Session Initiation Protocol

- SIP is a signaling protocol standardized by the IETF; it is used together with other protocols such as Session Description Protocol (SDP), Real Time Streaming Protocol (RTSP) and Session Announcement Protocol (SAP)
- SIP handles the setup, tear down e management of IP multimedia sessions
- SIP usually adopts RTP as a transport for media streams
- SIP signaling has a dedicated logical channel, different from the logical channels used for the transport of media on the user plane

SIP entities

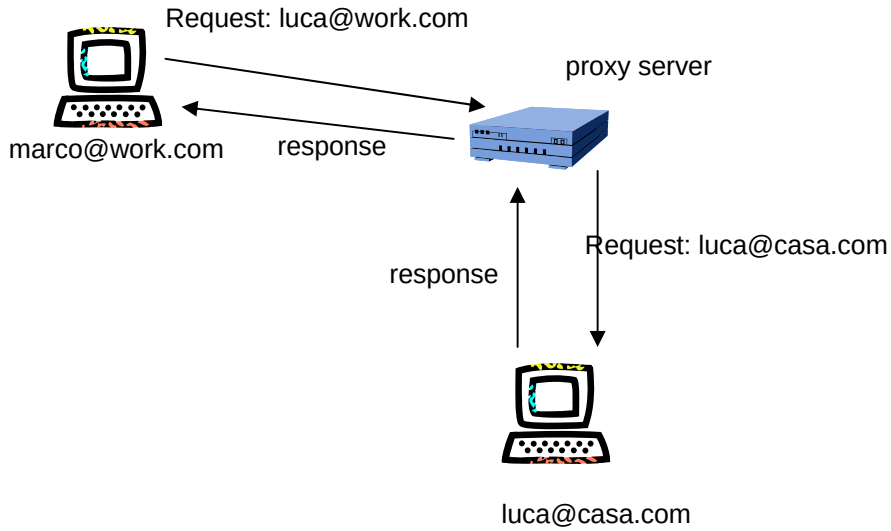
- SIP defines two types of entity:
  - client
    - Called also *user agent client*, it is an applications sending SIP requests
  - servers
    - Applications committed to the establishment of connections; serves respond to SIP requests from clients
- A SIP call involves at least two remote entities
  - user agent client
  - user agent server

SIP entities

- A SIP device can host both user agent client and server, this is the typical case of a user's device
- There are four basic types of SIP server
  - proxy server
  - redirect server
  - registrar server
  - user agent server
- A physical device can host multiple servers

Proxy (redirect) server

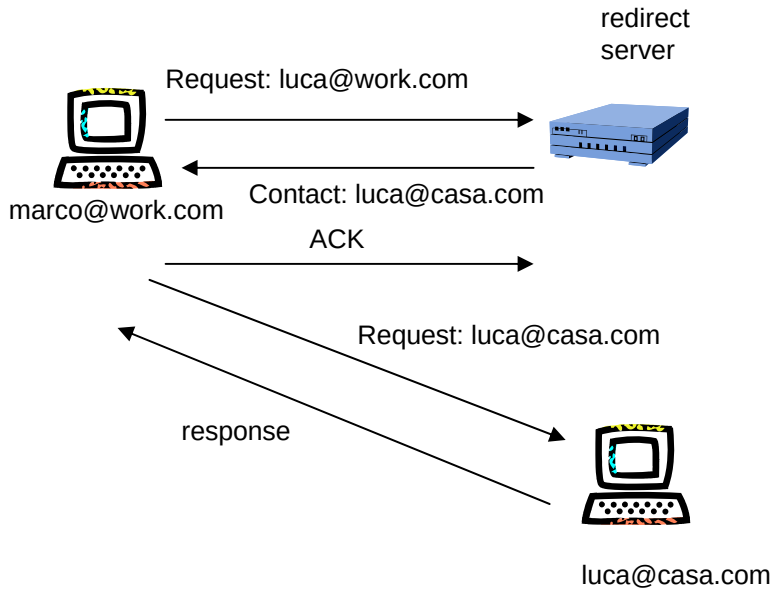
- A proxy server can implement multiple types of SIP server
- It can colloquiate both with user devices and other SIP servers in the network
- Through proxy servers it is possible to implement many value-added services such as
  - call forwarding
  - time-of-day routing



The example shows a simple use case of redirect service.
Luca is at home and Marco calls him at work.
The proxy redirects the call to the current Luca's location

Proxy (redirect) server

- The redirect server can also provide the calling terminal with an alternative address to call, instead of redirecting the call automatically



SIP applications (example)

