SIP Demystified

Book Overview

Content

- Signalling in CS Network
- Packet Switching, IETF
- The Internet Multimedia Conferencing Architecture
- Session Initiation Protocol - SIP
- SIP – Protocol Operation
- Extending SIP – The SIP Toolkit
- Bulding Applications with SIP Toolkit
- Appendix A – Call Flows examples
- Appendix B – 3GPP IMS Call Flows Examples

Signalling in CS Network

- CS
  - Strenghts: Fast, Small Latency, QoS guaranteed
  - Weaknesses: Path establishment, occupies resources 100% of time, intelligence in network needed, resillience
- Signalling – to setup, control and tear-down sessions
  - Evolution: DC and AC analog->In-band (FDM transport)->Digital (TDM transport)
    - Access signalling (between terminals and exchange)
      - DTMF
      - Pulse
      - DSS-1 (Digital Subscriber Line No 1) (ISDN, GSM)
    - Trunk Signalling (Between exchanges)
      - CAS (Channel Associated Signalling)
      - CCS (Common Channel Signalling)

CAS and CCS

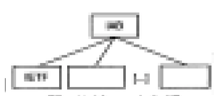
- CAS
    - 16th channel in PCM link
    - Some signalling in voice channels (Caller ID, Callee ID, ...)
- CCS
    - Separate, dedicated signalling network exist
    - One signalling channel handles thousands voice timeslots
    - SS7 dominant CCS signalling
    - Services: toll-free calls, etc
  - SS7
    - Circuit-related
      - E.g. ISUP (Basic Services, circuit management, supplementary services)
      - ISUP has national variations – gateway needed
      - Problem: Intelligence in network, dumb terminals
      - Hard to introduce new services (not flexible architecture)

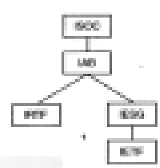
Switching, IP, IETF

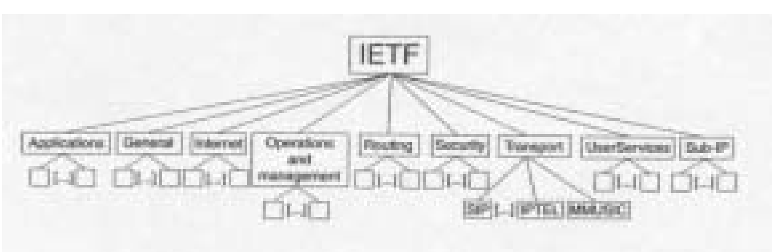
- Packet Switching (efficient, cheaper, more delay, overhead)
  - Datagrams
    - Routing based on destination address
    - No state in network
    - Dynamic routing, load balancing
  - Virtual circuits
    - Virtual circuit established
    - Stateful network
- IP – dominant packet network technology
  - IP layer common
    - Various underlying technologies (ATM, Ethernet, Frame Relay, ...)
    - Various applications above (email, web, VoIP) using common IP layer
  - Intelligence on the edge of the network (centrifugism)
    - Entities on edge provide and consume services
    - Stateless and fast core entities
    - End-to-end services, end-to-end security, etc...

IETF

- IETF toolkit
  - bottom-up approach *("one problem – one protocol")*
  - Reusable protocols
  - Protocols are simple, reusable, scalable, rob

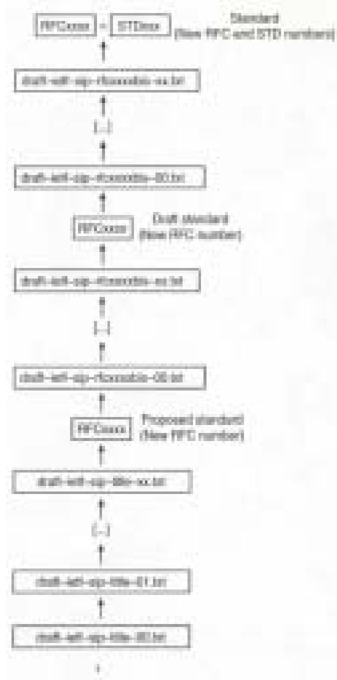






IETF specifications

•Every standard follows the route Proposed standard-> Draft Standard-> Standard



What Protocols Are Needed?

- <u>Signaling protocol</u> to establish presence, locate user, and for session control
- <u>Media Transport Protocols</u> for transmission of media over IP
- <u>Supporting Protocols</u>
  - Gateway location, QoS, AAA,etc…

Multicast

- Several parties involved
  - IPv4 Multicast from 224.0.0.0 – 239.255.255.255
- Saves bandwidth
- Entity that is sending does not have to know all the participants
- Multicast Routing protocols
  - Dense Mode (shortest-path tree per sender)
  - Sparse Mode (shared tree used by all sources)
- IGMP (Internet Group Management Protocol)
  - For hosts that want to become part of multicast group
- Mbone – part of Internet that supports multicast
- RTP – transport of real-time data
  - Sequence number, timestamps
- RTCP – controls RTP transport (every RTP session has paralell RTCP ses.)

QoS – Integrated Serv. and DiffServ

- Integrated Services
  - Different treatment to different flows
  - State info stored in network, routers examine packets!!!(not good)
  - Reservation merging
  - RSVP protocol – for reservation of resources
- DiffServ
  - Defines several traffic classes with different priority levels
  - Packets tagged with level tags at the beginning
  - Routers just examine tags
  - Better scaling

Other Protocols

- SAP (Session Announcement Protocol)
  - Distribute info about multicast sessions on a well-known address and port
- SDP (Session Description Protocol)
  - Describes session, text-based
  - E.g. time, media used, codec used, port used, subject, etc…)
  - Extensible (*a* line gives extension, e.g. *a=volume:8*)
- SDPng (next generation)
- RTSP (Real-Time Streaming Protocol)
  - Simmilar to VCR controls (stop, pause,play, record)
- RSVP – ReSerVation Protocol
  - For resource reservation in the network

Session Initiation Protocol - SIP

- Originally designed to invite users to Mbone sessions
- In IETF:
  - SIP WG – for SIP specifications and extensions
  - SIPPING WG – for applications that use SIP
- Was RFC2543, now RFC 3261!
- For <u>session establishment</u>, <u>modification</u> and <u>termination</u>
- Independent of media session and on mechanism for describing session
- Used to distribute SDP among potential participants
- Reusable addresses: SIP addresses similar to email addresses
  - E.g. sip:someone@somewhere.com

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
    - Group addresses
  - Registrars
    - Accepts registrations
  - 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)

SIP Features

- Part of IETF toolkit
  - Reusing other protocols & mechanisms
  - 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 reusage
- Same routing as SMTP
- Reuses infrastructure (all applications will use SIP entities for different services)

