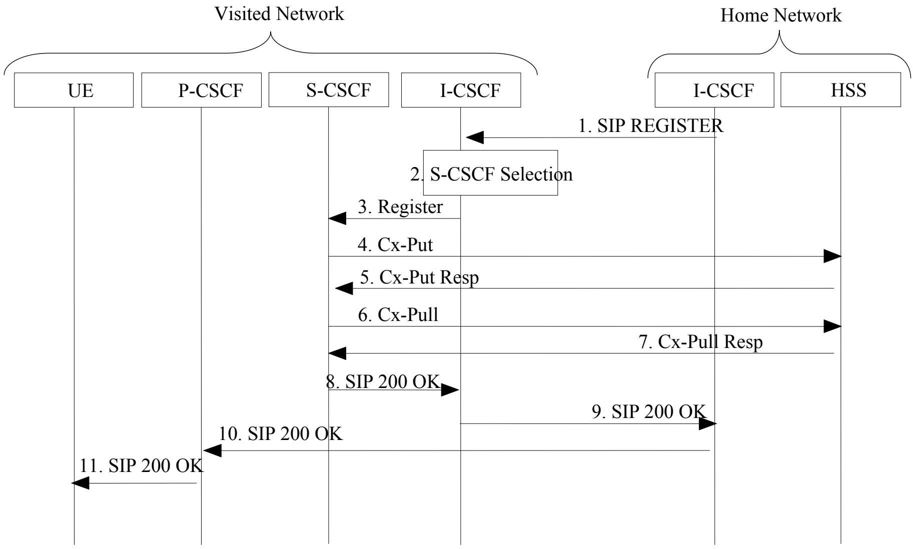
2. The I-CSCF sends the REGISTER (P-CSCFs “name” in the contact header, visited network capabilities, subscriber identity, visited network contact name) to the selected S-CSCF.

3,4. The S-CSCF sends Cx-Put (subscriber identity, S-CSCF name) to the HSS. The HSS stores the S-CSCF name for that subscriber. The HSS acknowledges Cx-Put.

5,6. The S-CSCF sends the Cx-Pull (subscriber identity) to the HSS in order to be able to download the relevant information from the subscriber profile to the S-CSCF. The S-CSCF stores the P-CSCFs name, as supplied by the visited network. The HSS returns the user information to the S-CSCF.

7,8,9. The S-CSCF returns 200 OK (serving network contact name, S-CSCF name) to the I-CSCF. The I-CSCF sends 200 OK (serving network contact name) to the P-CSCF. The P-CSCF stores the serving network contact name, and sends 200 OK to the UE.

IMS Registration 1b. - S-CSCF in visited network



Visited Network

Home Network

UE

P-CSCF

S-CSCF

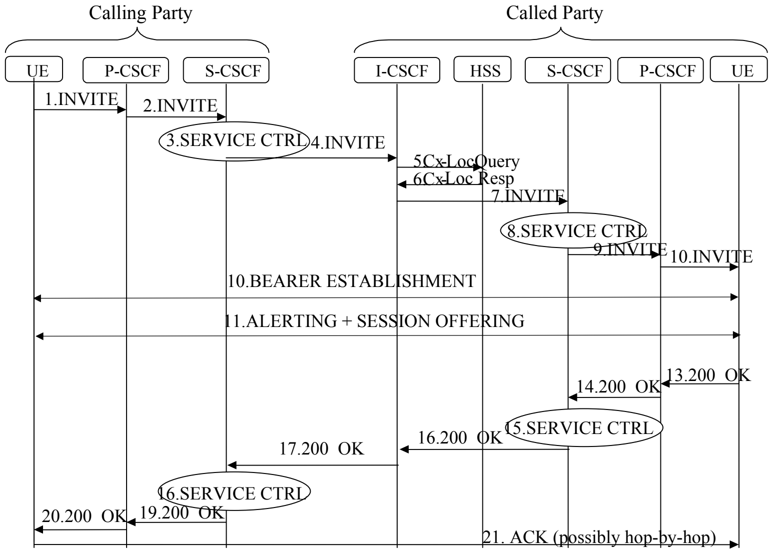
I-CSCF

I-CSCF

HSS

1. SIP REGISTER
2. S-CSCF Selection
3. Register
4. Cx-Put
5. Cx-Put Resp
6. Cx-Pull
7. Cx-Pull Resp
8. SIP 200 OK
9. SIP 200 OK
10. SIP 200 OK
11. SIP 200 OK

Mobile to Mobile Call



Raimo Kantola –S- 2005 Signaling Protocols 13 - 88

The UE sends a INVITE (session destination) to the P-CSCF.

The P-CSCF forwards the INVITE (session destination) to the next hop name/address. In this case the next hop address is the S-CSCF.

The S-CSCF can read the information on who originated the INV, and forwarded this based on the session destination. The S-CSCF forwards the invite message to the I-CSCF.

The I-CSCF sends ‘Cx-Location Query’ to the HSS to obtain the identity of the of the next hop which in this case is the S-CSCF.

