UNIVERSITA' DEGLI STUDI DI PARMA
Dipartimento di Ingegneria dell'Informazione

Session Initiation Protocol (SIP)

Luca Veltri

(mail.to: luca.veltri@unipr.it)

Corso di Reti di Telecomunicazioni C, a.a. 2008/2009

http://www.tlc.unipr.it/veltri

The Session Initiation Protocol (SIP)

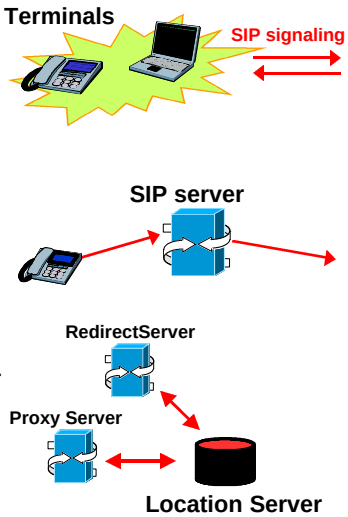
- Application-layer signaling protocol for creating, modifying and terminating sessions with one or more participants
  - **sessions include Internet telephone calls, Internet multimedia conferences, multimedia distribution and instant messaging**
  - **SIP can be used to initiate sessions as well as invite members to sessions previously established**
  - **SIP invitations carry session descriptions which allow participants to agree on a set of compatible media types**
  - **members in a session can communicate via multicast or via a mesh of unicast relations, or a combination of these**
  - **SIP does not reserve resources**
- RFC 3261, current standard (SIP/2.0), June 2002
  - **obsolates proposed standard RFC 2543 (March 1999)**
- SIP has been chosen by 3GPP as signaling protocol for next UMTS (from release 2000)

SIP characteristics

- Application level protocol (client-server) on top of UDP (preferred), TCP, TLS or SCTP
  - **SIP provides its own mechanisms for reliability**
- Text based protocol (similar to HTTP)
- SIP addresses users by an email-like addresses (SIP URLs)
  - **example: `<sip:luca.veltri@unipr.it>`**
- User capabilities
  - **by means of SDP**
  - **determination of media and media parameters during call set-up**
- IN-style and supplementary services can be provided
  - **(re)-negotiate session parameters**
  - **"forking" of calls**
  - **transfer of calls**

SIP architectural elements

- User Agent
  - **end system (terminal)**
  - **can initiates calls, acting as caller (User Agent Client)**
  - **can respond, redirect and refuse entering calls, acting as callee (User Agent Server)**
- SIP Servers
  - **systems that may proxy or redirect SIP messages (similar to the H.323 Gatekeeper)**
  - **may keep information on user location**
- Location Servers
  - **A location server is used by a SIP redirect or proxy server to obtain information about a called party's possible location(s).**
  - **It is not a SIP element and the protocols between the SIP (proxy/redirect) server and the location server is not SIP**

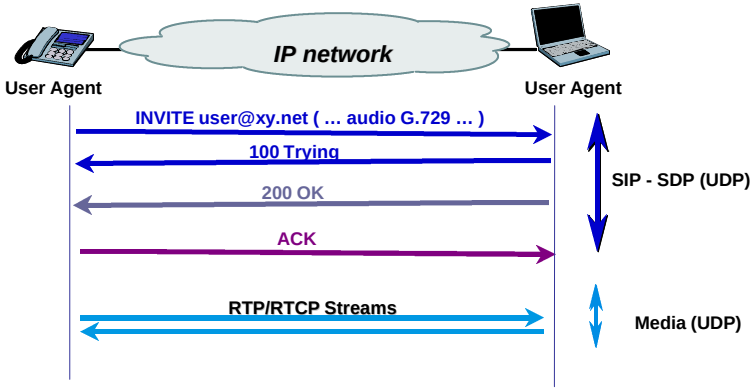


SIP architectural elements (cont.)

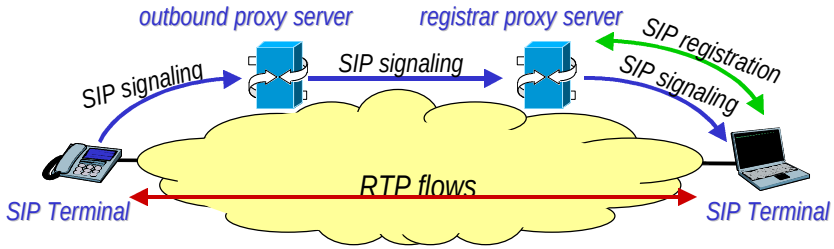
- SIP Gateway
  - **An application that interfaces a SIP network to a network using another signaling protocol (SGM)**
  - **It can also terminate the media path (MGW):**
    - a SIP/PSTN gateway terminates both the signaling and the media path
    - a SIP/H323 gateway terminates only the signaling path
  - **"It is just a special type of user agent" (supporting lots of users)**

SIP principles - SIP call setup

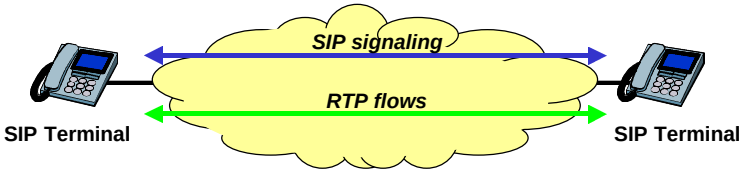
- SIP is based on a client server paradigm, the protocol is similar to HTTP
- Each request is sent by a client to a server…
- Proxy can be easily introduced in the call flows...



SIP principles - SIP call setup

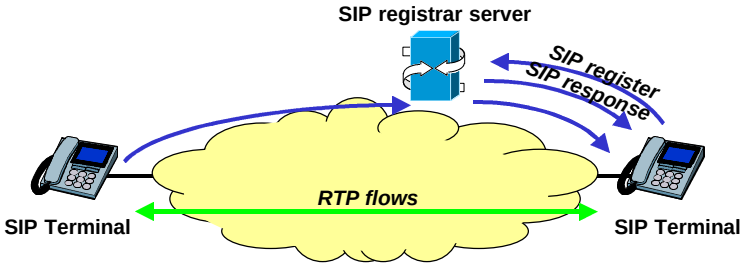


SIP architecture: peer-to-peer calls



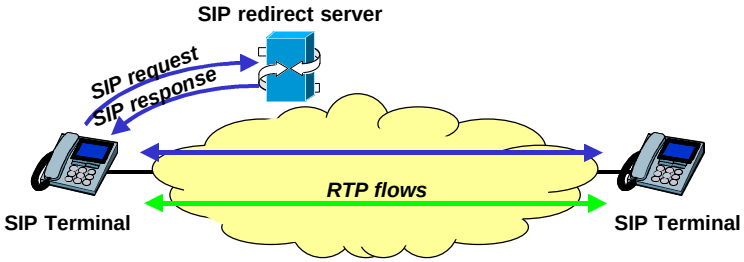
SIP architecture: registrar server

- Registrar server
  - **a server that accepts REGISTER requests**
  - **the register server may support authentication**
  - **a registrar server is typically co-located with a proxy or redirect server and may offer location services**



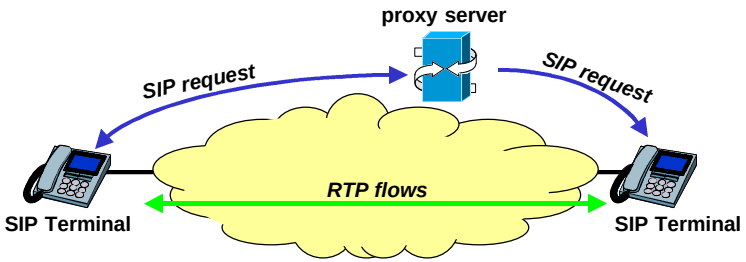
SIP architecture: redirect server

- Redirect server
  - **a server that accepts a SIP request, maps the address into zero or more new addresses and returns these addresses to the client. Unlike a proxy server, it does not initiate its own SIP request. Unlike a user agent server, it does not accept calls.**