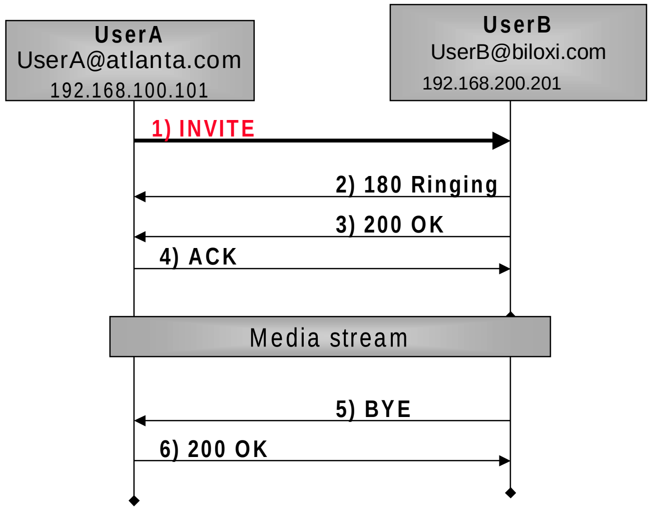
Addressing

- **sip:user@host[parameters][headers]**
- SIP-addresses 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… is easy implemented

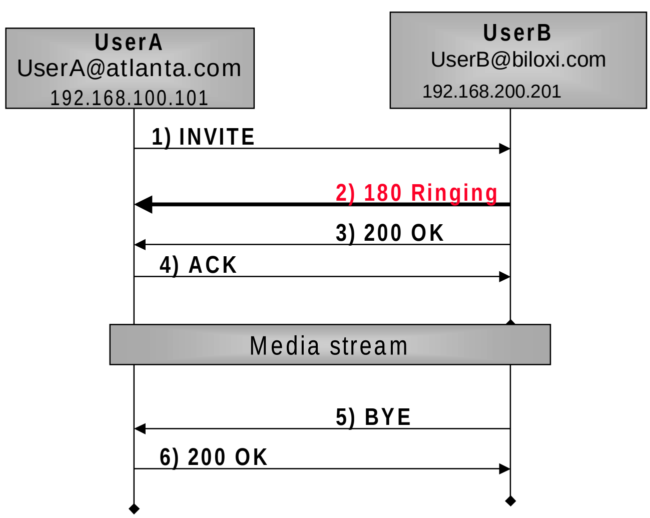
”Basic call” Example

- 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 !

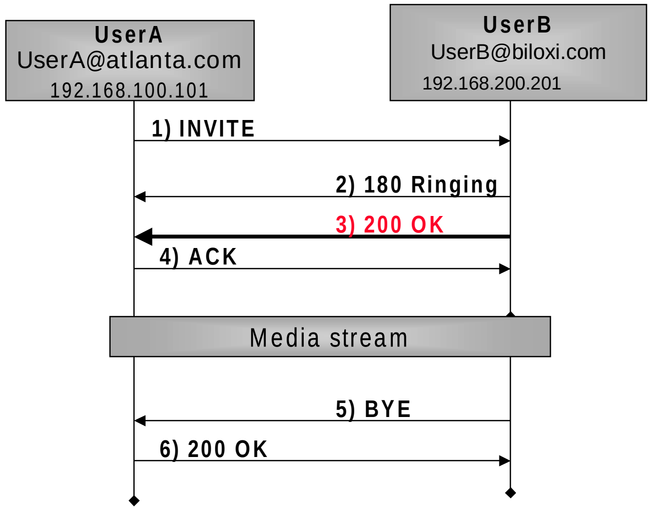
"Basic Call" call flow



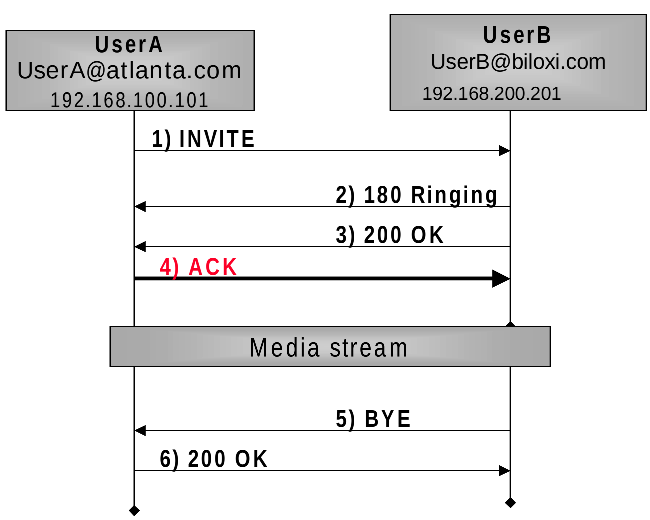
"Basic Call" call flow



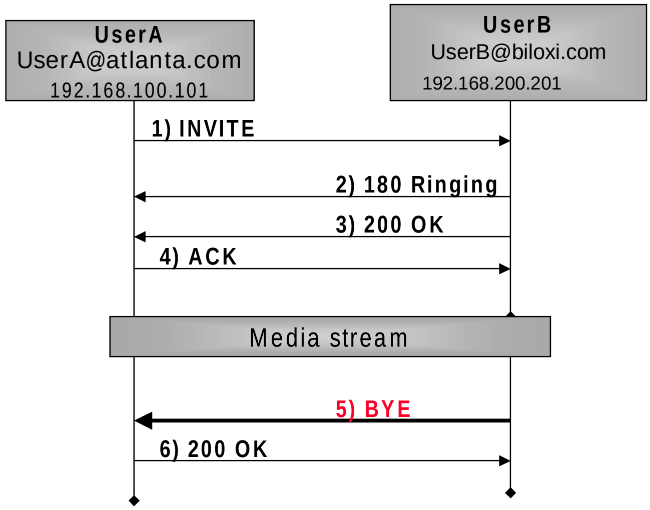
"Basic Call" call flow



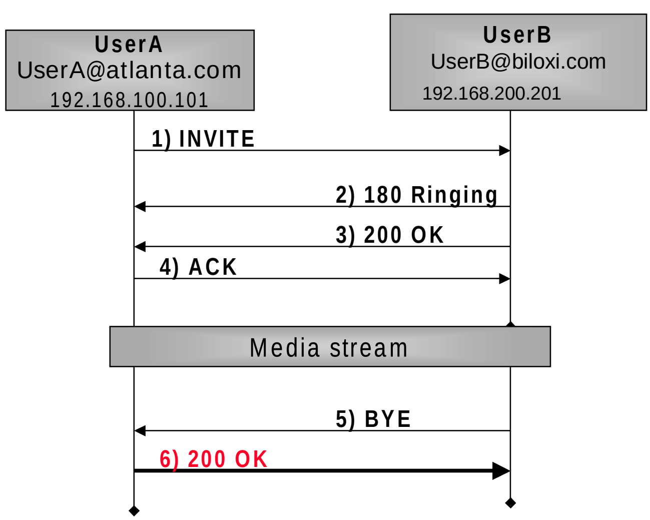
"Basic Call" call flow



"Basic Call" call flow



"Basic Call" call flow



SIP methods (requests)

