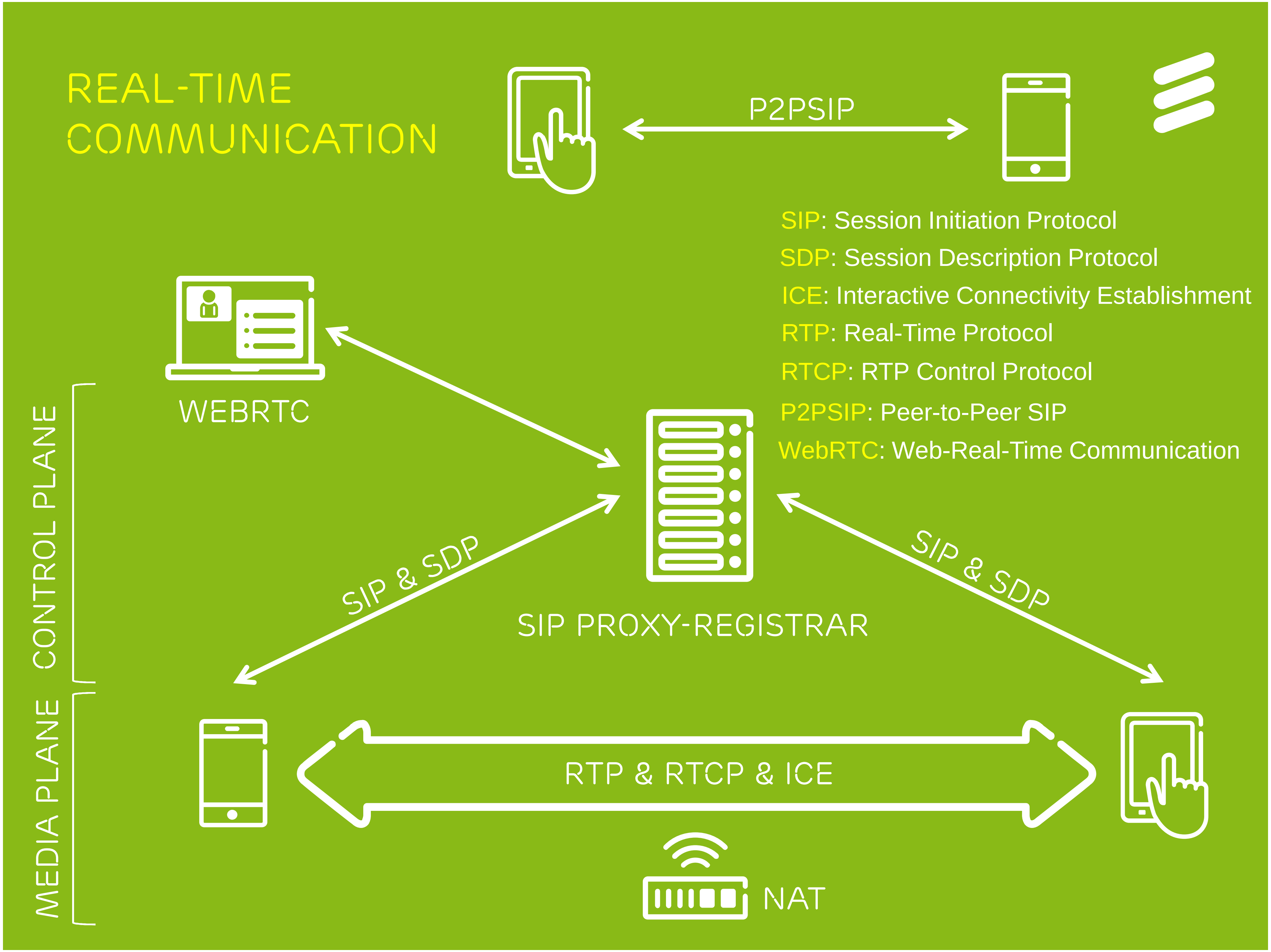
SESSION INITIATION PROTOCOL (SIP) OVERVIEW

T-110.5150
APPLICATIONS AND SERVICES IN INTERNET
OCTOBER 15TH, 2013

Jouni Mäenpää, Dr.Sc.(Tech)
NomadicLab
Ericsson Research Multimedia Technologies
Communication Services

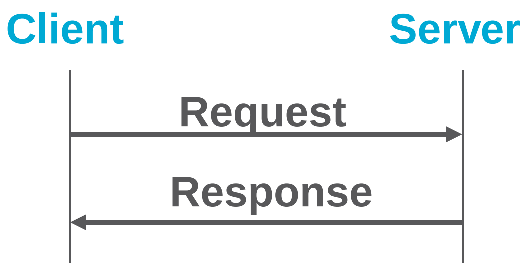
AGENDA

- SIP introduction, history and functionality
- Key concepts of SIP
- SIP addresses
- SIP messages
- SIP registrations
- SIP routing
- The Session Description Protocol (SDP)
- Real-Time Protocol (RTP)
- RTP Control Protocol (RTCP)


REAL-TIME
COMMUNICATION
P2PSIP
SIP: Session Initiation Protocol
SDP: Session Description Protocol
ICE: Interactive Connectivity Establishment
RTP: Real-Time Protocol
RTCP: RTP Control Protocol
P2PSIP: Peer-to-Peer SIP
WebRTC: Web-Real-Time Communication
WEBRTC
CONTROL PLANE
SIP & SDP
SIP & SDP
SIP PROXY-REGISTRAR
MEDIA PLANE
RTP & RTCP & ICE
NAT