The HSS sends the ‘Cx-Location Query Response’ to the I-CSCF.

The I-CSCF forwards the INVITE message to the S-CSCF for the called party.

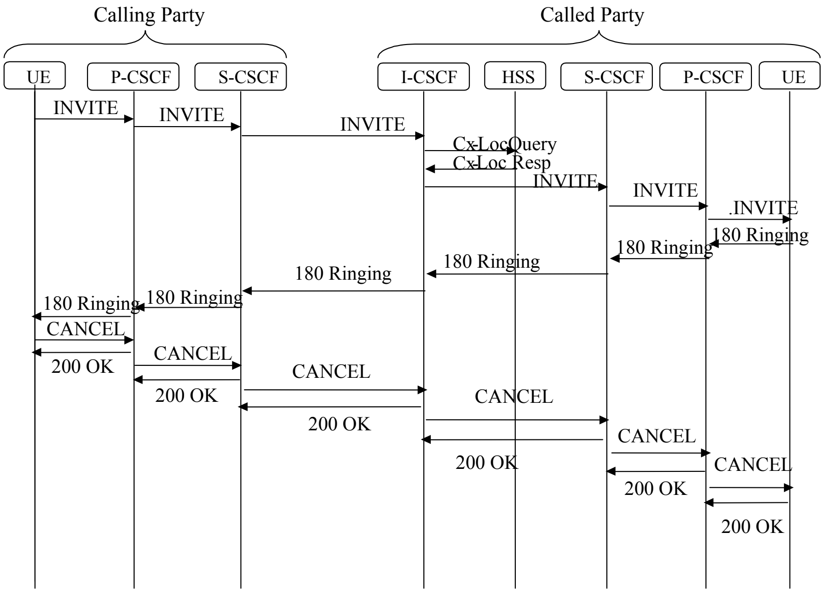
The S-CSCF carries out Service Control for Called Party. This includes applying the filter criteria to the incoming call and retargeting the the INVITE request. The latter means modifying the request-URI. The original URI is placed in the P-Called-Party-ID header field so the callee terminal will know to which of its public identities the call was addressed. Retargeting is necessary because the called party may be moving and thus keeping the original request-URI might create a loop.

The S-CSCF forwards the INVITE message to the P-CSCF (Called Party)

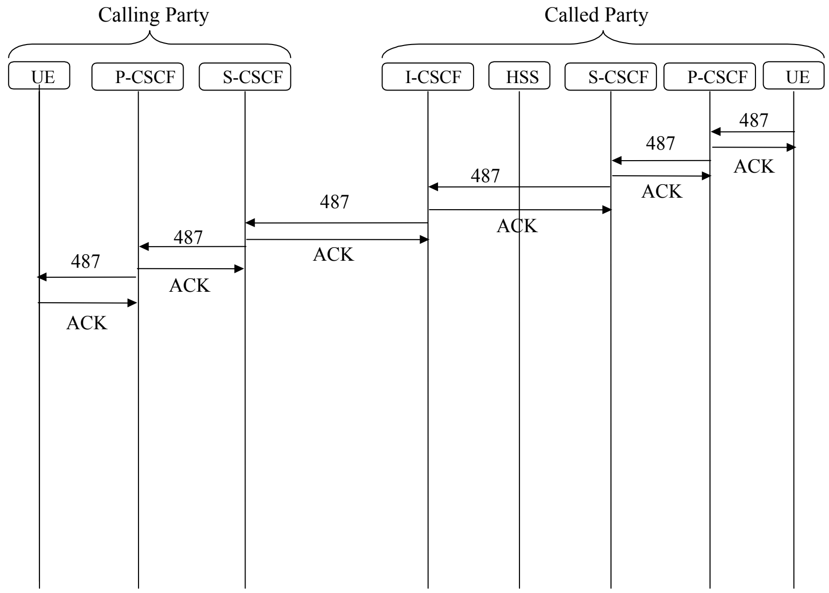
The P-CSCF forwards the INVITE message to the UE (Called Party)