- SIP methods are invoked on servers when requests arrive:
  - REGISTER requests 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

- 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

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
```

Start line

Headers

Body

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
```

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

- 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 come, with all Record-route headers in it (then UAC knows which relays want to stay in signalling**
- **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)

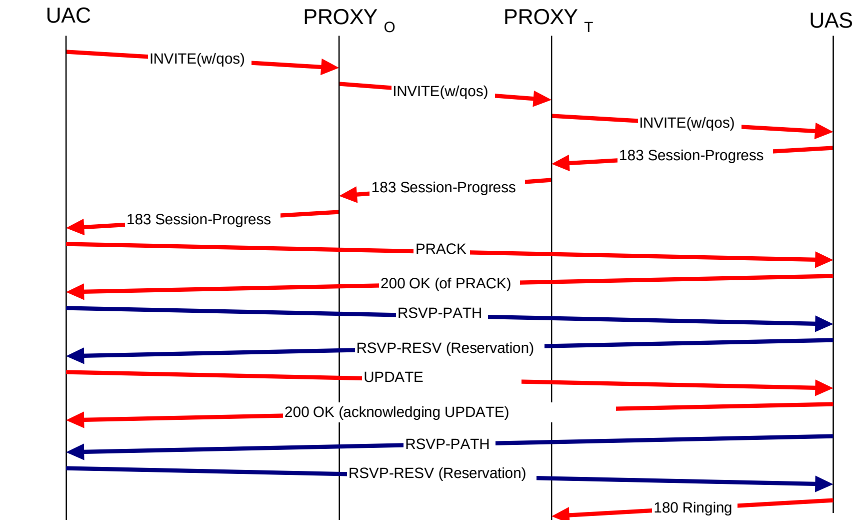
Feature Negotiation (OPTIONS)

- *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

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
- Additional Status Response - 580 Precondition Failure
  - If a mandatory precondition can’t be met, UAS terminates INVITE with this status response

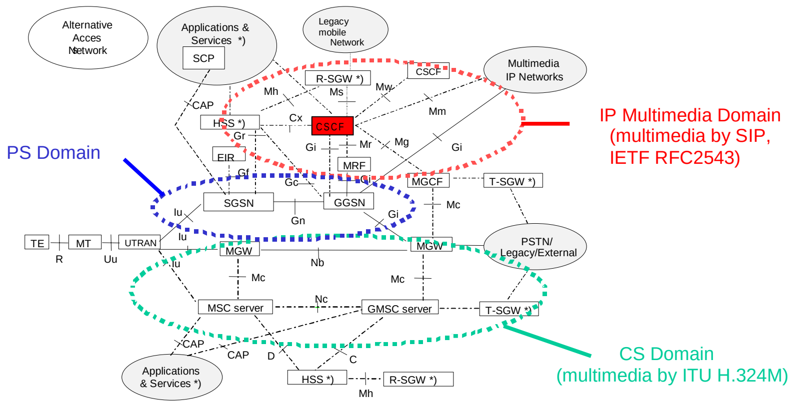
Sample Invite Sequence, with Resource Reservation



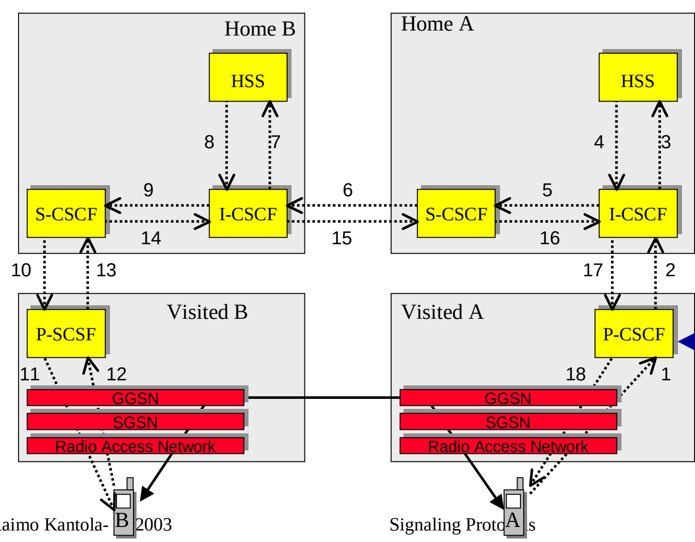
More SIP extensions

- MESSAGE
  - For instant messaging
- INFO
  - To transport mid-session information (very useful in SIP-PSTN gateways)
- Automatic configuration
  - DHCP or Service Location Protocol (SLP)
- Caller Preferences
  - New headers: Accept-Contact, Reject-Contact, Request-Disposition
- SUBSCRIBE/NOTIFY
  - Two new methods for async. notifications. Used in Presence concept
- REFER
  - For session transfer (Refer-To: and Reffered-By:)
- DO (for sending commands to appliances)