SIP GENERAL OVERVIEW

- Session Initiation Protocol (**SIP**)
  - Application-level
  - End-to-end
  - Client-server
  - Extensible
  - Text based



- Designed by Internet Engineering Task Force (**IETF**)
- Design base: **HTTP** and SMTP
- Mainly used to
  - Establish multimedia sessions (e.g., VoIP)
  - Modify multimedia sessions
  - Terminate multimedia sessions
- SIP messages are either requests or responses
  - Carry zero or more “bodies”.
  - Session Description Protocol (**SDP**) is the common body
- Runs on any transport protocol (UDP, TCP, TLS, SCTP)

HISTORY OF SIP

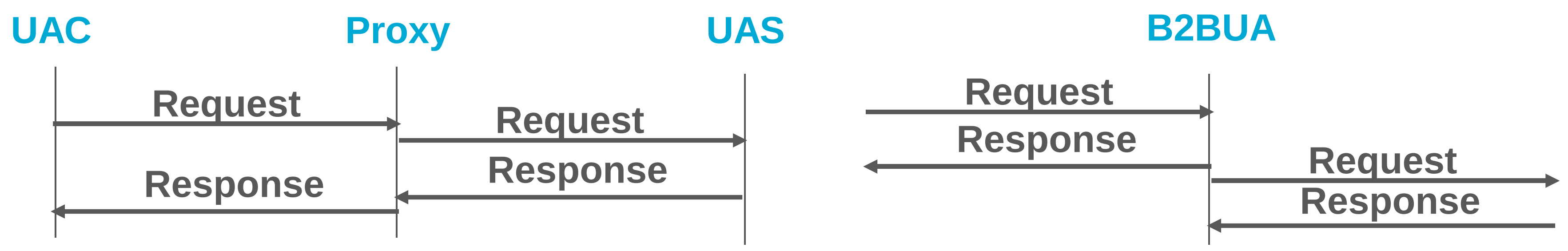
- Specified in the Internet Engineering Task Force (IETF)
- February 1996: Session *Invitation* Protocol (SIPv1)
  - SIPv1 used Session Description Protocol (SDP)
  - Text-based
  - UDP-based
- February 1996: Simple Conference Invitation Protocol (SCIP)
  - New format for session descriptions
  - Based on HTTP
  - TCP-based
- March 1996: Presentations at the 35th IETF meeting
- December 1996: Session *Initiation* Protocol (SIPv2)
  - Merged SIPv1 and SCIP
  - Based on HTTP
  - UDP and TCP
  - SDP
- December 1997: decision to split SIP into a base spec and extensions
- February 1999: proposed standard level
  - Published as RFC 2543
- June 2002: RFC 3261 was published
- 2013: Work continues in the SIPCORE and DISPATCH WGs

OVERVIEW OF SIP FUNCTIONALITY

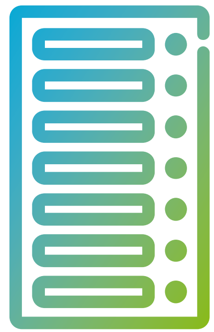
- Functionality
  - User **location** (not geographical location)
    - End system used for communication
  - User **availability**
    - Willingness of the other party to engage in communications
  - User **capabilities**
    - Media parameters
  - Session **set-up**
    - Establishment of session parameters at both called and calling party
  - Session **management**
    - Transfer and termination of sessions, modifying session parameters
- SIP does not provide services
  - But it enables the system to provide services
  - It has been demonstrated that it is easy to provide services with SIP

SIP LOGICAL ENTITIES

- User Agent (UA): An endpoint
  - User Agent Client (UAC): sends requests, receives responses
  - User Agent Server (UAS): receives requests, sends responses
- Proxy server: A network host that proxies requests and responses, i.e., acts as a UAC and as a UAS.
- Registrar: A special UAS that accepts only registrations
- Redirect server: a UAS that redirects request to other servers.
- Back-to-back User Agent (B2BUA): UAS linked to a UAC
  - Acts as a UAS and as a UAC linked by some application logic