Bearer Establishment Process: Bearer reservation (temp establishment) which will

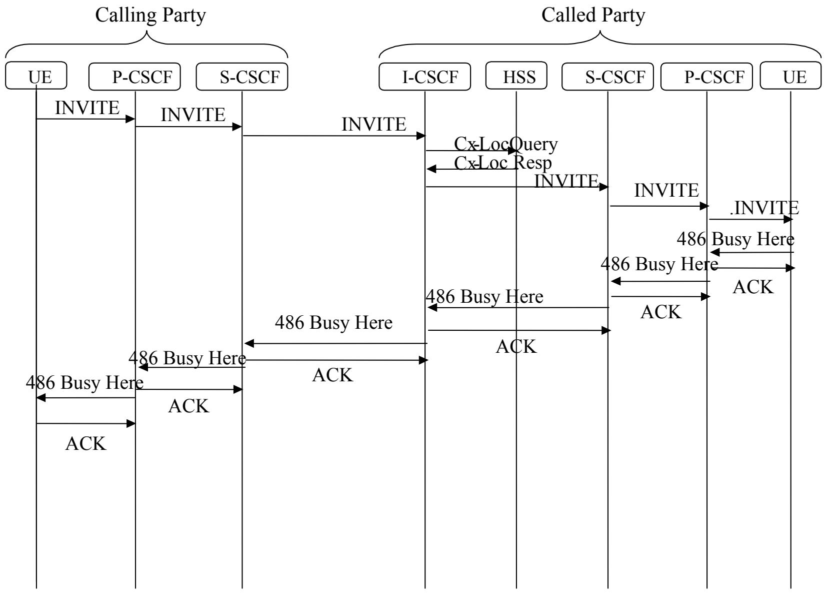
Call flow examples 1. - no answer



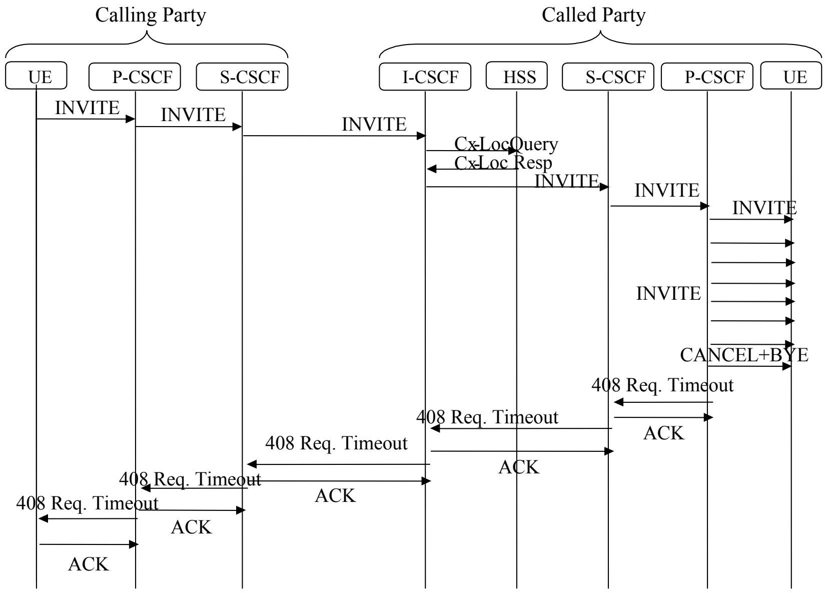
Call flow examples 1. - no answer 2.



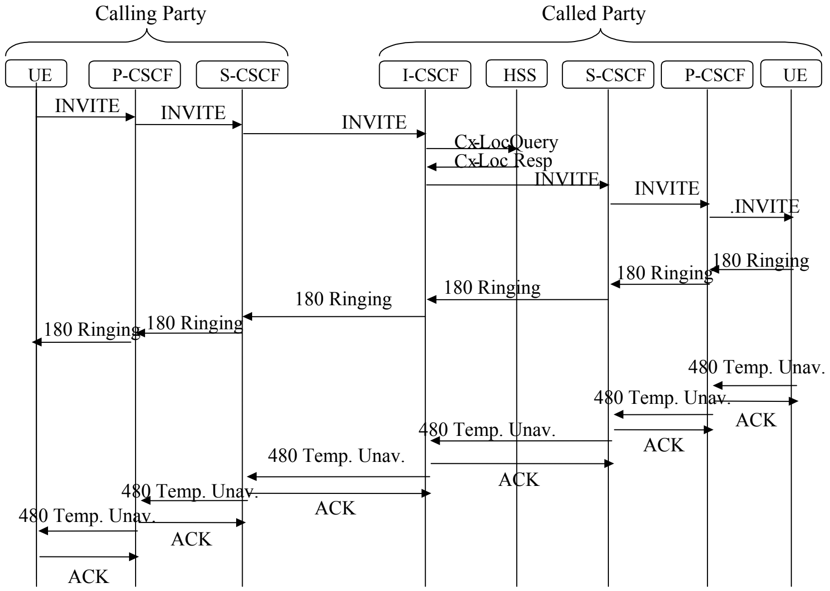
Call flow examples 2. - busy



Call flow examples 3. - no response



Call flow examples 4. - temporarily unavailable


Calling Party
Called Party
UE
P-CSCF
S-CSCF
I-CSCF
HSS
S-CSCF
P-CSCF
UE
INVITE
INVITE
INVITE
Cx-LocQuery
Cx-Loc Resp
INVITE
INVITE
.INVITE
180 Ringing
180 Ringing
180 Ringing
180 Ringing
180 Ringing
180 Ringing
480 Temp. Unav.
480 Temp. Unav.
ACK
480 Temp. Unav.
ACK
480 Temp. Unav.
ACK
480 Temp. Unav.
ACK
480 Temp. Unav.
ACK
ACK