3GPP Network Model

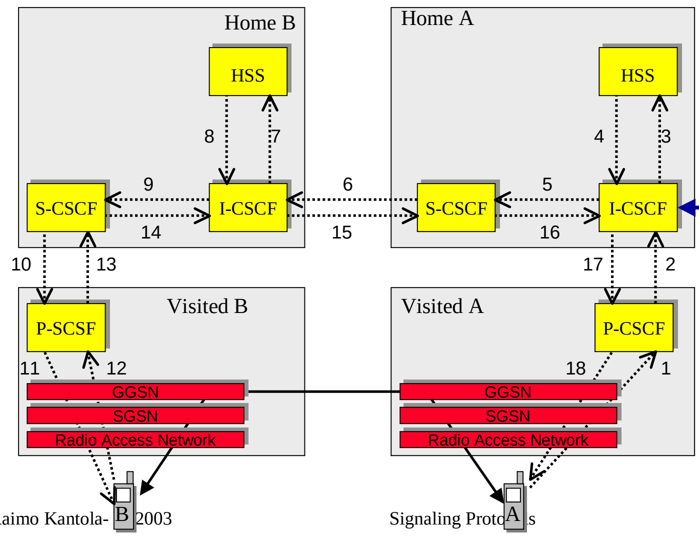


Different Kinds of CSCFs



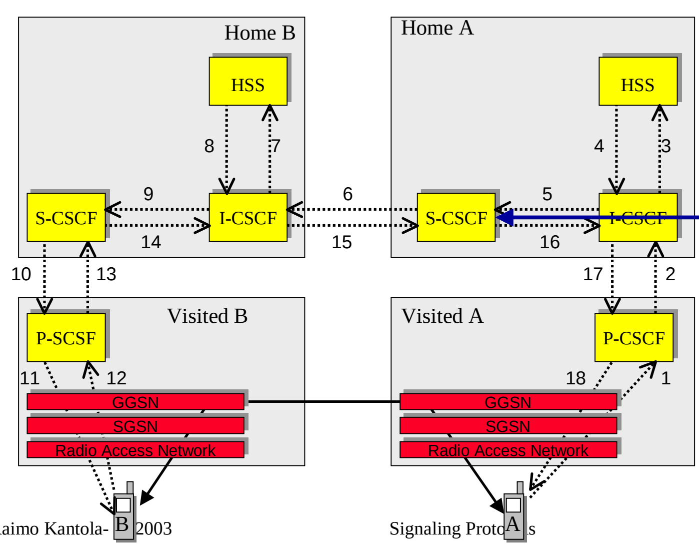
Proxy CSCF: Provides emergency service breakout, triggers for locally-provided services, and number normalizing (per local dialing plan)

Different Kinds of CSCFs



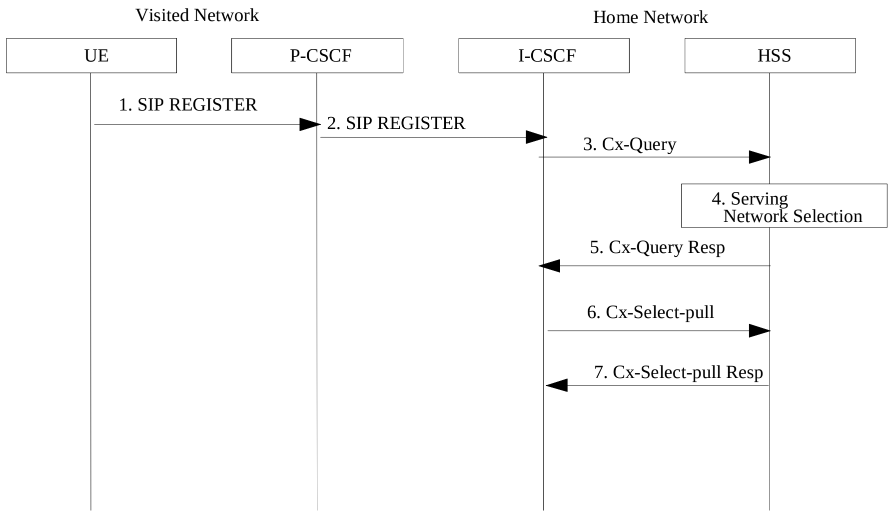
Interrogating CSCF: Queries the HSS to find the correct S-CSCF. First point of contact for incoming call signalling.

Different Kinds of CSCFs



Serving CSCF: Provides subscriber services.

IP Multimedia Registration 1.



SIP and new service architectures

- **Call Processing Language**
  - XML based language to define call service
  - CPL definition is now simple and can be used to describe basic services
- **SIP/CGI**
  - SIP Application Server could implement a CGI interface to describe services
- **SIP Servlet API**
  - SIP Servlet Application Server is comparable in principal to CGI
  - Advantage: use of the java tools and security, more performant than CGI

SIP and new service architectures 2.

- **Java Enhanced SIP (JES)**
  - Defines a way to transport Java applets, objects or agents in a SIP message
  - It could be used to send service logic to a SIP client
  - The Java entity can be embedded in the message if small enough or referenced for later downloading using HTTP or SOAP
- **SIP and WEB/WAP Integration**
  - SIP is working well in conjunction with Web (MIME, URI, DNS). Lot of SIP services could be originated by Web pages (click to call, Web phone book…)
  - SIP and HTTP stacks have a lot in common. Advanced SIP implementation usually provides simple HTTP stack

SIP and new service architectures 3.

- **SIP and WEB/WAP Integration cont.**
  - SIP and other IP services will probably come to 3G also and they will replace WAP services if WAP does not evolve
  - WML could be carried inside SIP messages. WML has effective syntax for menus and SIP is the one and only transport which also include intelligent features.
- **SIP and SOAP**
  - SOAP can be used in conjunction with SIP to offer services. SOAP can be used by a SIP client to discover and access services on the network. SIP application server can use SOAP to access Business Logic and databases in the network.