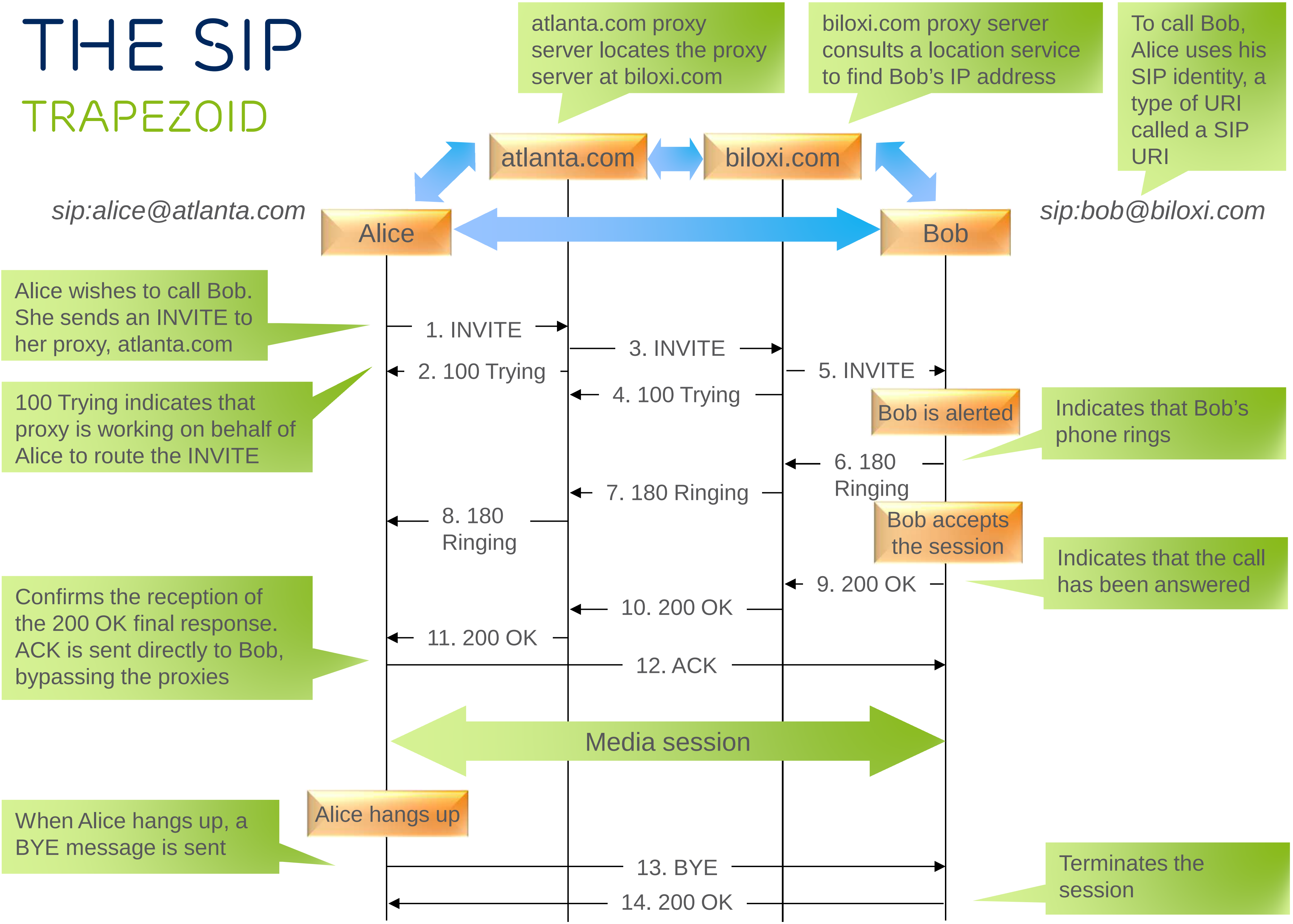


STATELESS AND STATEFUL PROXIES



- There are several types of SIP proxies, depending on the state they keep
- **Stateless proxy**
  - Does not keep any state when forwarding requests and responses
  - A simple message forwarder
- **Transaction stateful proxy**
  - Stores state during the duration of the transaction
  - Maintains a server transaction and a client transaction
- **Call stateful proxy**
  - Stores all the state pertaining to a session (e.g., from INVITE to BYE)
  - A call stateful proxy is always a transaction stateful proxy, but not the other way round

THE SIP
TRAPEZOID



SIP METHODS

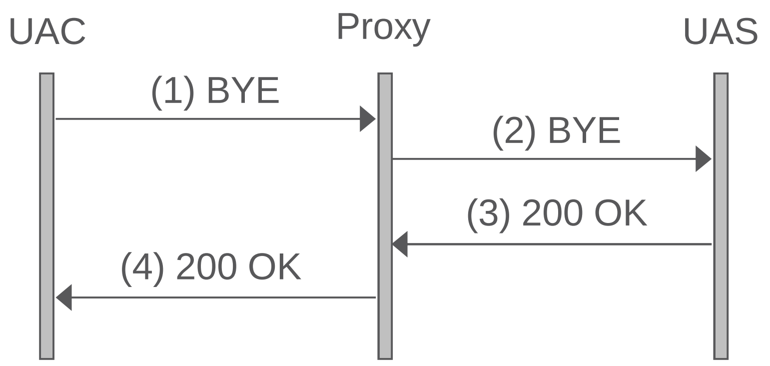
- INVITE
- BYE
- ACK
- CANCEL
- REGISTER
- OPTIONS
- SUBSCRIBE
- NOTIFY
- PUBLISH
- MESSAGE
- REFER
- PRACK
- UPDATE
- INFO

SIP ADDRESSES

- SIP uses Uniform Resource Identifiers (URIs)
  - SIP URIs and SIPS URIs
  - Others (such as TEL URL) also commonly supported.
- Examples
  - sip:john.doe@example.com
  - sips:john.doe@example.com
  - tel:+358-9-299-3283
  - sip:proxy.atlanta.com:5060
  - sip:another-proxy.biloxi.com;transport=UDP
- SIP and SIPS URIs
  - Must include a host name
  - May include username, port numbers, parameters
  - sip:user:password@host:port;uri-parameters
- Non SIP/TEL URIs are also valid under certain circumstances: IM, PRES

SIP TRANSACTIONS (1/2)