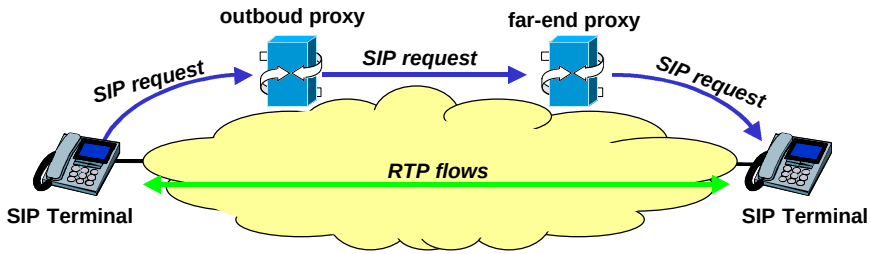
SIP architecture: proxy server

- Proxy server
  - **an intermediary system that acts as both a server and a client for the purpose of making requests on behalf of other clients; requests are serviced internally or by passed, possibly after translation, to other servers.**
  - **may fork requests à parallel or sequential search**
- types: stateless (forward request or response), transaction stateful (remember full request/response), call stateful

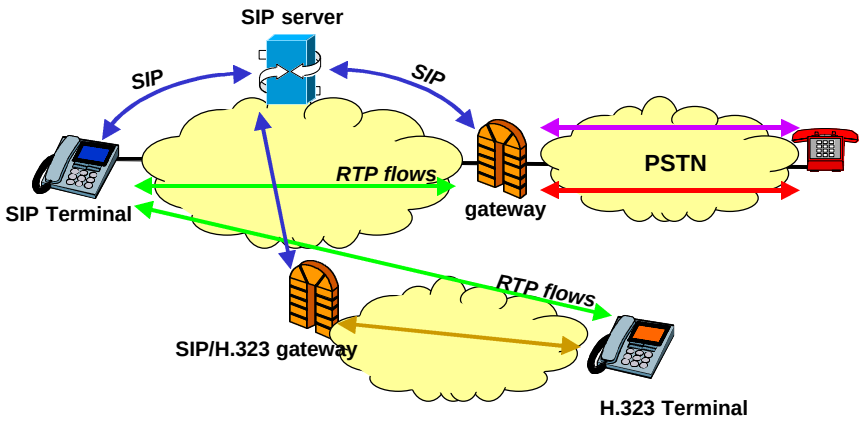


SIP architecture: proxy server (cont.)

- outbound (near-end) proxy:
  outgoing calls à address lookup, policy, firewalls
- far-end proxy:
  closer to callee à callee firewall, call path hiding, registrar



SIP architecture: gateway to PSTN (or H.323)



SIP principles - SIP messages

- HTTP look-alike
  - **SIP is textual client-server protocol as HTTP or SMTP, in which requests are issued by the client and responses are returned by the server**
- two kind of messages:
  - **requests (INVITE, ACK, CANCEL, BYE…)**
  - **provisional and final responses (as HTTP)**

```
INVITE sip:watson@boston.bell-tel.com SIP/2.0
Via: SIP/2.0/UDP kton.bell-tel.com
From: A. Bell <sip:a.g.bell@bell-tel.com>
To: T. Watson <sip:watson@bell-tel.com>
Call-ID: 3298420296@kton.bell-tel.com
CSeq: 1 INVITE
Contact: <sip:watson@saturn.bell-tel.com:3890;transport=udp>
Subject: Mr. Watson, come here.
Content-Type: application/sdp
Content-Length: ...

v=0
o=bell 53655765 2353687637 IN IP4 128.3.4.5
s=Mr. Watson, come here.
c=IN IP4 kton.bell-tel.com
m=audio 3456 RTP/AVP 0 3 4 5 (PCM, GSM, G.723)
```


SIP messages: requests and responses

- Requests
  - **INVITE**
  - **ACK**
  - **CANCEL**
  - **BYE**
  - **OPTION**
  - **SUBSCRIBE**
  - **NOTIFY**
- Provisional and final responses (as HTTP)
  - **1xx = searching, queueing, . . . (e.g. 180 ringing, 183 progress)**
  - **2xx = success (e.g. 200 OK)**
  - **3xx = forwarding**
  - **4xx = client mistakes (e.g. 403 forbidden)**
  - **5xx = server failures**
  - **6xx = busy, refuse, not available anywhere**

SIP requests (methods)

- INVITE, ACK, CANCEL, BYE, OPTION, extension methods
- A SIP request consists of
  - **a request line,**
  - **header fields,**
  - **a message body**
- header fields
  - **contain information on call services, addresses, and protocol features**
- body
  - **is opaque to SIP and can contain anything**
  - **is an object containing a description of the media content of the request, i.e. multiple codecs, ports, protocols**
  - **usually is represented by SDP (Session Description Protocol)**

SIP requests (methods)

- INVITE: invites a party to participate in a session; may include SDP descriptions; reINVITES are use to change session state
- ACK: confirms the reception of an INVITE response (i.e. the session establisment)
- BYE: to leave a session
- REGISTER: to register an address with a SIP server; may convey user data (e.g. call processing scripts)
- CANCEL: cancels a pending request (i.e. a pending INVITE)
- OPTIONS: query about a server capabilities

Example: INVITE message

```
INVITE sip:alice@wonderland.net SIP/2.0
Via: SIP/2.0/UDP phone32.wonderland.net
CSeq: 5452 INVITE
To: <sip:alice@ wonderland.net>
From: <sip:peter@neverland.net>
Content-Type: application/sdp
Call-ID: 1804289383@phone32.neverland.net
Subject: New Call
Content-Length: 182
Contact: <sip:peter@phone32.neverland.net >

v=0
o=username 0 0 IN IP4 192.168.200.2
c=IN IP4 192.168.200.2
t=0 0
m=audio 33422 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=ptime:20
m=video 22000 RTP/AVP 31
```


Provisional or final response codes

1xx	**Informational**	(Provisional)
2xx	**Success**	(Final)
3xx	**Redirection**	(Final)
4xx	**Client error**	(Final)
5xx	**Server error**	(Final)
6xx	**Global failure**	(Final)

SIP response codes

1xx Informational

100 continue

180 ringing

2xx Success

200 OK

3xx Redirection

301 multiple choices

302 moved permanently

303 moved temporarily

SIP response codes (cont.)

4xx Client error

400 bad request
401 unauthorized
403 forbidden
404 not found
407 proxy auth required
408 request timeout
420 bad extension
480 temporarily unavailable
481 call leg doesn't exist
482 loop detected
483 too many hops
484 address incomplete
485 ambiguous
486 busy here
487 request cancelled

SIP response codes (cont.)

5xx Server error

500 server internal error
501 not implemented
502 bad gateway
503 service unavailable
504 gateway timeout
505 version not supported

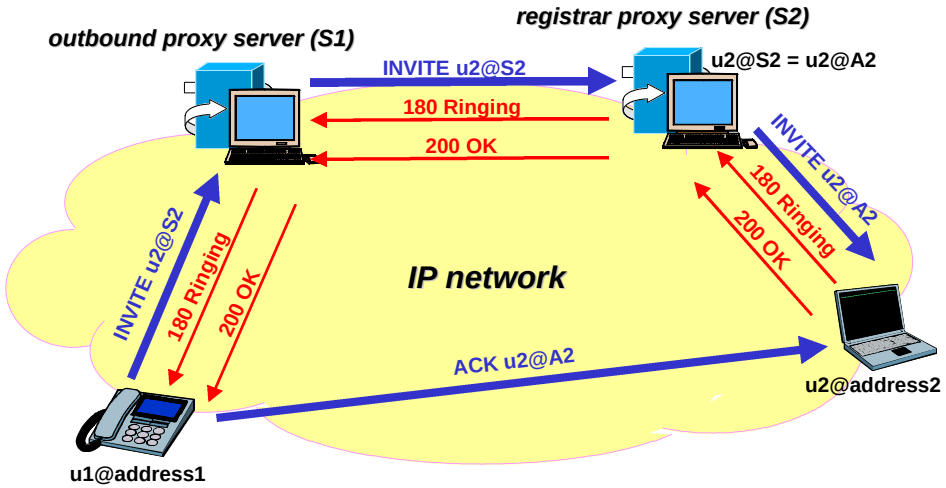
6xx Global failure

600 busy
601 decline
602 does not exist
606 not acceptable

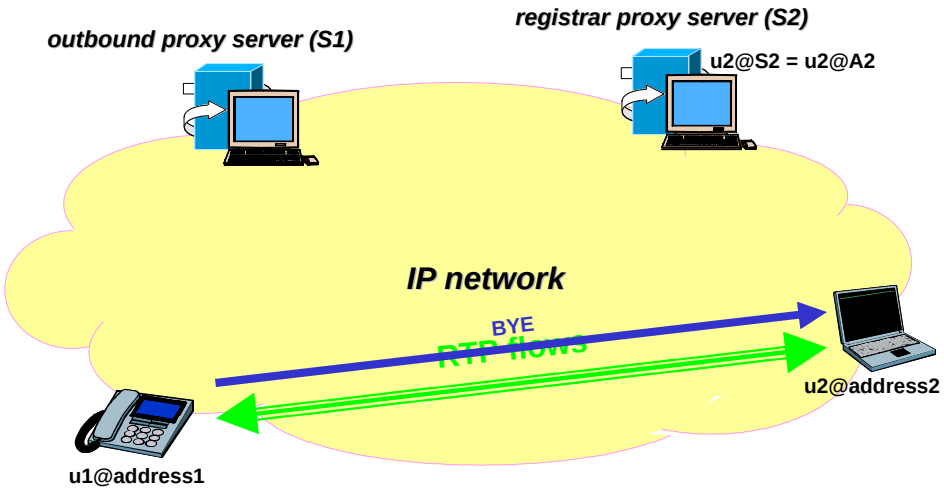
Transport of SIP messages

- Unlike HTTP and SMTP, SIP can run on top of either TCP or UDP (preferred!)
  - **SIP provides its own mechanisms for reliability**
- SIP can use multicasting
  - **multicasting allows, for example, group invitations**
- UDP allows:
  - **multicasting**
  - **fast operation, by avoiding TCP handshake**
- default port = 5060
- It can use also TCP, TLS, SCTP