Appendix A – Call Setup Examples

Registration example with SIP

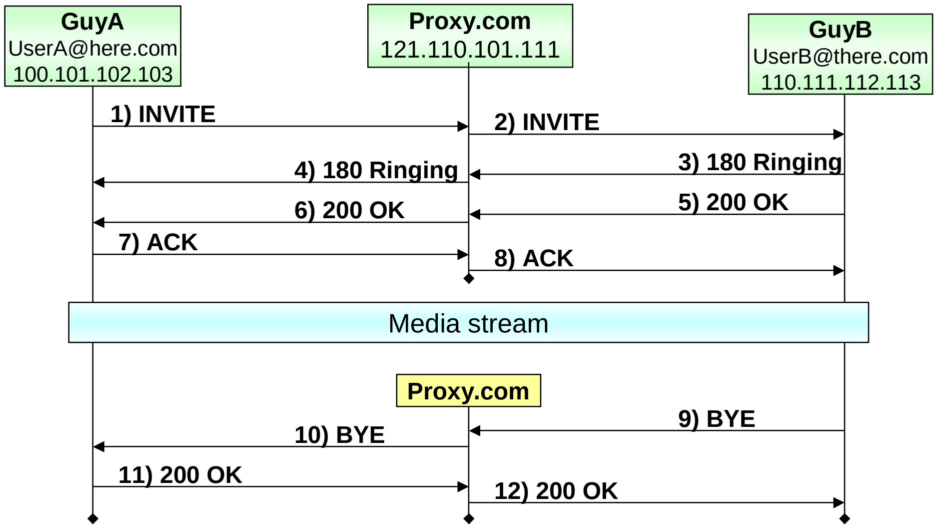
Bob
bob@biloxi.com

biloxi.com

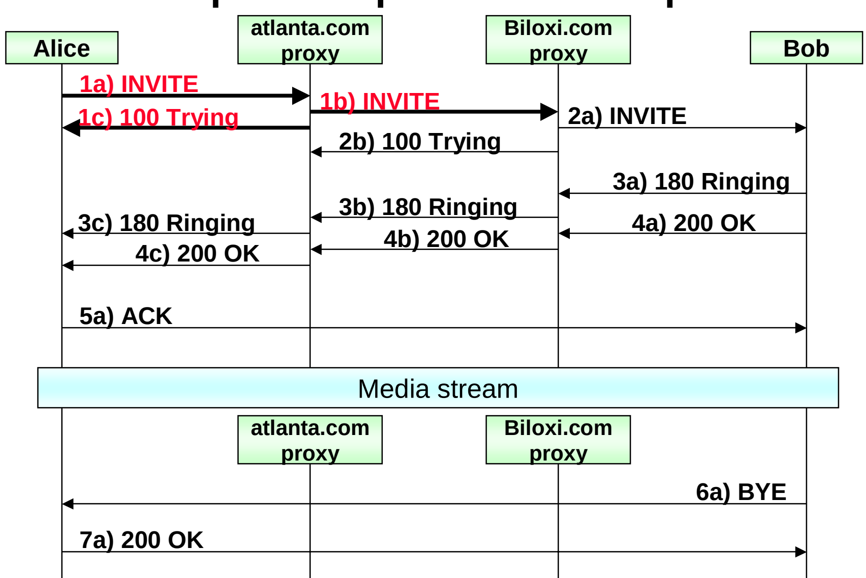
REGISTER sip:registrar.biloxi.com SIP/2.0
Via: SIP/2.0/UDP bobspc.biloxi.com:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Bob <sip:bob@biloxi.com>;tag=456248
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 REGISTER
Contact: <sip:bob@192.0.2.4>
Expires: 7200
Content-Length: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