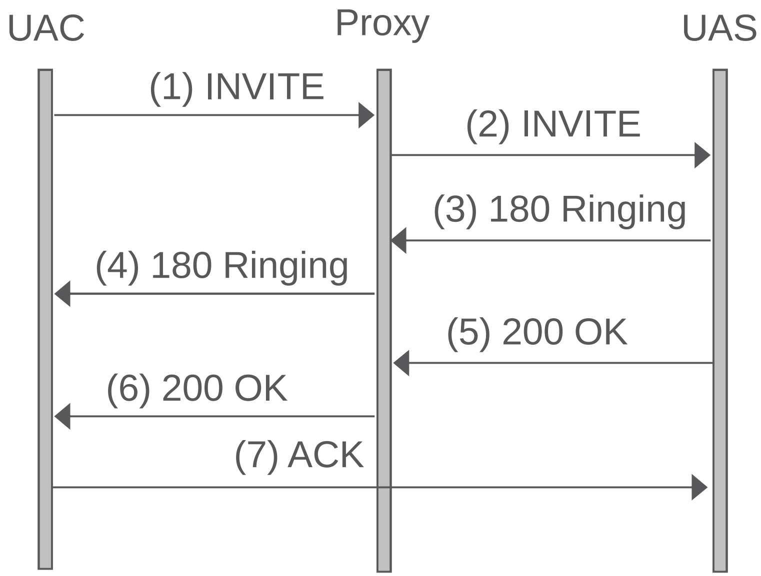
- SIP **transaction**
  - Occurs between a client and a server
  - Consists of a request and at least one response
  - Comprises all messages from the first request sent up to a **final response**
  - May contain zero or more **provisional responses** before the last final response
- Three types of transactions
  - Regular transactions: other than INVITE, ACK or CANCEL
  - INVITE-ACK transactions
  - CANCEL transactions

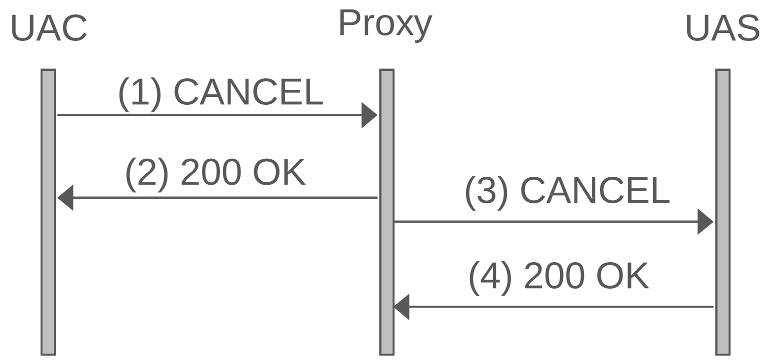


Regular transaction

SIP TRANSACTIONS (2/2)

- An INVITE-ACK involves two transactions: an INVITE transaction and an ACK transaction
  - The ACK request confirms the reception of the final response

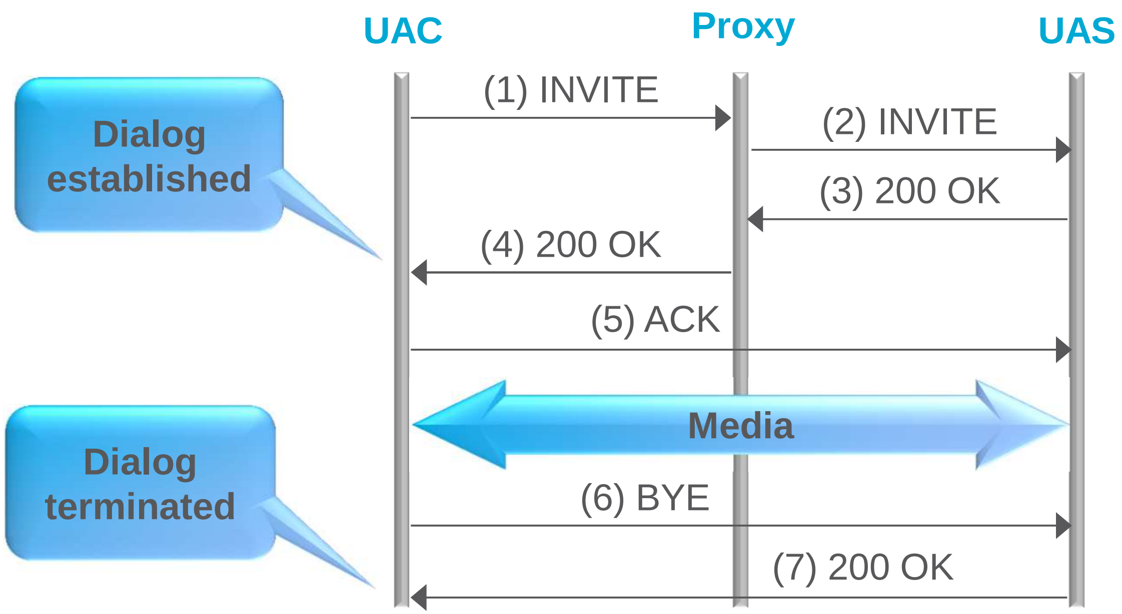




- A CANCEL transaction cancels a previous transaction
  - Connected to a previous transaction
  - Similar to regular transactions
  - Exception: final response generated by the next SIP hop (proxy) instead of the UAS

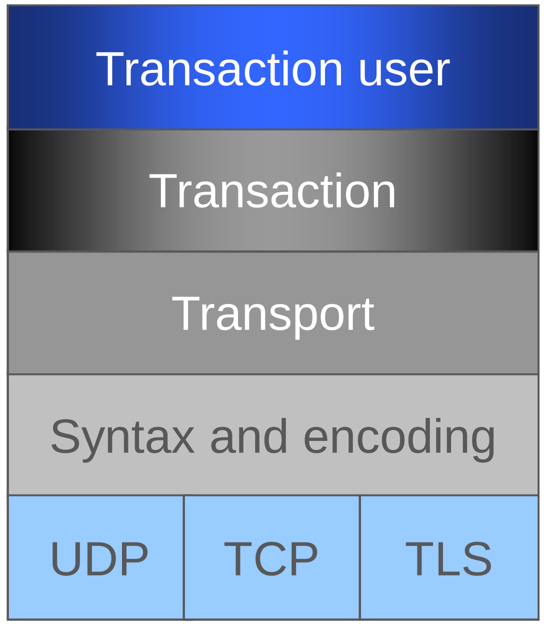
SIP DIALOGS

- A dialog is a SIP relationship between two endpoints that persists for some time
- SIP methods that can create a dialog include INVITE, SUBSCRIBE and REFER
  - When a dialog is established, all the subsequent requests within that dialog follow the same path