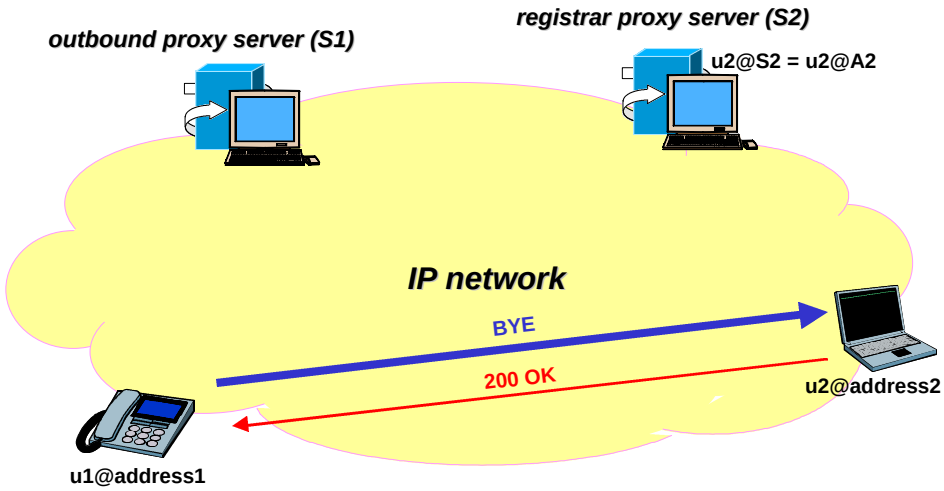
SIP call



SIP call



SIP call



SIP principles - Independence of session description

- SIP Protocol is de-coupled from description of session
- The session is described by the Session Description Protocol - SDP
- SIP carries SDP elements as “Payload”

```
INVITE sip:watson@boston.bell-tel.com SIP/2.0
Via: SIP/2.0/UDP kton.bell-tel.com
From: A. Bell <sip:a.g.bell@bell-tel.com>
To: T. Watson <sip:watson@bell-tel.com>
Call-ID: 3298420296@kton.bell-tel.com
CSeq: 1 INVITE
Contact: <sip:watson@saturn.bell-tel.com:3890;transport=udp>
Subject: Mr. Watson, come here.
Content-Type: application/sdp
Content-Length: ...

v=0
o=bell 53655765 2353687637 IN IP4 128.3.4.5
s=Mr. Watson, come here.
c=IN IP4 kton.bell-tel.com
m=audio 3456 RTP/AVP 0 3 4 5 (PCM, GSM, G.723)
```

SIP HEADER

SDP PAYLOAD

SDP - Session Description Protocol

- SDP (Session Description Protocol, RFC 3550) allows endpoints to describe media types and encodings
  Not really a protocol, but a description syntax
- SDP describes IP addresses, TCP/UDP ports, coding type…
- SIP carries the SDP information transparently

- Roughly speaking, SIP corresponds to H.225.0 call handling, and SDP descriptions correspond to H.245 capabilities description
  - **the maior difference is that with SIP the capabilities description (SDP) is inserted directly within the call setup messages (INVITE, 200 OK, ACK)**

Session Description Protocol (SDP)

- RFC 3550, July 2003
  - **obsoletes RFC 2327 (April 1998)**
- **«The purpose of SDP is to convey information about media streams in multimedia sessions to allow the recipients of a session description to participate in the session»**
- SDP contains the following information about the media session:
  - **IP address (IPv4 address or host name)**
  - **port number (UDP or TCP)**
  - **media type (audio, video, interactive whiteboard, etc.)**
  - **media encoding scheme**
- Other information regarding the session:
  - **subject of the session,**
  - **start and stop times,**
  - **etc..**

SDP format

- Like SIP, SDP uses text coding
- an SDP message is composed of a series of lines (called field)
- the SDP field are in a required order (to simplify the parsing)
- each field is composed by
  - **a field identifier (one lower-case letter)**
  - **a sequence of parameters**

    x= parameter1 parameter2 … parameterN
- example:

```
v=0
o=alice 947205083 947205083 IN IP4 hegel.tlc.unipr.it
s=
c=IN IP4 192.168.101.2
t=3156193883  3156197483
m=audio/* 10000 RTP 0
```


SDP fields

v= Protocol version number
o= Owner/creator and session identifier
s= Session name
i= Session name
u= Uniform Resource Identifier
e= Email address
p= Phone number
c= Connection information
b= Bandwidth information
t= Time session starts and stops
r= Repeat times
z= Time zone correction
k= Encryption key
a= attribute lines
m= Media information
a= Media attribute

--- Mandatory
--- Optional

SIP principles - Addressing

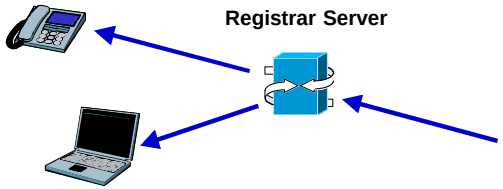
- SIP addresses users by an email-like address
  - **and re-uses some of naming infrastructure**
- SIP is addressing-neutral
  - **addresses are expressed as URLs of various types such as SIP, H.323 or telephone (E.164)**
- The format of the addresses is the URL format
  - **examples:**
    - sip:alice@iptel.org
    - sip:+39-0521-905055:3333@unipr.it;user=phone
    - sip:alice@160.78.0.5
    - sip:voicemail@iptel.org?subject=callme
    - sip:sales@hotel.xy;geo.position:=48.54_-123.84_120

SIP principles - Addressing

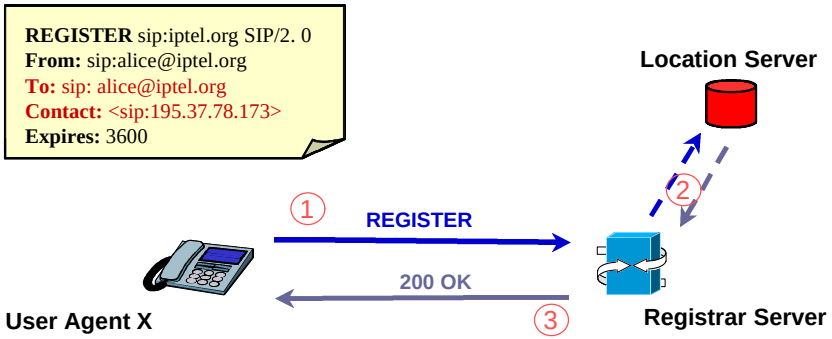
- Global reachable users
  - **callees bind to their Address-Of-Record (AOR) using SIP REGISTER method (or other non-SIP mechanisms)**
  - **callers use this AOR to establish real-time communication with callees**
- SIP URLs
  - **sip:[user@]host:[port];transport=UDP;maddr=224.2.0.1**
    - used in Request-URI, Location headers (redirect, registration), web pages
    - "transport" and "maddr" fields specify transport
- The URL must include host, may include user name, port number, parameters (e.g.,transport), etc. It may be embedded in Webpages, email signatures, printed on your business card, etc.
- The address space is unlimited

SIP principles - Addressing

- SIP addresses users (not terminals)
- Thanks to user-based addressing (AOR), SIP provides personal mobility
- The user can access the service from different points and can associate different terminals with different capabilities (phone, videophone, answering machine)