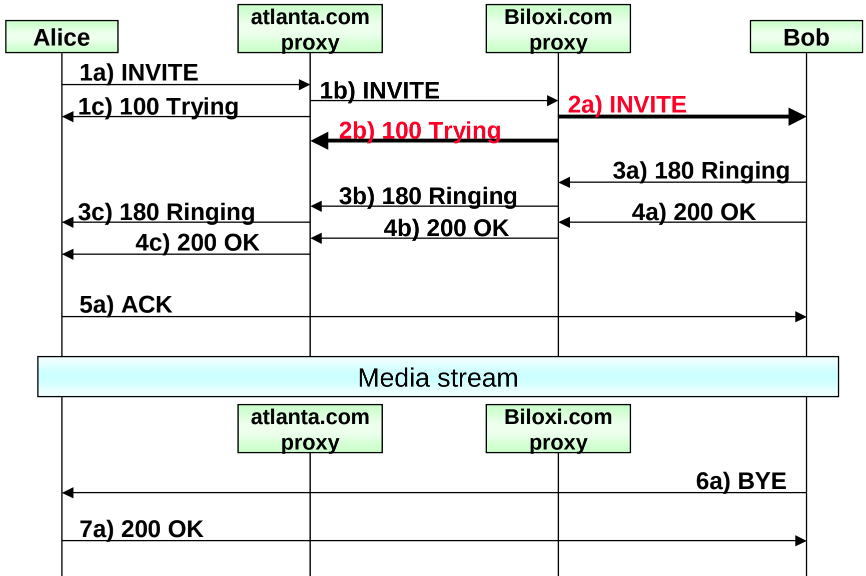
Call Setup example with one proxy



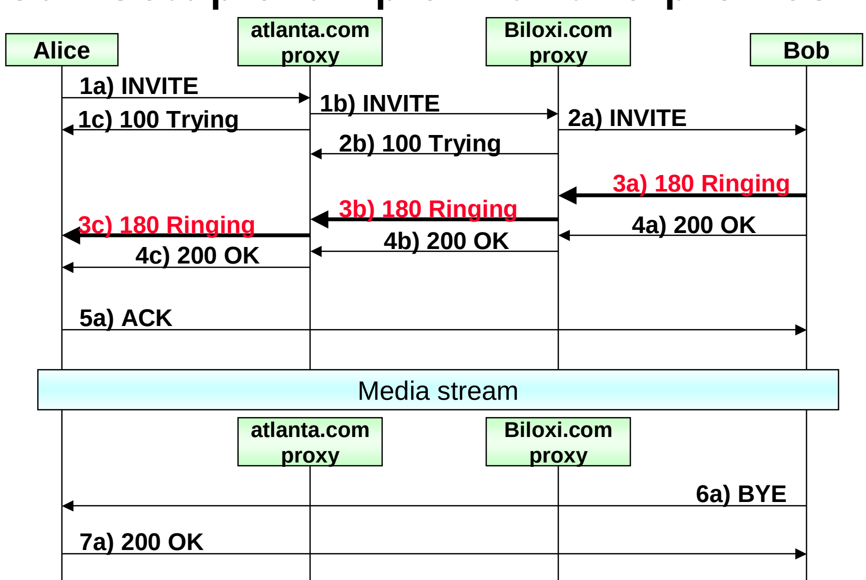
Call Setup example with two proxies



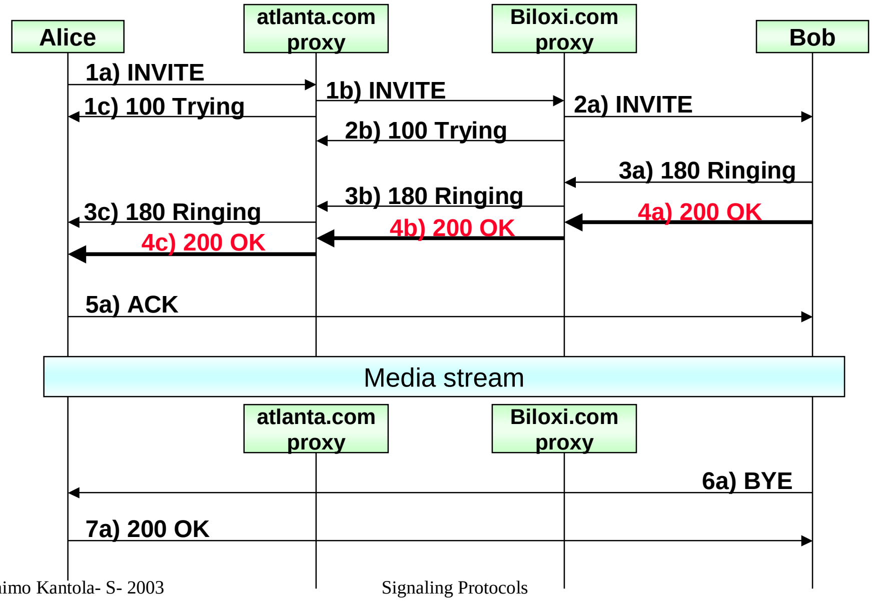
Call Setup example with two proxies



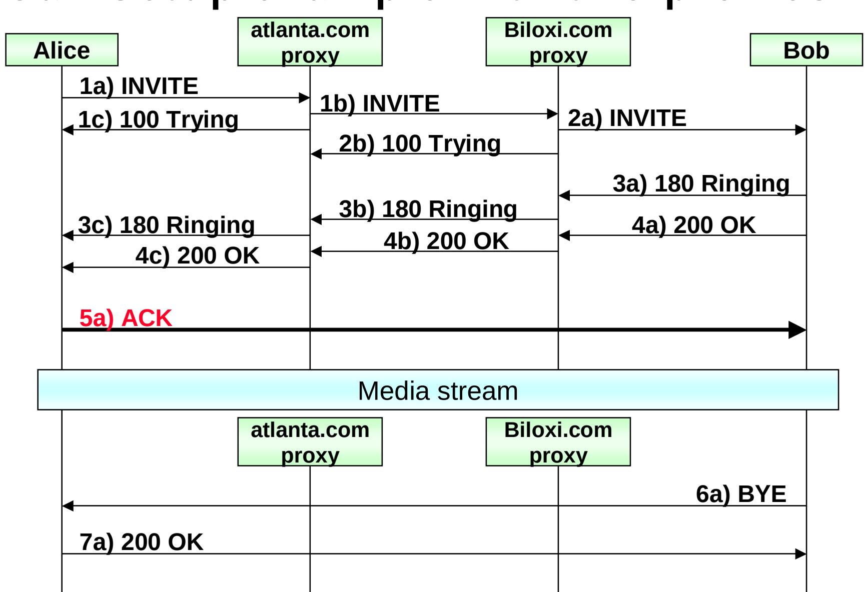
Call Setup example with two proxies



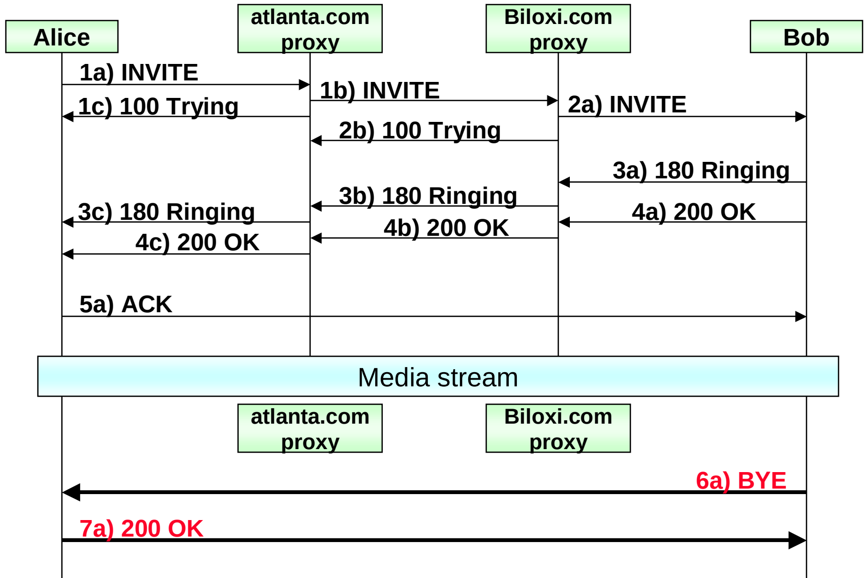
Call Setup example with two proxies



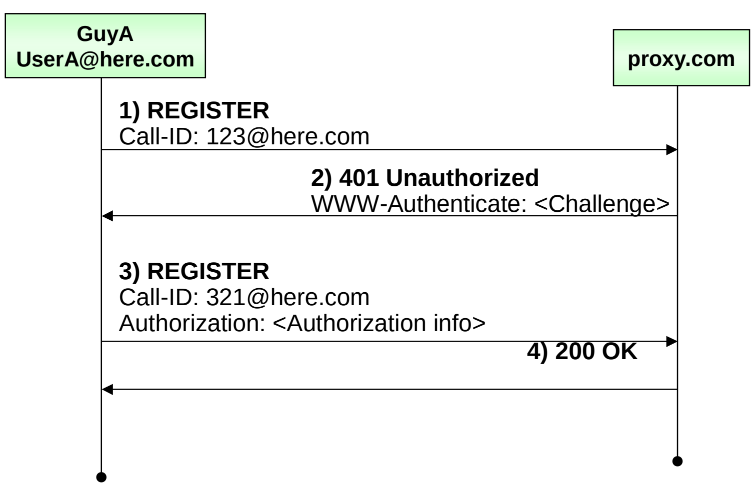
Call Setup example with two proxies