- Find-Me-Follow-Me:
  - By using a web interface, a user can specify where calls should be forwarded to him, also in a time-dependent fashion
  - The user can ask the system to ring concurrently multiple terminals, or to scan a list of terminals sequentially
- With IVR (interactive voice response):
  - incoming calls can be redirected to an automatic responder
- Click-to-Dial:
  - It is a web browser-based application by which SIP calls are triggered by clicking an hyperlink in a web page
- Contact Manager
  - Access to a centralized contact list, from any user terminal and independently on the current geographic location of the user

User agent server

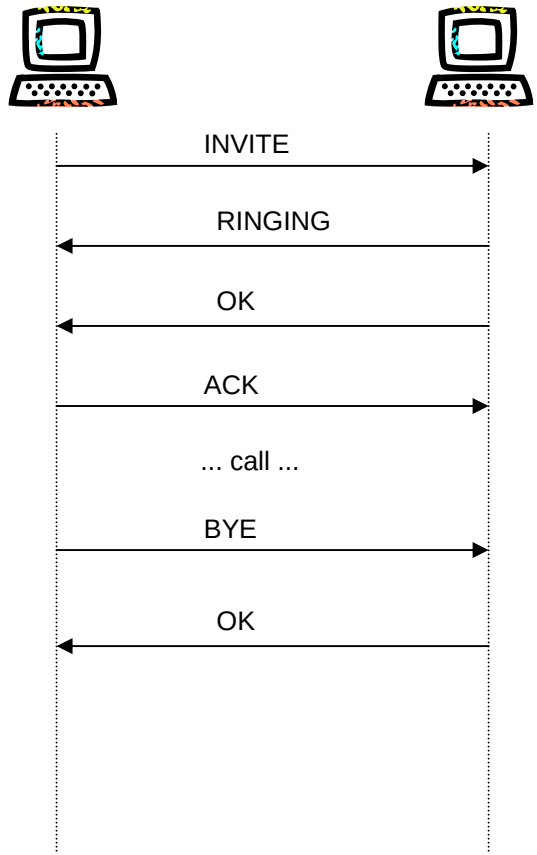
- It is the server usually installed on user terminals
- User terminals host also the user agent client
- The client is used to send signaling requests
- The user agent server receives incoming requests and responds

Registrar server

- The registrar server handles SIP REGISTER requests
- SIP REGISTER requests are used by devices to register into the SIP network
- After a successful registration, a SIP device is operative

SIP base call without proxy

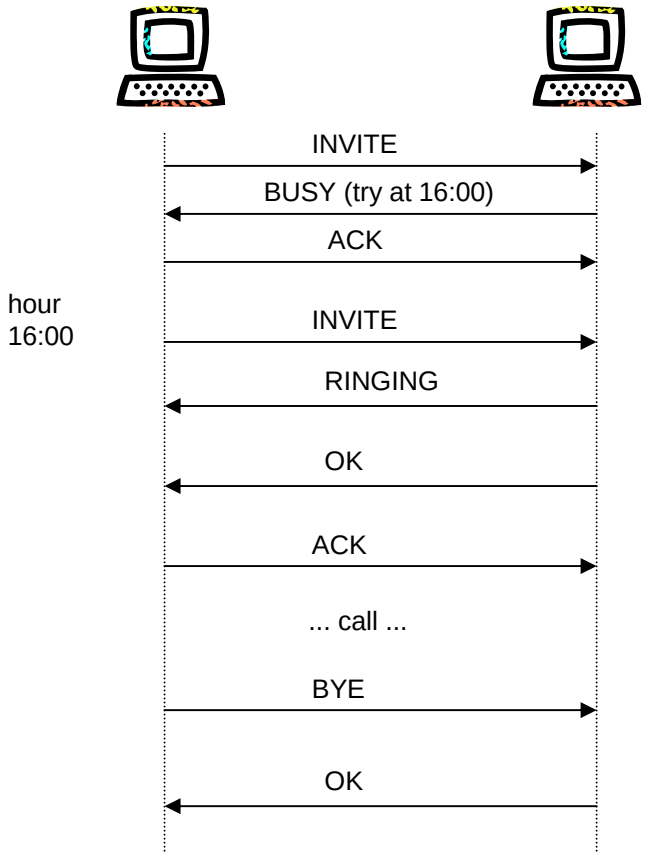
- The INVITE message is used to start a call
- In the simplest case shown in the figure, the user agent client sends the INVITE directly to the called device
- The remore user server sends a RINGING message to let the calling device know that the called terminal is ringing
- When the call is accepted at the called end, the OK message is sent
- The call is teared down with the BYE/OK pair of messages



SIP features

Caratteristiche di SIP

- SIP is a simple and flexible signaling protocol
- The parameters for media negotiation are carried in the body of SIP messages
- The figure shows an example of deferred call