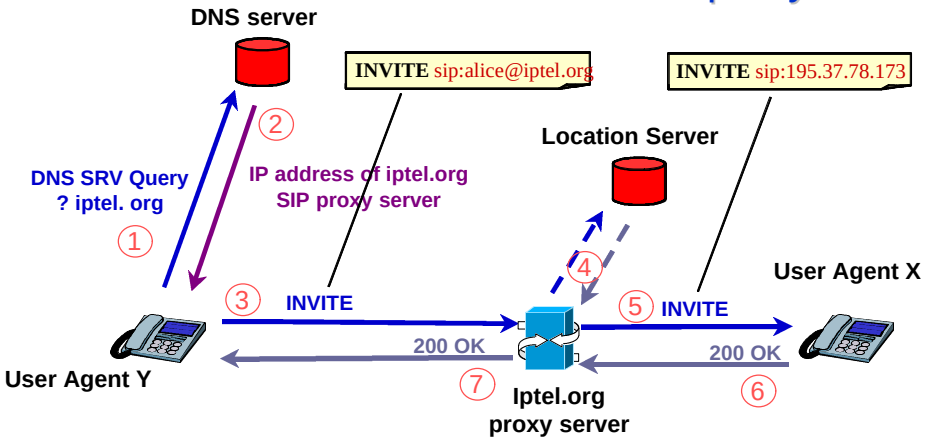


SIP Registration



This registration example establishes presence of user with address alice@iptel.org and binds this address to user's current location 195.37.78.173

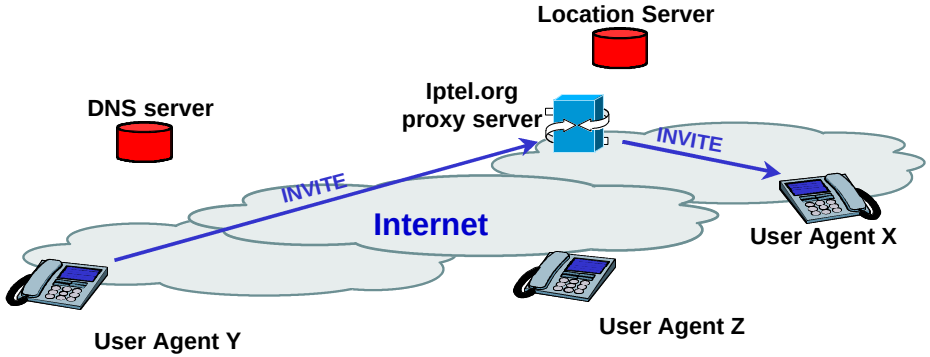
SIP URL resolution with DNS & proxy



User Agent Y wants to call alice@iptel.org

More on proxy servers

SIP call using a destination proxy

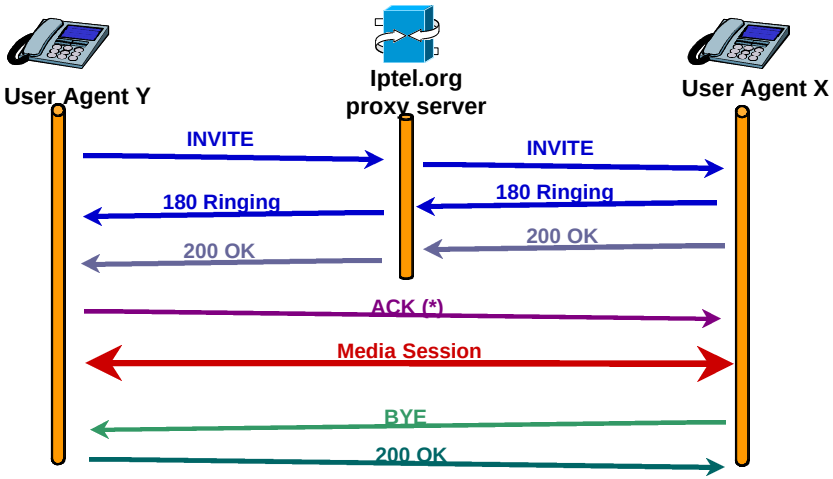


User Agent Y wants to call alice@iptel.org

alice@iptel.org could be at User Agent X or Z …

User Agent Y will contact the destination proxy

SIP call using a destination proxy

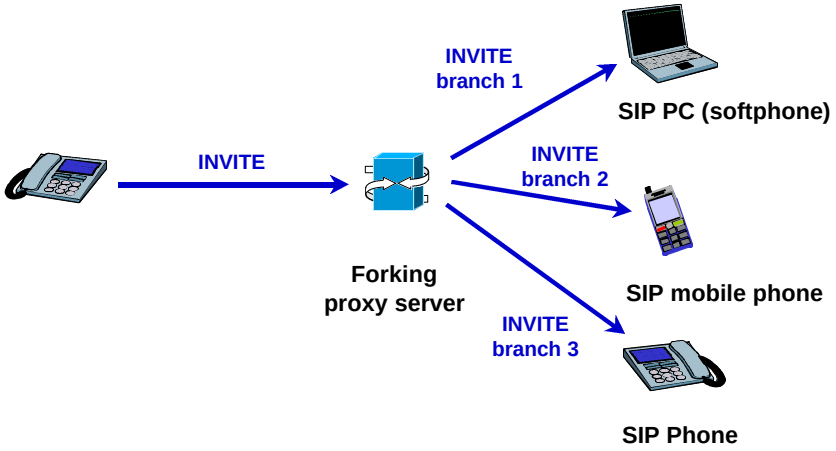


(*) There can be a direct communication after the 200 OK Message

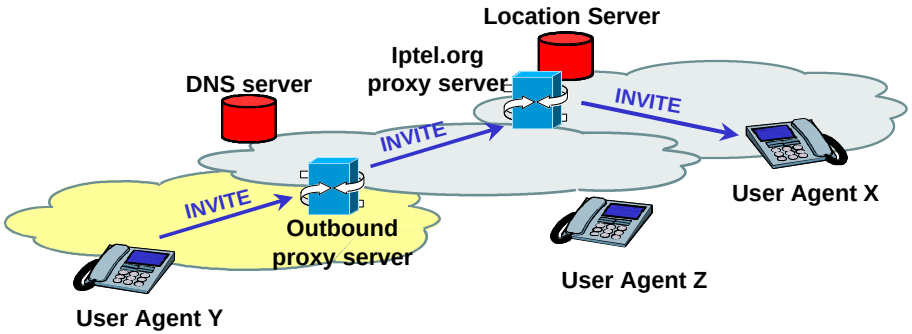
(Destination) proxy server functionality

- Serves as rendezvous point at which callees are globally reachable
- Performs routing function, i.e., determines to which hop the SIP message should be relayed
- Allows the routing logic to be programmable
- Forking: several destination for a request may be tried, sequentially or in parallel
- A proxy server has no media capability

Forking proxy operation



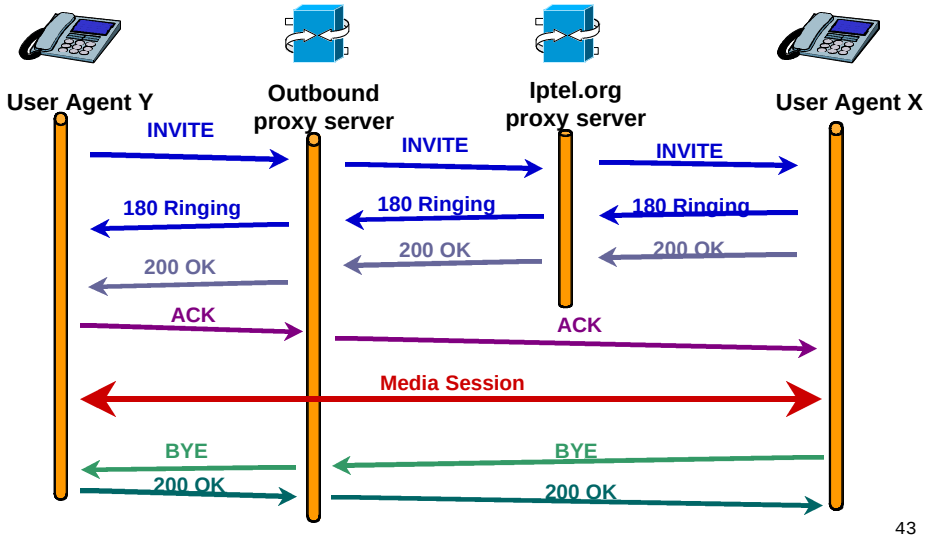
Outbound proxy - Proxy Chaining



User Agent Y wants to call jiri@iptel.org

It has to use its own Outbound proxy server...