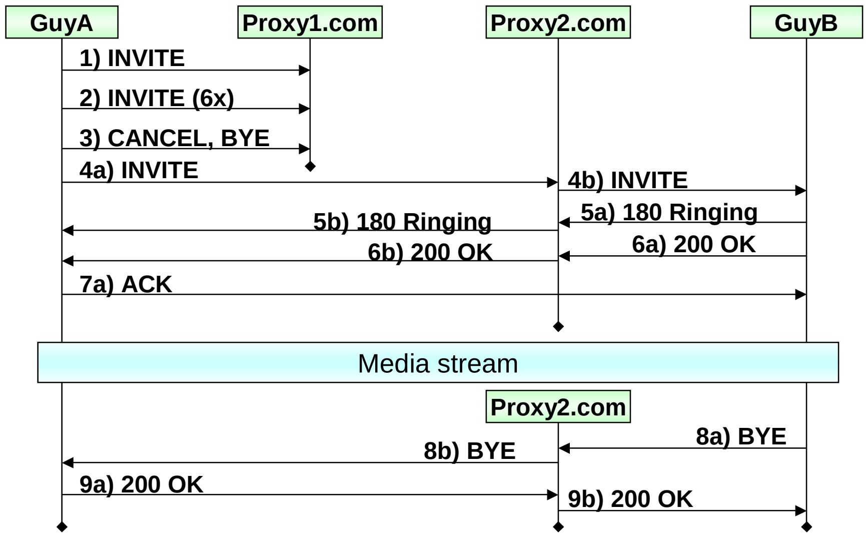
Call Setup example with two proxies



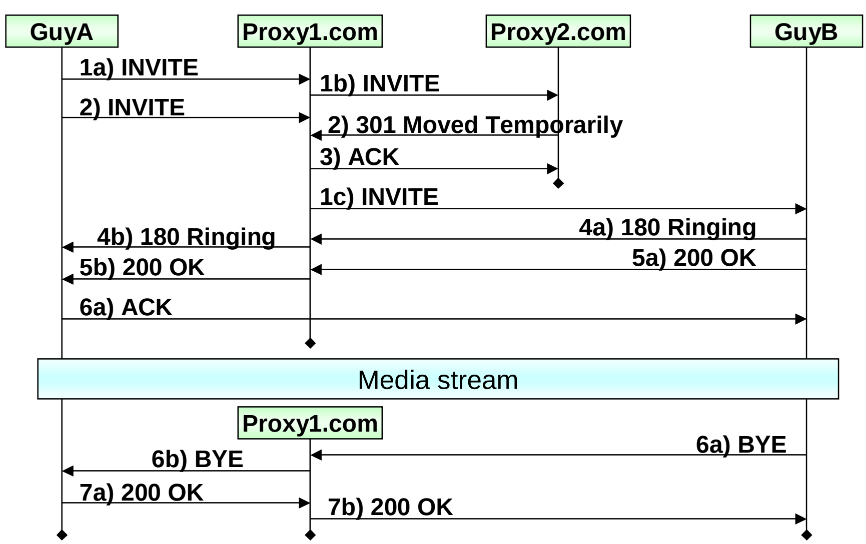
Registration example with SIP authentication



Call Setup example with a non-working proxy

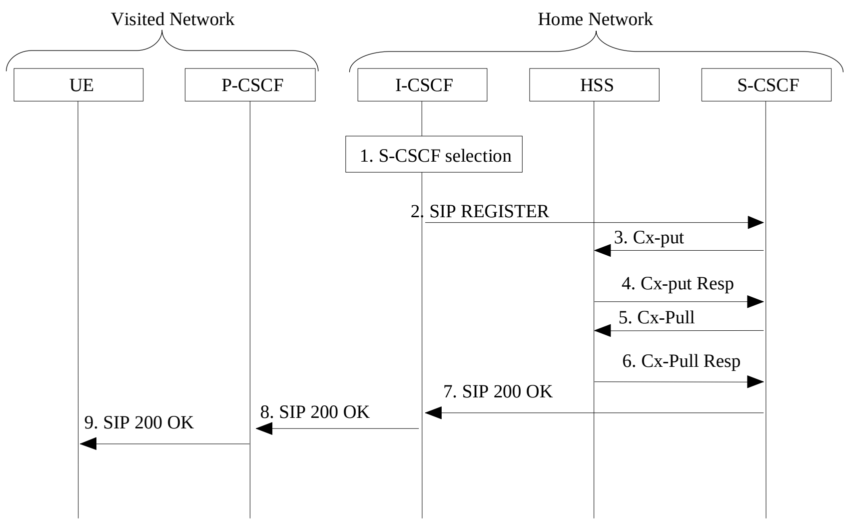


Call Setup example with a Redirect server

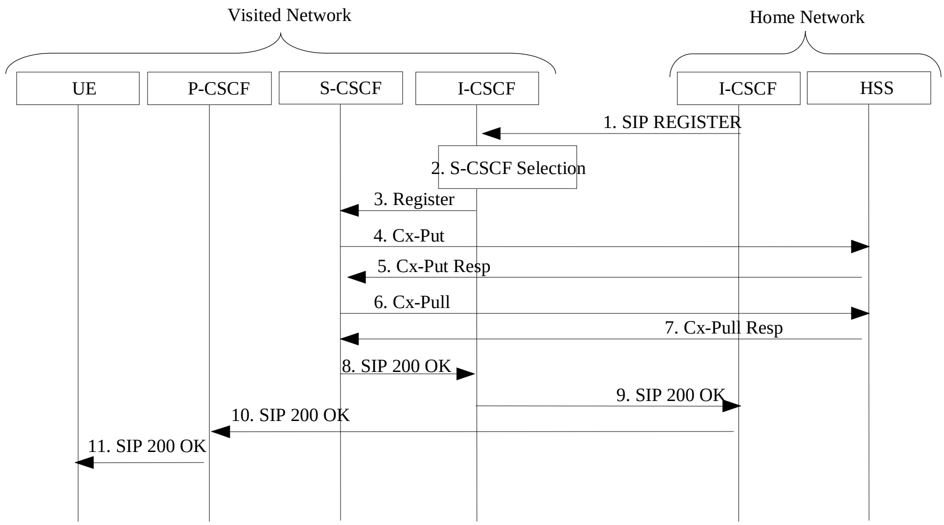


Appendix B – 3GPP IMS call flows

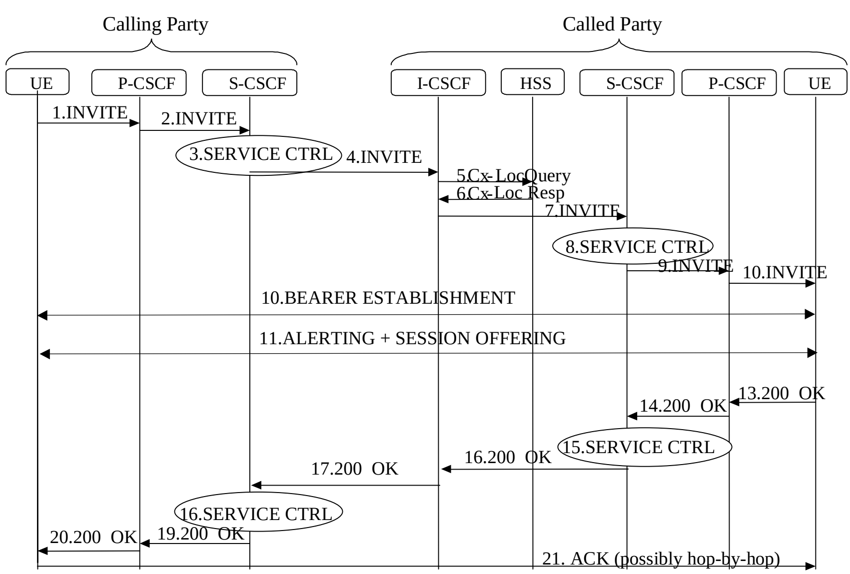
IMS Registration 1a. - S-CSCF in home network