SIP messages

- SIP messages can be either requests or responses
- SIP messages are in textual format
- In this way, it is simpler to analyze the behavior of signaling just parsing exchanges IP packets
- This is an advantage over binary signaling protocols
- However, textual messages usually consumes more bandwidth than binary protocols

```
message = start-line
          *message header
          CRLF
          [message-body]
```

```
start-line = request-line || status-line
```

SIP messages

- The request-line specifies the type of request and its parameters
- The status line specifies the type of response and its parameters
- The message-header carries information such as source and destination identification

```
message = start-line
          *message header
          CRLF
          [message-body]
```

```
start-line = request-line || status-line
```

SIP messages

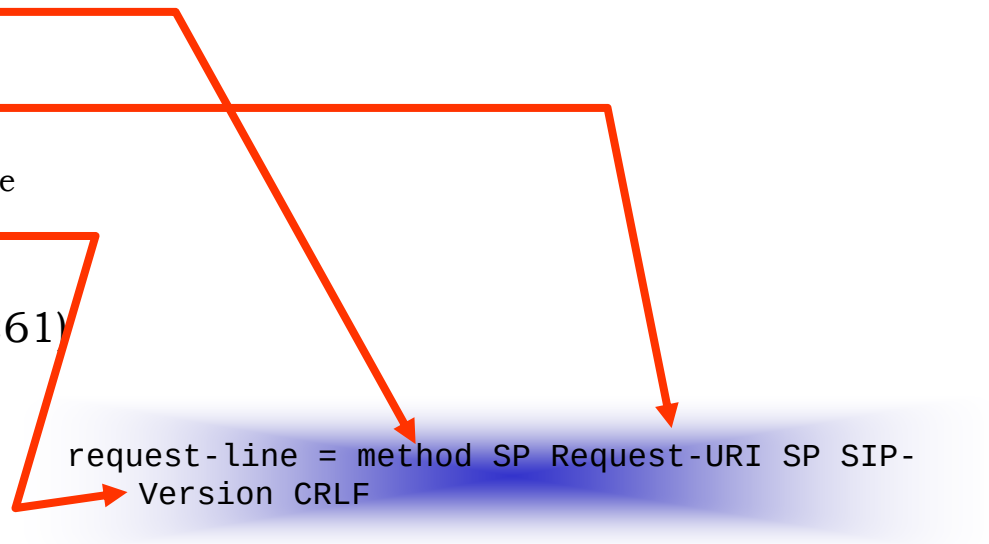
- The message-body carries information such as
  - type of session
  - format of media
  - altra caratterizzazione
- SIP does not standardize the message body; in the case of media format information, the structure of the message body is standardized by the Session Description Protocol (SDP)
- The message body can also carry messages of other sugnaling systems: this is useful for signaling interworking

```
message = start-line
          *message header
          CRLF
          [message-body]
```

```
start-line = request-line || status-line
```

SIP requests

- SIP requests start with the request-line, consisting of blank-separated tokens:
  - method
    - Type of requests
  - Request-URI
    - Destination address of the request
  - SIP-Version
- Basic methods are (RFC 3261)
  - INVITE
  - ACK
  - OPTIONS
  - BYE
  - CANCEL
  - REGISTER