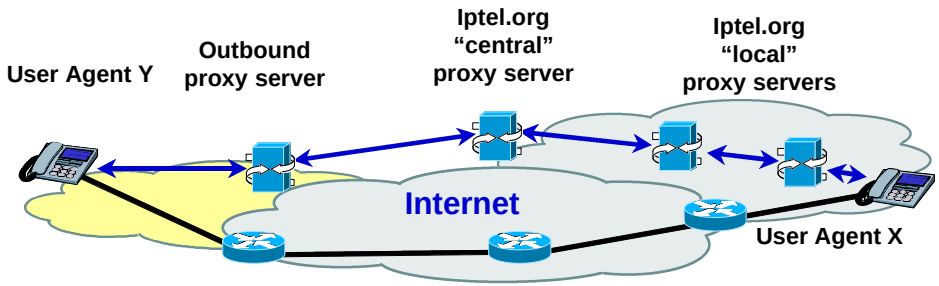
Outbound proxy - Proxy Chaining



Outbound proxy - Proxy Chaining

- Some reasons to use outbound proxy:
  - **locally important call processing (i.e. emergency calls)**
  - **firewall interaction**
  - **searches the gateway with least cost**
  - **manages accounting for local users**
- IP phones needs to know Outbound GW address
- Proxy chaining can be used to distribute signaling
- Server must take care of avoiding loops

More on Proxy Chaining

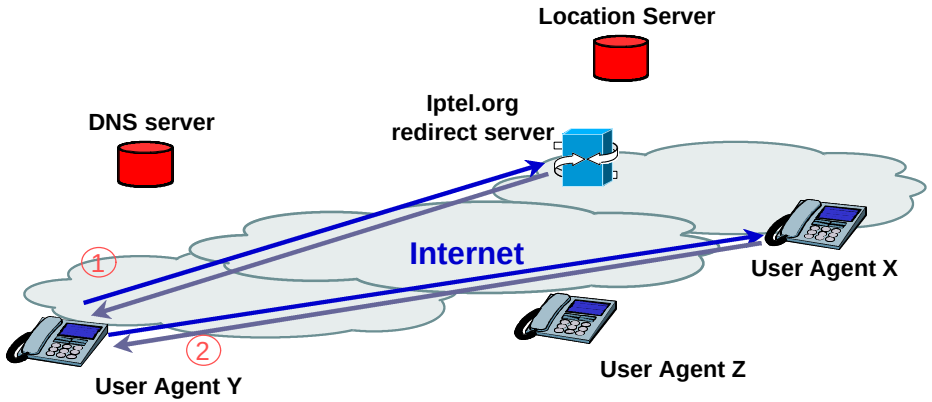


Note: Signaling (in blue) may take a different path from Media (in black)!

Media Path is different from Signaling Path

- SIP proxies have no notion of media path
- SIP proxies can not usually control media path as there is split between signaling and media
- A SIP proxy may be located far apart from media path
- IP, DiffServ, and RSVP are the protocols for communication between end devices and network
- Attempts to manipulate media flows in the middle of path will tend to fail:
  - **A proxy does not know all IP hops along an end-to-end media path**
  - **Hops may belong to foreign administrative domains.**
- Signaling and media transport (possibly with QoS are different businesses)

SIP call using a redirect proxy

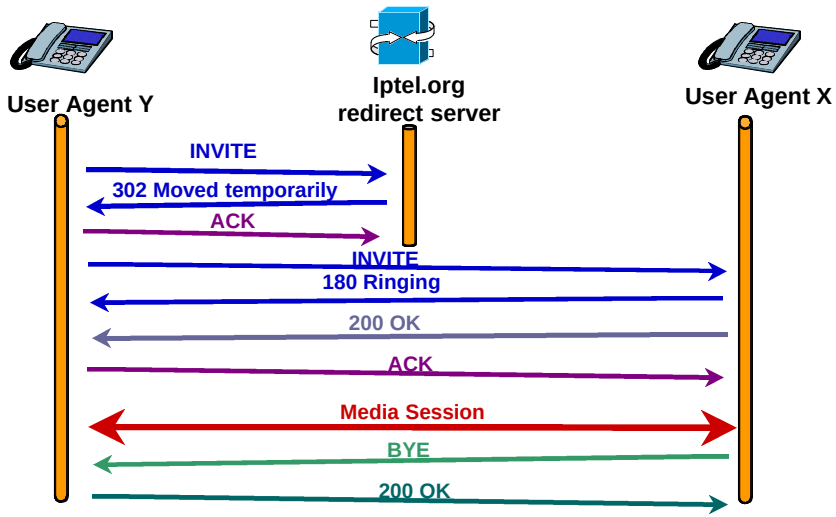


User Agent Y wants to call jiri@iptel.org

User Agent Y will contact the destination redirect server

The redirect server just give back the address

SIP call using a redirect proxy



Proxy vs. Redirect Server

- A SIP server may either proxy or redirect a request
- Which of the two method applies is a configuration issue
  - **It may be statically configured or dynamically determined (Call Processing Language - CPL)**
- Redirection useful if a user moves or changes his provider (PSTN: “The number you have dialed is not available)
- Proxy is is useful if forking, authentication & accounting, firewall control are needed
  - **In general proxying grants more control to the server**

Call *Processing* Scripts

- Specifies behaviour of a server or terminal in responding to incoming and outgoing calls
- Textual description of desired operation
  - **e.g. CPL, XML**
- Independent of underlying signalling protocol
- Generated in many ways
  - **Written by savvy users**
  - **Written and provided by system administrators**
  - **Created by the user as a result of some GUI application**
  - **Written and provided by third party “call logic providers”**
    - Call screening services
    - Automatic mobility services

Example Scripted Services

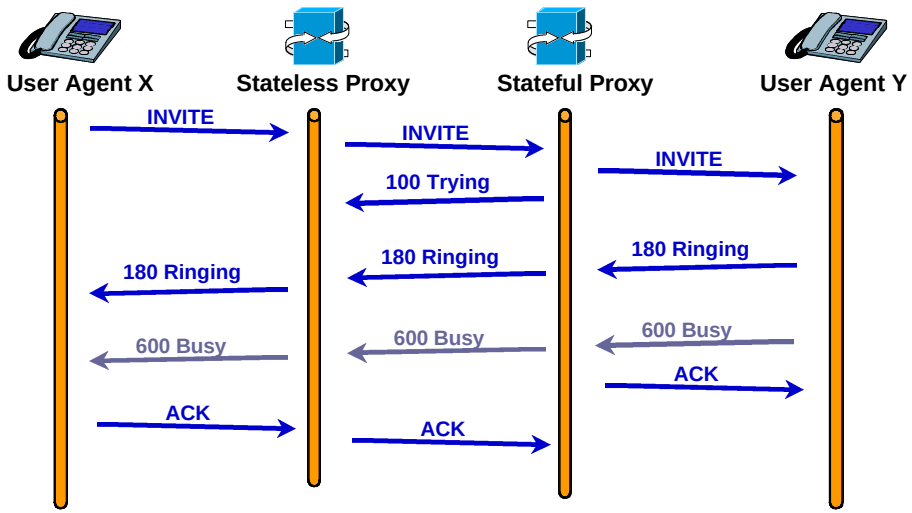
- Call forward
  - **Based on time of day, caller, number of current calls**
- Call redirect
- Automatic call reject
  - **Based on caller**
  - **Spam protection**
- Distinctive ringing
  - **Also based on caller, priority of call, etc.**
- Outgoing call screening
  - **Prevent calls to adult numbers**
  - **Prevent calls to recruiters..**

What might it look like?