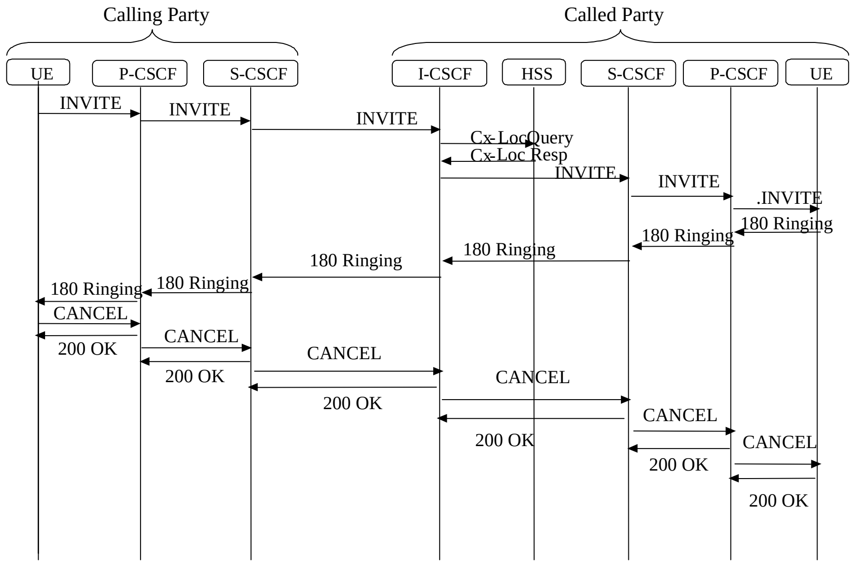
IMS Registration 1b. - S-CSCF in visited network



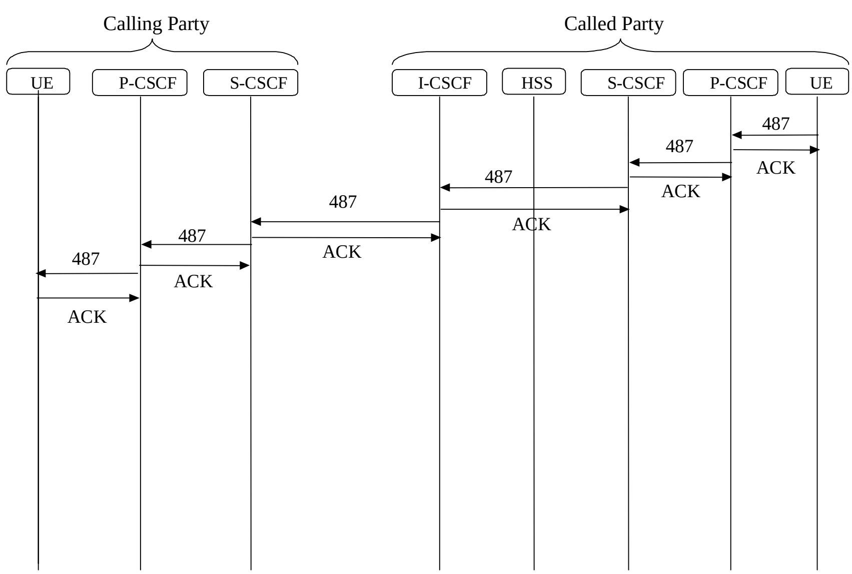
Mobile to Mobile Call



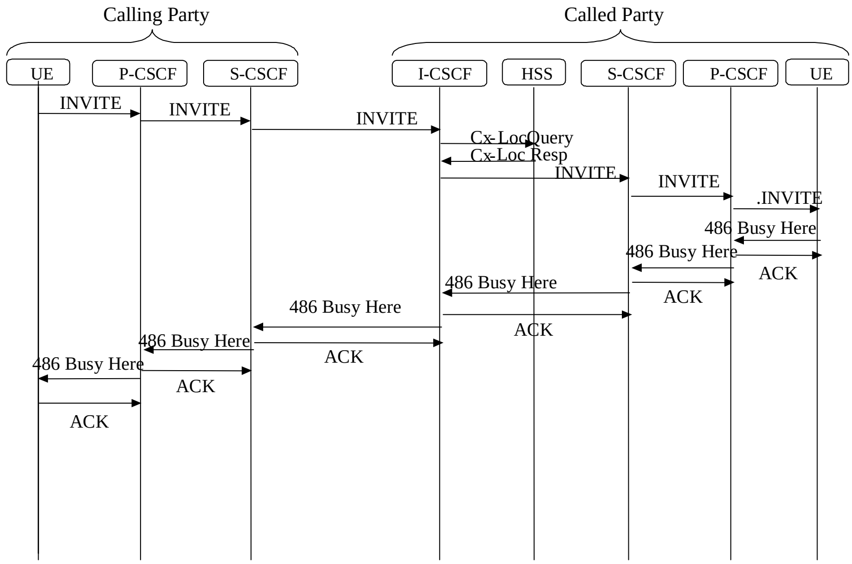
Call flow examples 1. - no answer



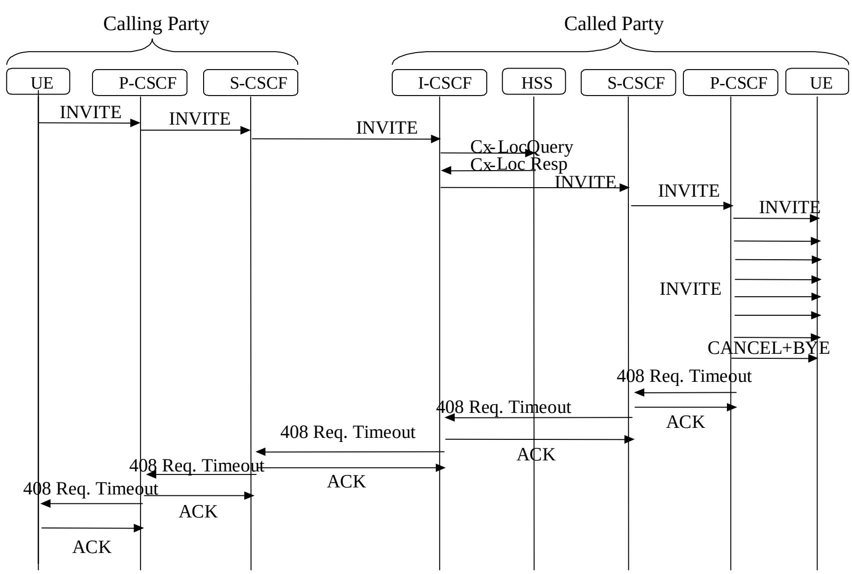
Call flow examples 1. - no answer 2.



Call flow examples 2. - busy



Call flow examples 3. - no response



Call flow examples 4. - temporarily unavailable