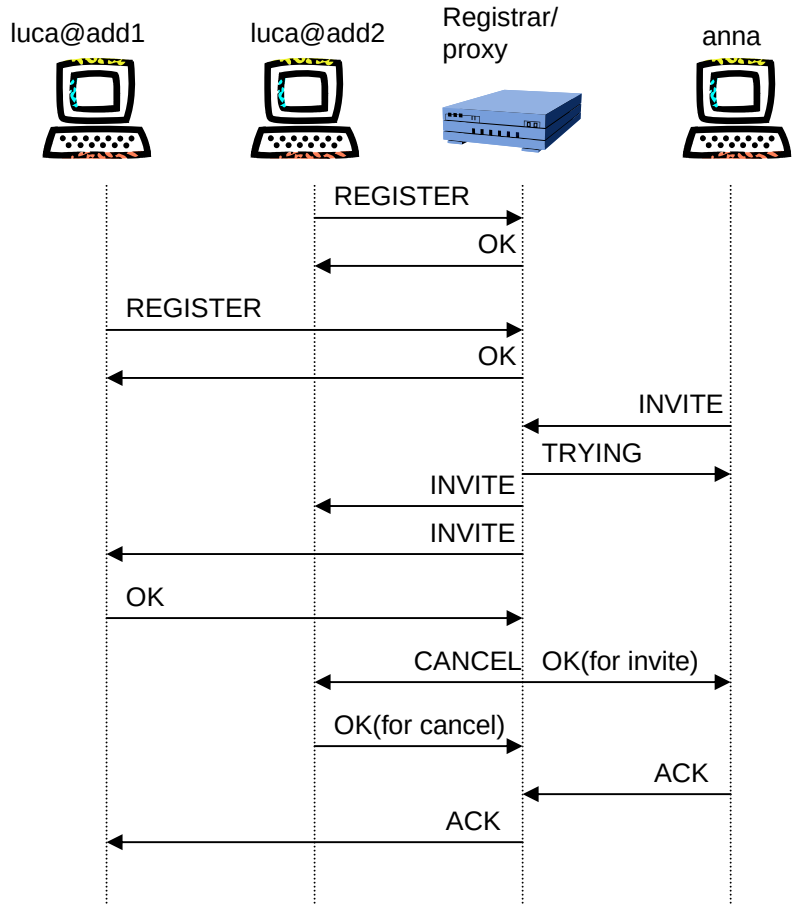


SIP requests

- INVITE
  - To establish a call; it specified involved users and also the format of media
- ACK
  - Messages are usually ACKed; the ACK message must refer explicitly to the message it acknowledges
- BYE
  - To close a call
- CANCEL
  - To terminate an ongoing transaction
- REGISTER
  - Used to register a device

Example: one-number

- Luca is registered with multiple SIP addresses
- Anna calls Luca: the call is forwarded to all active registrations of Luca
- When one of the called terminal responds, the call is started



Additional methods

- The INFO method is useful to transmit informnation when a call is already active
- For example, this can be used to transport multifrequency tones

SIP response

- The status code is a three-digit number specifying the type of message
- The reason-phrase is a text field carrying additional information, depending of the type of message
- Status codes range from 100 to 699
- The first digit indicates the category of the rsponse
  - 1xx: provisional (eg: 181 call forwarded)
  - 2xx: success (eg 200)
  - 3XX: redirect (eg, 302: retry calling the address specified in the reason phrase)
  - 4XX: failure
  - 5XX: Server failure
  - 6XX: Global failure (eg: called user does not exist)

```
status-line = SIP version SP status code SP
    reason-phrase CRLF
```

NOTE: all responses, with the exception of the 1XX Provisional responses, must be ACKed

SIP response

- Provisional
  - 100 trying
  - 180 ringing
  - 181 call forwarded
  - 182 call queued
  - 183 session in progress
- Success
  - 200 OK
- Redirection
  - 301 user moved permanently
  - 302 user mover temporarily
- Request failure
  - 400 bad request
  - 401 user not authorized
  - 415 unsupported media
  - 486 busy
- Server failure
  - 500 server internal error
- Global failure
  - 604 does not exist anywhere

```
status-line = SIP version SP status code SP
    reason-phrase CRLF
```

SIP addressing

- SIP addresses are URIs (Uniform Resource Indicator)
  - SIP URI: sip:luca@home.milano.it

Message headers

- There are multiple header types
- To e From headers identify called and calling users
- Header can be:
  - general header
  - request header
  - response header
  - entity header

Message headers

- General header
  - They are used for both requests and response, e.g.: To, From ,Call ID …
  - The Contact header is used for the redirection of calls
- Request Header
  - Used only for requests
- Response Header
  - Used only for responses

Message headers

- Entity Header
  - The main purpose of the entity header is to describe the content of the message body
  - Frequently used fieds are
    - Content-length (*content length in bytes*)
    - Content-type (e.g. *application/SDP*)
    - Content-encoding
    - Content-disposition
      - To specify, for example, if the contents of the body must be displayed for the user
    - Content-language