- Textual
  - **Allows for cross platform usage, easy editing**
- Simple commands
- A few variables representing call state
- Several approaches
  - **Tcl/Tk**
  - **XML**

```
<call>
  <proxy dest=“sip:bob@mci.com” timeout=“8s”>
    <busy>
      <redirect dest=“sip:joe@mit.edu”/>
    </busy>
    <timeout>
      <condition from=“hgs@cs.columbia.edu”>
        <match>
          <gateway dest=“phone:+19175551212”/>
        </match>
        <nomatch>
          <redirect dest=“sip:bill@att.com”/>
        </nomatch>
      </condition>
    </timeout>
  </proxy>
</call>
```

Call with Busy Callee ad Stateful/less Proxies

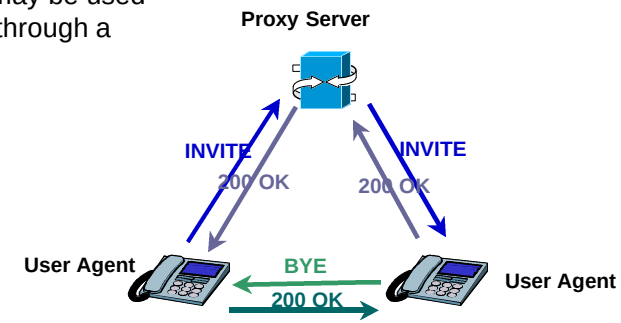


Stateless and stateful proxy

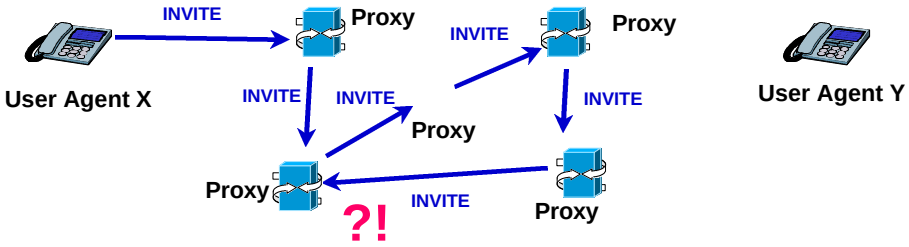
- A proxy server can be stateless or stateful
- A stateless proxy processes each SIP request or response based only on the message content
  - **A stateless proxy is still capable of detecting message loops, as SIP uses a stateless mechanism for loop detection**
- A stateful proxy keeps track of request and responses and uses this information in processing future messages
  - **For example it starts a timer when a request is sent. At timer expiry it resends the request**
  - **A stateful proxy is used to request authentication**
  - **A stateful proxy is needed to "fork" INVITEs to multiple destinations**
  - **A stateful proxy usually sends 100 Trying responses (the 100 Trying response is single hop)**
- A forking proxy server must be stateful!

Stateless and stateful proxy

- In general the state is referred to SIP transactions, not to SIP calls… a SIP server does not records all the active calls
- The problem is that subsequent transaction bypass proxy
- "Record route" may be used to force routing through a proxy server



Loop avoidance



- In order to avoid loops, each proxy adds a `Via` header with its own address. Before forwarding the message, proxies make sure their own address is not already on the list
- In order to limit the number of proxy that can forward a message, "`Max-Forwards`" header is used
  - **it is decremented by each proxy. If it reaches 0, the proxy discards the request and sends a "`483 Too Many hops`" to the originator**

Routing of messages - Requests

- the host sends requests to local proxy or resolve the hostname in the Request-URI using DNS
- each proxy checks for loops, then prepends a Via header with its own address

```
Via: SIP/2.0/UDP proxy.domain.org:5060
```

- UAS copies the VIA headers to response
- `branch` indicates forking
- there is a mechanism (`received` field) to work with NAT

Routing of messages - Responses

- response message follows back the same route of request without the need of *proxy server state*
- when receiving the response, each proxy checks the first line and removes it
- each proxy forward the response to the host/port in next `Via`

```
Via: SIP/2.0/UDP server.domain.org:5060; received=128.1.2.3
```

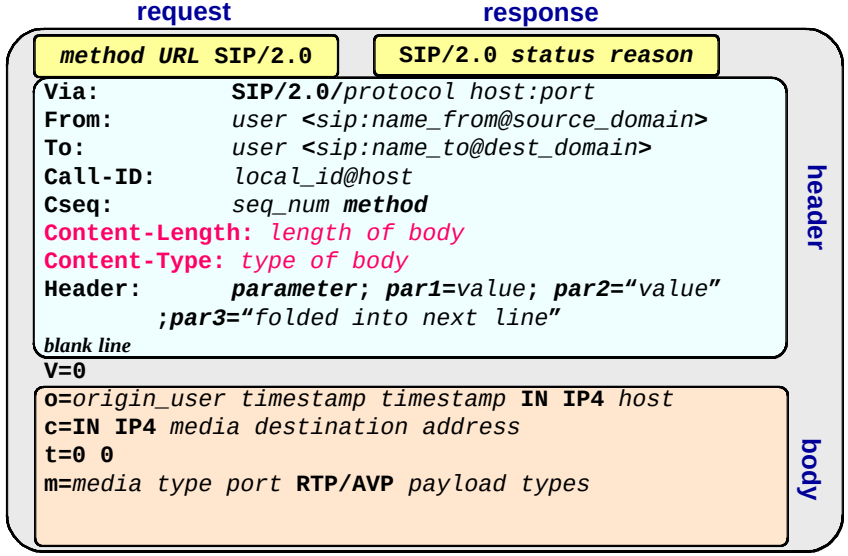

SIP Message Syntax

- Many header fields from http
- New ones are SIP specific
- Payload contains a media description
- SDP - Session Description Protocol →

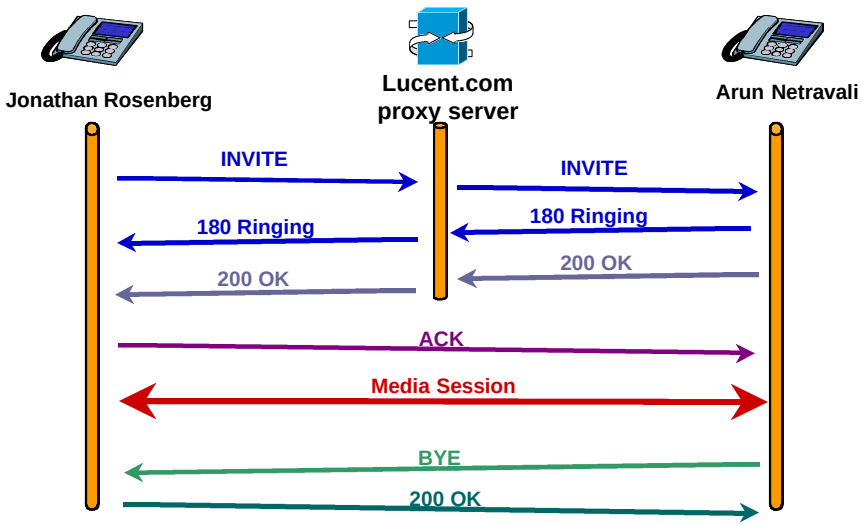
```
INVITE ann@lucent.com SIP/2.0
From: Rosenberg <sip:rosen@bell-labs.com>
Subject: SIP will be discussed, too
To: Netravali <sip:ann@lucent.com>
Call-ID: 1997234505.56.78@
Content-type: application/sdp
CSeq: 4711
Content-Length: 187

v=0
o=user1 53655765 2353687637 IN IP4
128.3.4.5
s=Mbone Audio
i=Discussion of Mbone Engineering Issues
e=mbone@somewhere.com
c=IN IP4 224.2.0.1/127
t=0 0
m=audio 3456 RTP/AVP 0
```

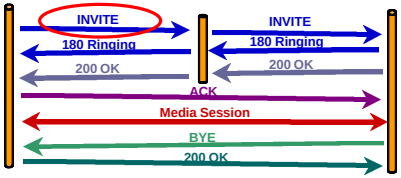

SIP Message Syntax



SIP call with a destination proxy



INVITE

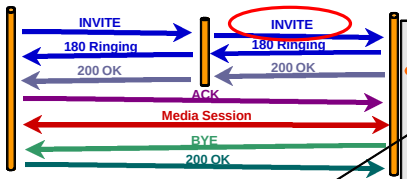


Request URI

```
INVITE sip:ann@lucent.com SIP/2.0
Via: SIP/2.0/UDP 193.205.0.1:5060
To: Netravali <sip:ann@lucent.com>
From: Rosenberg <sip:rosen@bell-labs.com>
Subject: SIP will be discussed, too
Call-ID: 1997234505.56.78@193.205.0.1
CSeq: 4711 INVITE
Contact: sip:rosen@bell-lab.com
Content-type: application/sdp
Content-Length: 183

v=0
o=user1 53655765 2353687637 IN IP4
193.205.0.1
s=Mbone Audio
i=Discussion of Mbone Engineering
e=mbone@somewhere.com
c=IN IP4 193.205.0.1
t=0 0
m=audio 3456 RTP/AVP 0
```


INVITE



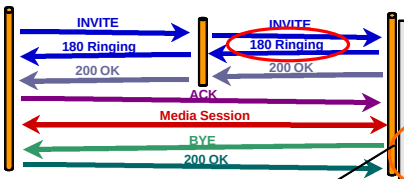
Proxy adds the Via header

Branch: first part is computed as hash of To, From, Call-ID and Cseq second part is the forking number

```
INVITE ann@lucent.com SIP/2.0
Via: SIP/2.0/UDP xy.lucent.com;branch=23.1
Via: SIP/2.0/UDP 193.205.0.1:5060
To: Netravali <sip:ann@lucent.com>
From: Rosenberg <sip:rosen@bell-labs.com>
Subject: SIP will be discussed, too
Call-ID: 1997234505.56.78@193.205.0.1
CSeq: 4711 INVITE
Content-type: application/sdp
Content-Length: 183

v=0
o=user1 53655765 2353687637 IN IP4
193.205.0.1
s=Mbone Audio
i=Discussion of Mbone Engineering
e=mbone@somewhere.com
c=IN IP4 193.205.0.1
t=0 0
m=audio 3456 RTP/AVP 0
```