FUNCTIONAL LAYERS

- SIP is structured as a layered protocol
- Syntax and encoding layer
  - Message parsing
  - Encoding is specified using an augmented Backus-Naur Form grammar (BNF)
    - E.g. SIP-URI = "sip:" [userinfo] hostport
- Transport layer
  - Defines how
    - a UAC sends requests and receives responses
    - a UAS receives requests and sends responses
- Transaction layer
  - Handles application layer retransmissions, matching responses to requests, and application-layer timeouts
- Transaction user (TU)
  - Session creation, application-specific processing
  - When a TU wishes to send a request, it creates a client transaction instance and passes it the request along with the destination IP address, port and transport



AN EXAMPLE OF A SIP REQUEST

Request Line

```
INVITE sip:John.Doe@example.com SIP/2.0
```

Header

```
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd];branch=z9hG4bKnashds7
Max-Forwards: 70
Route: <sip:pcscf1.visited1.net;lr>, <sip:scscf1.home1.net;lr>
From: <sip:user1_public1@home1.net>;tag=171828
To: <sip:John.Doe@example.com>
Call-ID: cb03a0s09a2sdfglkj490333
Cseq: 127 INVITE
Contact: <sip:[5555::aaa:bbb:ccc:ddd]>
Content-Type: application/sdp
Content-Length: 248
```

Body

```
v=0
o=- 2987933615 2987933615 IN IP6 5555::aaa:bbb:ccc:ddd
s=-
c=IN IP6 5555::aaa:bbb:ccc:ddd
t=907165275 0
m=audio 3458 RTP/AVP 97 96 0 15
a=rtpmap:97 AMR
a=fmtp:97 mode-set=0,2,5,7; maxframes=2
a=rtpmap:96 G726-32/8000
```

Method

Version

Request-URI

Header Field

Header Field Name

Header Field Value

AN EXAMPLE OF A SIP RESPONSE

Status Line

SIP Version

Reason phrase

```
SIP/2.0 200 OK
```

Header

```
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd];branch=z9hG4bKnashds7
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From: <sip:user1_public1@home1.net>;tag=171828
To: <sip:John.Doe@example.com>;tag=314159
Call-ID: cb03a0s09a2sdfglkj490333
CSeq: 127 INVITE
Contact: <sip:[5555::eee:fff:aaa:bbb]>
Content-Type: application/sdp
Content-Length: 220
```

Status code

Body

```
v=
o=- 2987933615 2987933615 IN IP6 5555::eee:fff:aaa:bbb
s=-
c=IN IP6 5555::eee:fff:aaa:bbb
t=907165275 0
m=audio 3458 RTP/AVP 97 0
a=rtpmap:97 AMR
a=fmtp:97 mode-set=0,2,5,7; maxframes=2
```

SIP RESPONSES (1/2)

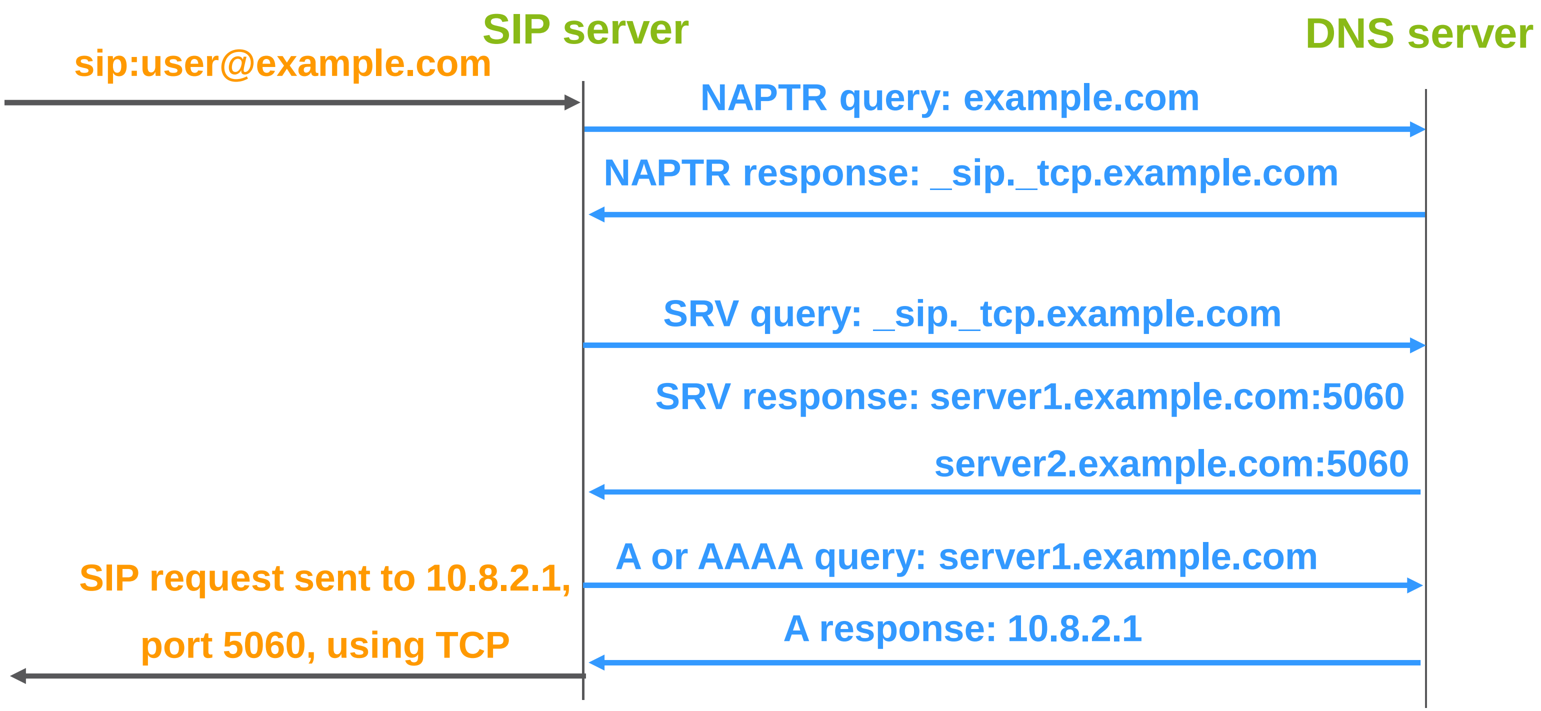
- SIP defines two types of responses
  - **Final responses** convey the result of the request processing, and are sent reliably
  - **Provisional responses** provide information on the progress of the request processing, but are not sent reliably in the core protocol (RFC 3261)
- Status codes ranges:
  - 100 – 199 **Provisional** (also known as informational responses)
    - Server is performing some further action and does not yet have a definitive response
    - Example: 180 Ringing – UA receiving the INVITE is trying to alert the user
  - 200 – 299 **Success**
    - Request was successful
    - Example: 200 OK – the request has succeeded
  - 300 – 399 **Redirection**
    - 3xx responses give information about the user's new location or about alternative services that might be available to satisfy the call
    - Example: 302 Moved temporarily – retry the request at new address(es) specified in the Contact header