Registration example

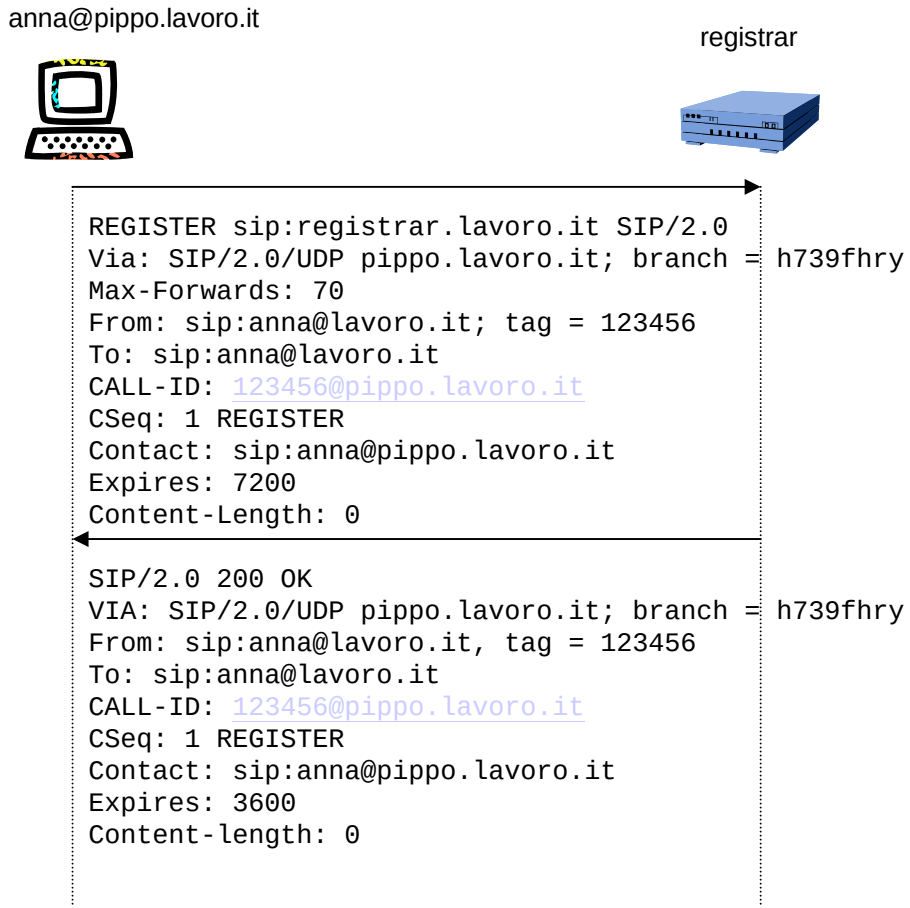
- Via header
  - The Via header specifies the SIP route followed by the message
  - The branch parameter is used to identify univocally the message, if proxies are used
- From header
  - Specifies the URI of the requesting entity
  - The tag field is used to help identifying univocally the user and user's transaction
- To: header
  - In this case, the To header coincides with the From header: this is to specify that Anna is registering herself

Registration example

- CALL-ID header
  - To help identifying univocally the registration transaction
- Cseq header
  - Has the purpose of mapping correctly responses to requests
- Contact header
  - Specifies who is to be contacted to deliver responses
- In this case, the message body is empty, thus we have
  - Content-Length: 0