180 Ringing



All `Via:` headers are copied

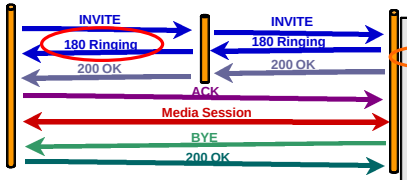
`To:` and `From:` are not exchanged

A `Tag:` is added

No SDP information is carried

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP xy.lucent.com;branch=23.1
Via: SIP/2.0/UDP 193.205.0.1:5060
To: Netravali
    <sip:ann@lucent.com>;tag=2349336
From: Rosenberg <sip:rosen@bell-labs.com>
Call-ID: 1997234505.56.78@193.205.0.1
CSeq: 4711 INVITE
Content-Length: 0
```

180 Ringing

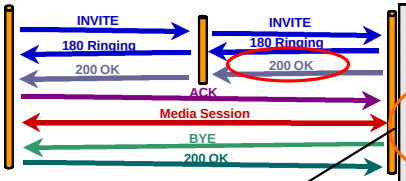


Topmost is removed

```
Via: SIP/2.0/UDP 193.205.0.1:5060
To:
    <sip:ann
From: Rosenberg      rosen@bell-labs.com>

Call-ID: 1997234505.56.78@193.205.0.1
CSeq
Content-Length: 0
```


200 OK



All `Via:` headers are copied

`To:` and `From:` are not exchanged

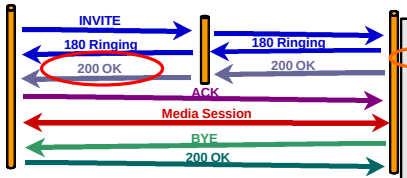
A `Tag:` is added

`Contact:` will be used to send messages bypassing proxy

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP xy.lucent.com;branch=23.1
Via: SIP/2.0/UDP 193.205.0.1:5060
To: Netravali
    <sip:ann@lucent.com>;tag=2349336
From: Rosenberg <sip:rosen@bell-labs.com>
Subject: SIP will be discussed, too
Call-ID: 1997234505.56.78@193.205.0.1
CSeq: 4711 INVITE
Contact: sip:Netravali@151.100.53.12
Content-type: application/sdp
Content-Length: 183

v=0
o=user1 53655765 2353687637 IN IP4
193.205.0.1
s=Mbone Audio
i=Discussion of Mbone Engineering
e=mbone@somewhere.com
c=IN IP4 193.205.0.1
t=0 0
m=audio 3456 RTP/AVP 0
```


200 OK

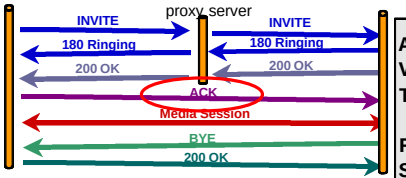


Topmost `Via:` is removed

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 193.205.0.1:5060
To: Netravali
    <sip:ann@lucent.com>;tag=2349336
From: Rosenberg <sip:rosen@bell-labs.com>
Subject: SIP will be discussed, too
Call-ID: 1997234505.56.78@193.205.0.1
CSeq: 4711 INVITE
Contact: sip:Netravali@151.100.53.12
Content-type: application/sdp
Content-Length: 183

v=0
o=user1 53655765 2353687637 IN IP4
193.205.0.1
s=Mbone Audio
i=Discussion of Mbone Engineering
e=mbone@somewhere.com
c=IN IP4 193.205.0.1
t=0 0
m=audio 3456 RTP/AVP 0
```


ACK



`Tag:` set by receiver is copied

The ACK message is sent end-to-end