SIP RESPONSES (2/2)

- Status code ranges continued:
  - 400 – 499 **Client error**
    - Definitive failure responses from particular server
    - Client should not retry the same request without modification
    - Example: 401 Unauthorized – request requires user authentication
  - 500 – 599 **Server error**
    - Server itself is the cause of the error
    - Example: 500 Internal server error – server encountered an unexpected condition
  - 600 – 699 **Global failure**
    - Server has definitive information about a particular user
    - Example: 600 Busy everywhere – the callee is busy and knows that no other end system will be able to accept the call

SIP ROUTING AND DNS

- SIP clients use **DNS** to route requests and find the next hop to route the request
  - By looking into a **NAPTR** (Naming Authority Pointer) record in DNS
  - By looking into a **SRV** (Services) record in DNS
  - By looking into **A** (IPv4) or **AAAA** (IPv6) records in DNS
- Example
  - Assumption: no transport and no port specified in the SIP URI

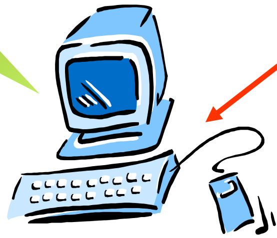


SIP REGISTRATIO

Request-URI names the domain for which the registration is meant.
To contains the address of record (AoR) whose registration is to be created.
From contains the AoR of the person responsible for the registration.

Public user identity *sip:bob.doe@biloxi.com* is bound to the contact address *sip:bob@laptop.biloxi.com*

1 REGISTER sip:biloxi.com SIP/2.0
From: <sip:bob.doe@biloxi.com>
To: <sip:bob.doe@biloxi.com>
Contact: <sip:bob@laptop.biloxi.com>