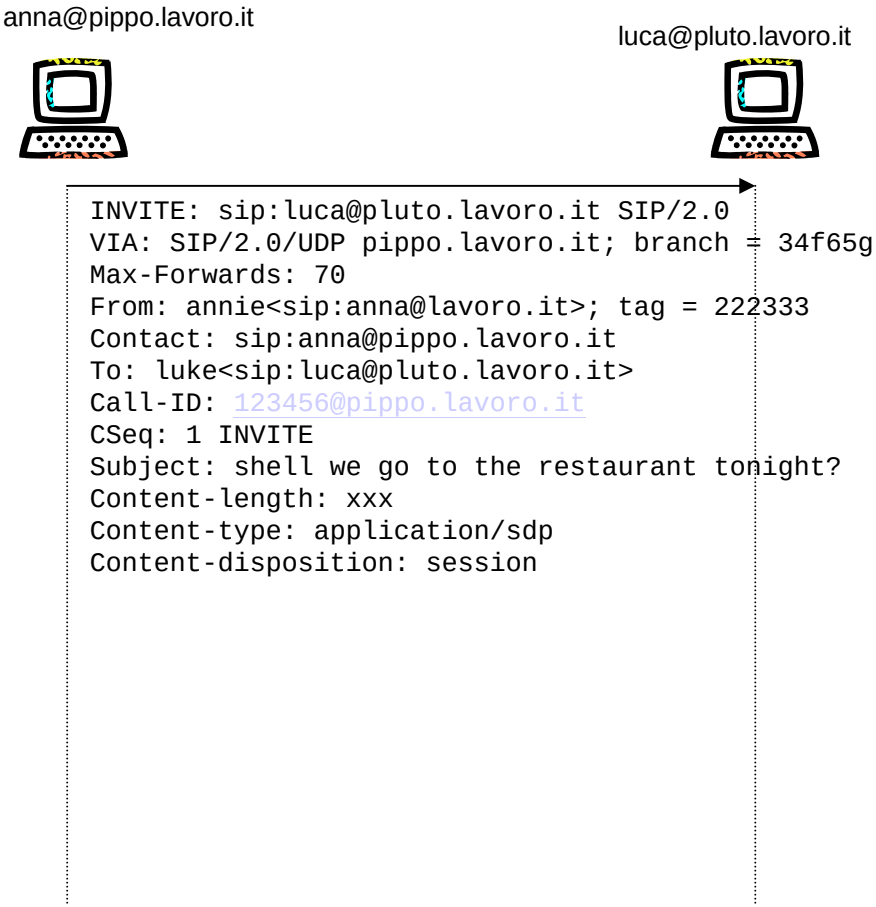
Registration example

- The figure plots a possible response to the registration request



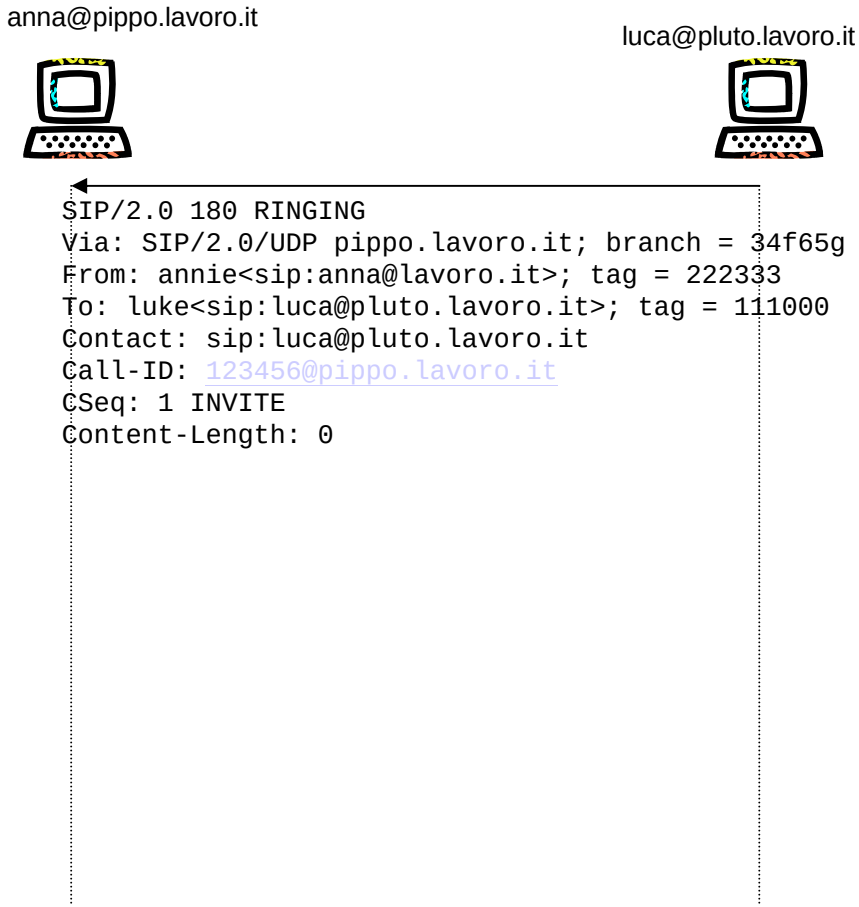
INVITE

- This example shows a direct call, thus the complete address of the destination host must be specified
- In case of proxy servers, the host indication can be omitted as it can be solved automatically by the proxy
- The nicks *annie* e *luke* are the “display names”, they are optional