```
ACK ann@lucent.com SIP/2.0
Via: SIP/2.0/UDP 193.205.0.1:5060
To: Netravali
    <sip:ann@lucent.com>;tag=2349336
From: Rosenberg <sip:rosen@bell-labs.com>
Subject: SIP will be discussed, too
Call-ID: 1997234505.56.78@193.205.0.1
CSeq: 4711 ACK
Content-type: application/sdp
Content-Length: 0
```

BYE

The BYE message is sent end-to-end

Tag: Same Tag: is used

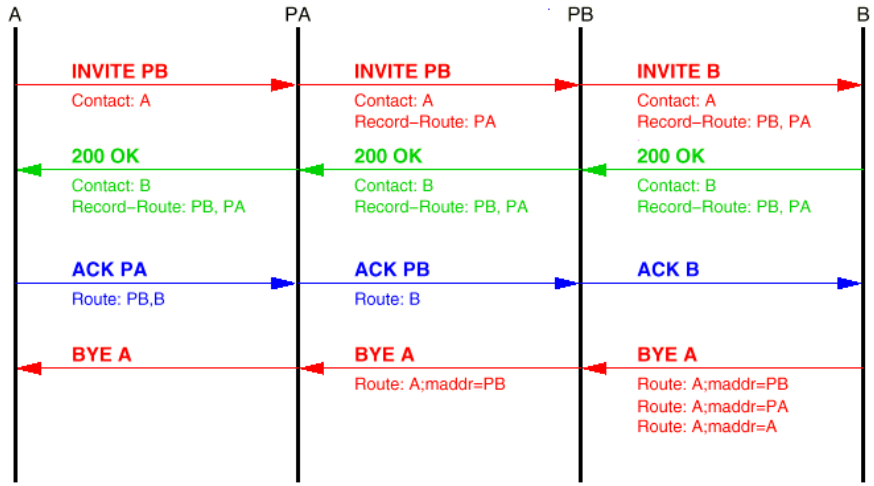
```
BYE sip:rosen@bell-lab.com SIP/2.0
Via: SIP/2.0/UDP 151.100.53.12
To: Rosenberg <sip:rosen@bell-labs.com>
From: Netravali
<sip:ann@lucent.com>;tag=2349336
Call-ID: 1997234505.56.78@193.205.0.1
CSeq: 2000 BYE
Content-Length: 0
```


Forcing a path for signaling

By default, after the 200 OK response is received, the signaling goes directly host-to-host, bypassing proxies

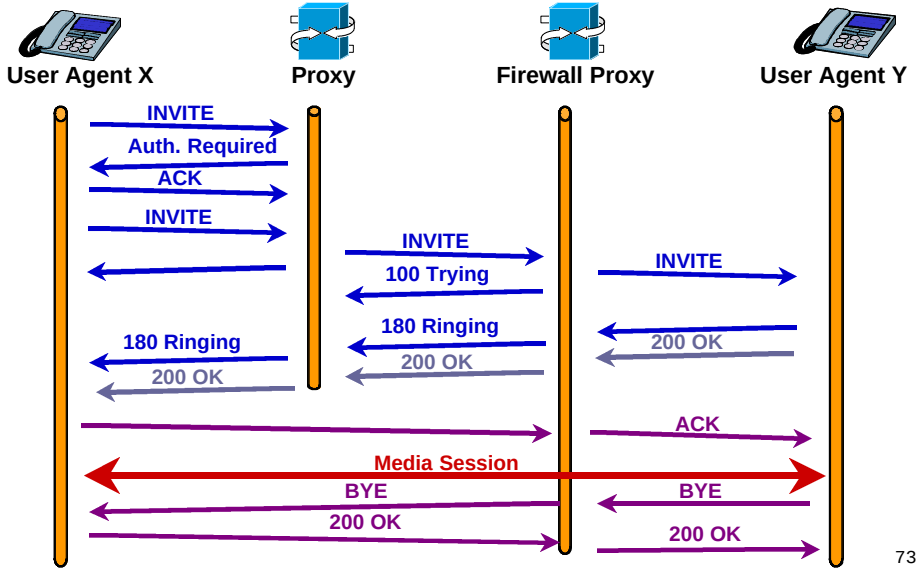
- Some proxies want to stay in the path → "call stateful"
  - **firewall control**
  - **gateway (e.g. to PSTN) control**
  - **anonymizer proxies**
- To this purpose, `Record-Route` and `Route` are used

Forcing a path for signaling

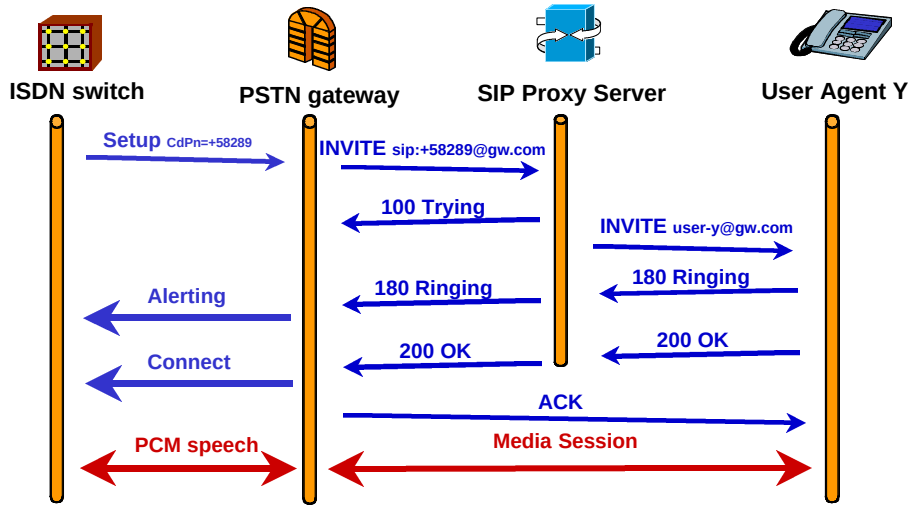


More SIP call flows

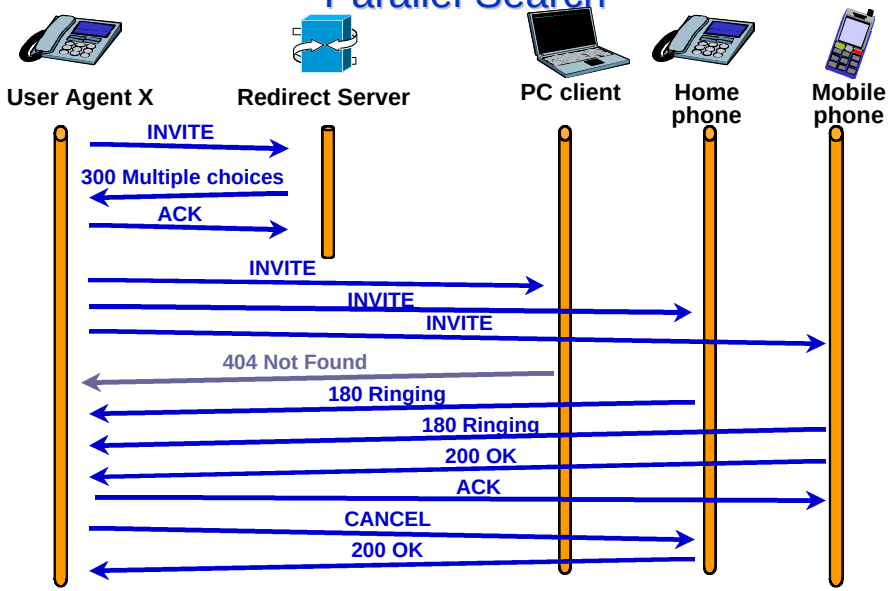
SIP call with Auth., Proxy, and Record Route



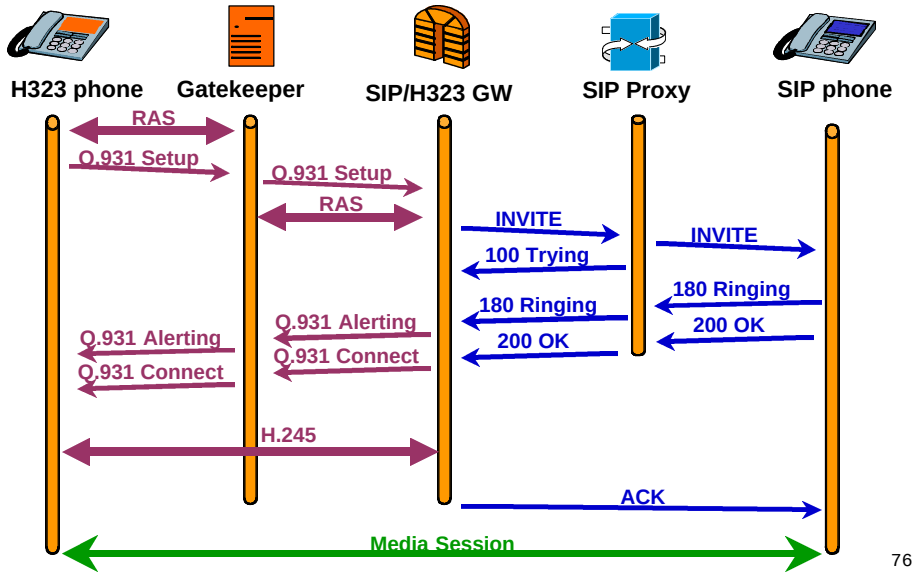
PSTN to SIP Call Through Gateway



Parallel Search



H.323 to SIP Call



SIP protocol design: robustness

- SIP is designed to be robust against several failures
  - **no state in proxy/server during call (cf H323 GK)**
  - **responses are self routing**
  - **subsequent requests and retransmission can take different paths (backup server)**
  - **proxy servers can "loose" memory any time**
  - **UDP means less state than TCP, no time-wait**

SIP and QoS

- SIP does not provide QoS support
- QoS is coupled with SIP through the notion of preconditions
- Objective is to ensure that resources are made available before telephone rings
- Invitation may indicate in SDP that QoS is mandatory
- Call setup should only proceed after satisfying the preconditions
- A SIP extended method (COMET) indicates the success or failure of the preconditions (not yet RFC…)

SIP Implementation: MjSIP

- Java SIP Stack
  - **open-source con licenza GPL**
  - **stack SIP completo, conforme alla RFC 3261 e compatibile all'indietro con RFC 2543**
  - **implementa gli stessi strati e API di JAIN-SIP + CC (Call Control)**
  - **sono disponibili implementazioni di riferimento di User Agent, Registrar, Stateless e Stateful Proxy, SBC (Session Border Controller), etc.**
  - **leggero**
    - gira su Standard VM, J2ME CDC PersonalProfile, J2ME CLDC1.1 MIDP2.0
    - su PC, PDA, smartphone
- http://www.mjsip.org

SIP through Firewalls / NAT

Firewall Traversal

- Firewalls (static)
  - **protect networks by enforcing a restrictive packet filtering policy**
  - **frequently deployed in corporate networks**
  - **policy permits flows from and to trusted addresses**
- Internet telephony (dynamic)
  - **signaling conveys dynamic addresses and port numbers**
  - **3-rd party call control**
  - **user mobility**
- Problems
  - **direct client-to-client signaling through static firewalls**
  - **dynamic media packet flows through static firewalls**
- Note
  - **changing policy within firewall seriously changes security model → not a valid solution!**

Network Address Translator (NAT)

- Used for:
  - **overcome the problem of address depletion with IPv4 (waiting for IPv6..)**
  - **simplicity of network reconfiguration (when changing the POP/ISP)**
  - **avoidance of routing table explosion**
  - **security**
- Basic NAT:
  - **translation from private addresses to public addresses**
- NAPT (Network Address and Port Translation):
  - **translation from private addresses/ports to public addresses/ports**
  - **widely used (generally called NAT): only one IP address for the whole network**
- Need of ad-hoc Application Level Gateways (ALGs) for specific application (ex. FTP, IRC, real audio, etc.)

NAT/NAPT

- Need of ad-hoc Application Level Gateways (ALGs) for specific application
  - **ALGs look for private addresses/ports within the IP datagram payloads**
  - **set translation tables**
  - **translate the address/port information**
- Advantages:
  - **all private IP addresses, address reuse**
  - **only one public address**
  - **no application proxies, the applications work trasparently**
  - **security**
- Disadvantages:
  - **ALGs required (sometimes)**
  - **addresses are not visible from the public Internet**

SIP through firewall

- Main issue: use of NAT boxes within firewalls
  - **problem with NAT/NAPT**
- caused by the use of local address and port information within the SIP messages
  - **SIP headers (via, from, to, contact, etc.)**
  - **SIP body (SDP)**
- Solutions:
  - **eliminating NAPT by introduction of IPv6 (solving the address depletion, but security?)**
  - **use of a SIP-ALG (Application Level Gateway) for NAT**
  - **STUN and TURN**
  - **use of a RTP-proxy (Media Gateway) → Session Border Controller**