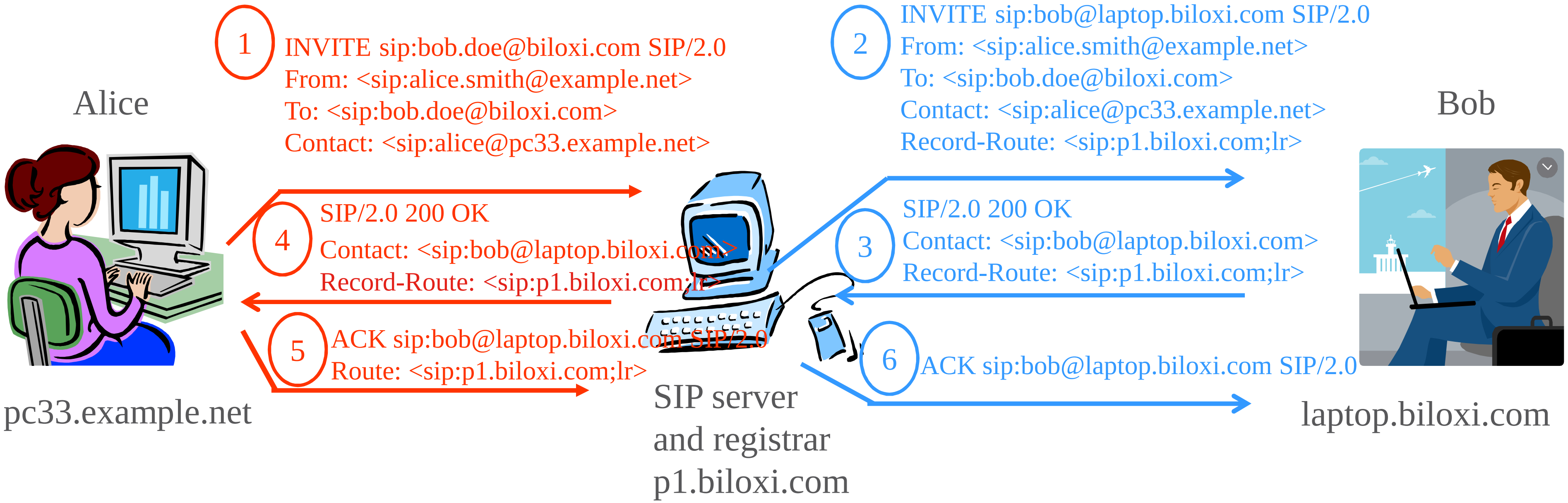
SIP server
and registrar

2 SIP/2.0 200 OK
From: <sip:bob.doe@biloxi.com>
To: <sip:bob.doe@biloxi.com>
Contact: <sip: bob@laptop.biloxi.com>

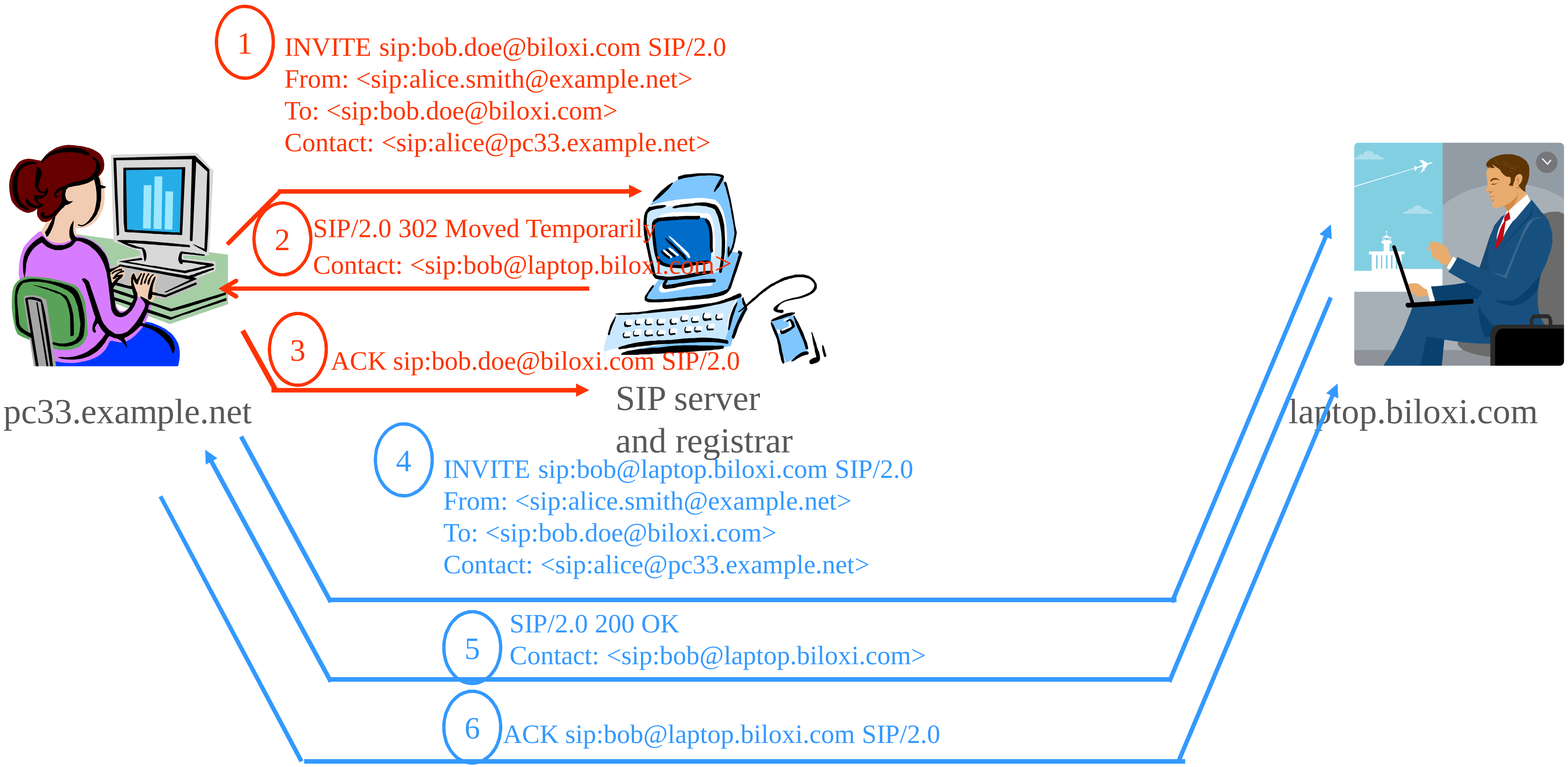
laptop.biloxi.com

- The SIP registration function allows users to upload their current locations for use by proxy servers
  - A REGISTER message associates a user’s SIP (or SIPS) URI with the machine into which the user is currently logged
  - The registrar writes this association into a database, from which it can be fetched by a proxy server