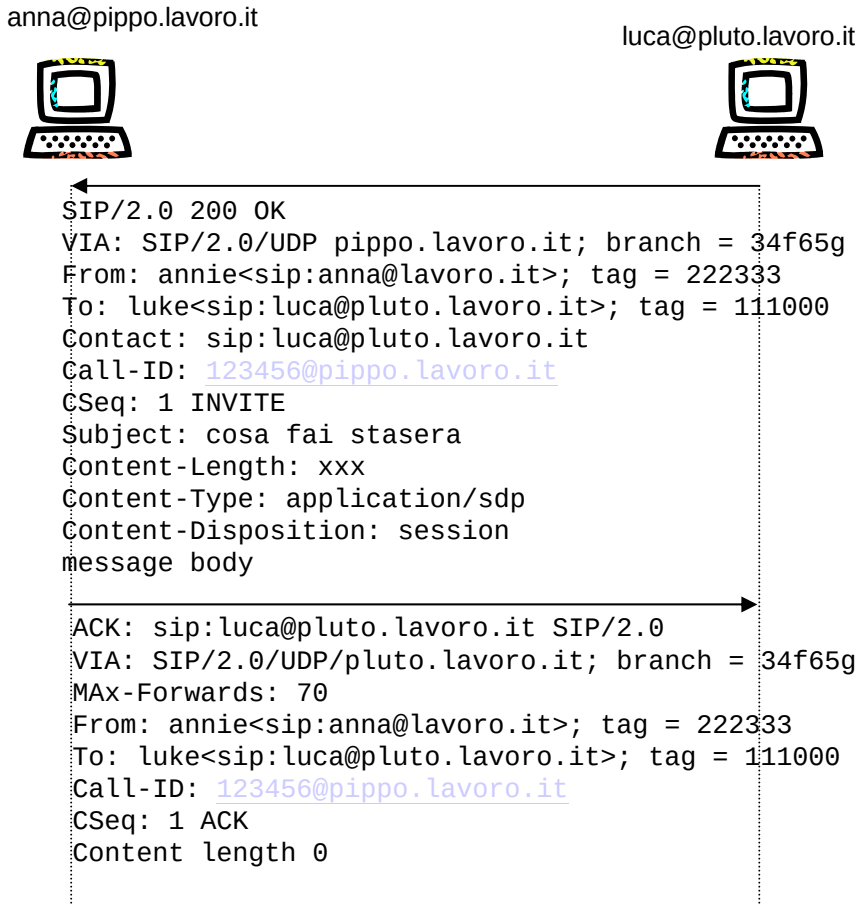
Response to INVITE

- The triplet
  - **(From, To, Call-ID)** is the “dialog ID” which identifies a transaction between two remote user terminals
  - At this point of the signaling transaction, the dialog is established but the call is not set up; the dialog is in the “early” status
  - The dialog becomes “confirmed” when the call is set up

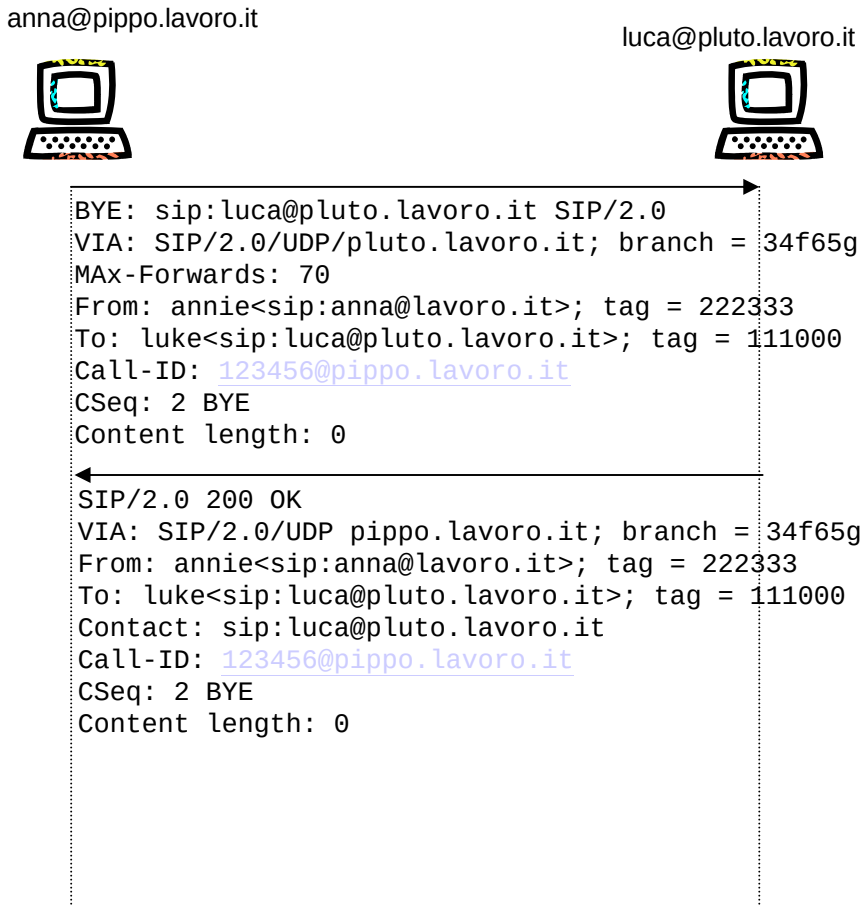


Confirmation to INVITE

- With the OK message the call is established and the dialog is “confirmed”



Tear down

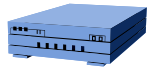

anna@pippo.lavoro.it
luca@pluto.lavoro.it
BYE: sip:luca@pluto.lavoro.it SIP/2.0
VIA: SIP/2.0/UDP/pluto.lavoro.it; branch = 34f65g
MAx-Forwards: 70
From: annie<sip:anna@lavoro.it>; tag = 222333
To: luke<sip:luca@pluto.lavoro.it>; tag = 111000
Call-ID: 123456@pippo.lavoro.it
CSeq: 2 BYE
Content length: 0
SIP/2.0 200 OK
VIA: SIP/2.0/UDP pippo.lavoro.it; branch = 34f65g
From: annie<sip:anna@lavoro.it>; tag = 222333
To: luke<sip:luca@pluto.lavoro.it>; tag = 111000
Contact: sip:luca@pluto.lavoro.it
Call-ID: 123456@pippo.lavoro.it
CSeq: 2 BYE
Content length: 0

Proxy server

- Proxy serversroute SIP messages to the proper destinations
- More than one proxy can be inviolved in the setup of a connection
- For example, the setup of a SIP call may involve two proxies: the local proxy of the calling user and the local proxy of the called user

Example (proxy)

BOSS<sip:manager@pc1.home.net> sip:server.work.com daniel<sip:collins@station1.work.com>


```
INVITE sip:collins@work.com SIP/2.0
Via: SIP 2/0/UDP pc1.work.net; branch z9hG4bK7890
Max-Forwards: 70
From: boss<sip:manager@home.net>;tag=ab12
Contact: boss<sip:manager@pc1.home.net>
To: daniel<sip:collins@work.com>
Call-ID: 123456@pc1.home.net
CSeq: 1 INVITE
```

```
INVITE sip:collins@work.com SIP/2.0
Via: SIP 2/0/UDP server.work.com; branch z9hG4bKxyz1
Via: SIP 2/0/UDP pc1.work.net; branch z9hG4bK7890
Record-route: <sip:server.work.com;lr>
Max-Forwards: 69
From: boss<sip:manager@home.net>;tag=ab12
Contact: boss<sip:manager@pc1.home.net>
To: daniel<sip:collins@work.com>
Call-ID: 123456@pc1.home.net
CSeq: 1 INVITE
```

```
SIP/2.0 100 Trying
Via: SIP 2/0/UDP pc1.work.net; branch z9hG4bK7890
From: boss<sip:manager@home.net>;tag=ab12
To: daniel<sip:collins@work.com>
Call-ID: 123456@pc1.home.net
CSeq: 1 INVITE
```

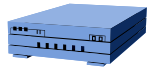
```
SIP/2.0 200 OK
Via: SIP 2/0/UDP server.work.com; branch z9hG4bKxyz1
Via: SIP 2/0/UDP pc1.work.net; branch z9hG4bK7890
Record-route: <sip:server.work.com;lr>
From: boss<sip:manager@home.net>; tag=ab12
To: daniel<sip:collins@work.com>; tag=xy45
Call-ID: 123456@pc1.home.net
CSeq: 1 INVITE
Contact: sip:collins@station1.work.com
```

```
SIP/2.0 200 OK
Via: SIP 2/0/UDP pc1.work.net; branch z9hG4bK7890
Record-route: <sip:server.work.com;lr>
From: boss<sip:manager@home.net>; tag=ab12
To: daniel<sip:collins@work.com>; tag=xy45
Call-ID: 123456@pc1.home.net
CSeq: 1 INVITE
Contact: sip:collins@station1.work.com
```

Example (proxy)

BOSS<sip:manager@pc1.home.net>

sip:server.work.com



daniel<sip:collins@station1.work.com>

```
ACK sip:collins@station1.work.com SIP/2.0
Via: SIP 2/0/UDP pc1.work.net; branch z9hG4bK7891
Max-Forwards: 70
Route: <sip:server.work.com;lr>
From: boss<sip:manager@home.net>; tag=ab12
To: daniel<sip:collins@work.com>; tag=xy45
Call-ID: 123456@pc1.home.net
CSeq: 1 ACK
```

```
ACK sip:collins@station1.work.com SIP/2.0
Via: SIP 2/0/UDP server.work.com; branch z9hG4bKxyz2
Via: SIP 2/0/UDP pc1.work.net; branch z9hG4bK7891
Max-Forwards: 69
From: boss<sip:manager@home.net>; tag=ab12
To: daniel<sip:collins@work.com>; tag=xy45
Call-ID: 123456@pc1.home.net
CSeq: 1 ACK
```

Comments

- The Via header is important when proxies are involved in signaling
- Each SIP device touched by a SIP message adds a Via header carrying its address
- In this way, the path of SIP devices crossed by a message is explicitly written in the message header
- It is easy to detect SIP routing loops
- Moreover, it is easy to force response messages to follows in the reverse direction the path followed by request messages

Comments

- In the backward path of responses, Via header are stripped from the message at each SIP hop
- The function of the branch parameter is to identify univocally the signalling transaction at each SIP entity crossed by messages
- A stateful proxy must be always in the path followed by all requests and all responses
- The Via header is not enough, because its scope is limited to one request-response interaction
- This is why also the record-route header is used