ROUTING: SIP SERVER IN PROXY MODE

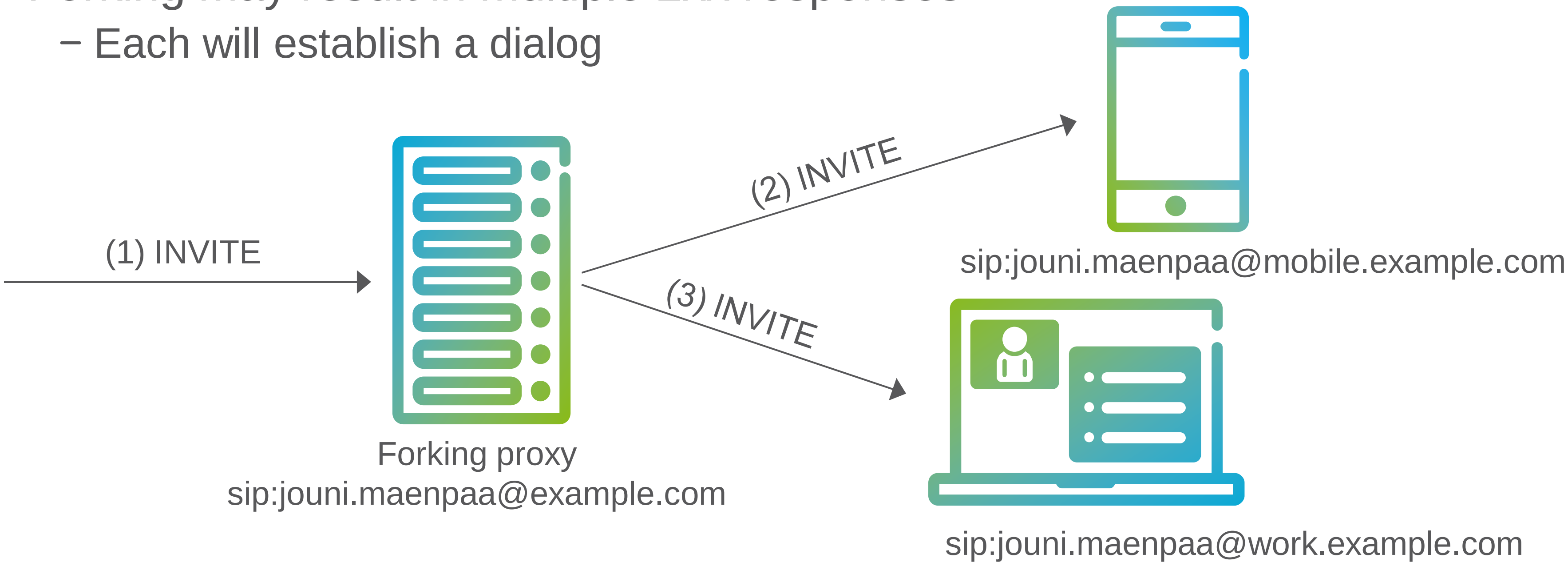


ROUTING: SIP SERVER IN REDIRECT MODE



FORKING PROXIES

- A proxy server can send an INVITE to a number of locations at the same time
  - This type of parallel search is known as **forking**
- A proxy can route messages in parallel or in sequence
  - In **parallel** forking, all locations are attempted simultaneously
  - In **sequential** forking, the proxy tries different locations one after the other
- Forking may result in multiple 2xx responses
  - Each will establish a dialog



SESSION DESCRIPTION PROTOCOL (SDP)

- **Session Description Protocol (SDP)** is the most common format to describe multimedia sessions
  - The details of the session to be established using SIP are not described using SIP, but by using SDP
- SDP is a textual format used to describe the set of
  - Media streams
  - Codecs
  - Other media related parameters supported by either party
- All SIP implementations MUST support SDP
  - Although they can support other bodies
- Used by other protocols than SIP: RTSP, SAP, etc.

SDP EXAMPLE

Session level	Version --->	v=0
	Origin --->	o=alice 2987933615 2987933615 IN IP4 10.0.0.8
	Session Name -->	s=training course
	URI -->	u=http://standards.example.com/sip
	E-mail address -->	e=john.doe@example.com
	Phone number -->	p=+358-9-123-4567
	Connection Data -->	c=IN IP4 10.0.0.8
	Times -->	t=907165275 0
Media level	Media -->	m=audio 3458 RTP/AVP 97 0
	Attributes -->	a=rtpmap:97 AMR
	Attributes -->	a=fmtp:97 mode-set=0,2,5,7; maxframes=2
	Media -->	m=video 3460 RTP/AVP 98
	Attributes -->	a=rtpmap:98 H263
	Media -->	m=application 32416 udp wb

IP address where media is received

Codecs

Type of media streams

Port number where media is received

THE OFFER/ANSWER MODEL

- SDP was initially developed to support multicast sessions
  - For a unicast session, two addresses are needed
  - Also, the set of codecs needs to be determined by finding an overlap in the set supported by each participant
- SIP provides a two-way session description exchange called the **offer/answer model**
  - Provides the semantics and operational details defining how SDP is used to describe unicast sessions
- By using the offer/answer model, two entities can make use of SDP to arrive at a common view of a multimedia session between them
  - As a result, they learn the formats they can use and the transport addresses for the session
- In the model
  - One participant called the **offerer** generates a session description (the **offer**), and sends it to the remote user (the **answerer**)
  - The answerer generates a new session description (the **answer**) and sends it to the offerer

OFFER/ANSWER MODEL – EXAMPLE 1

- Alice sends an offer to Bob:

```
v=0
o=Alice 2790844676 2867892807 IN IP4 192.0.0.1
s=Let's discuss
c=IN IP4 192.0.0.1
t=0 0
m=audio 20000 RTP/AVP 0
a=sendrecv
m=video 20002 RTP/AVP 31
a=sendrecv
```

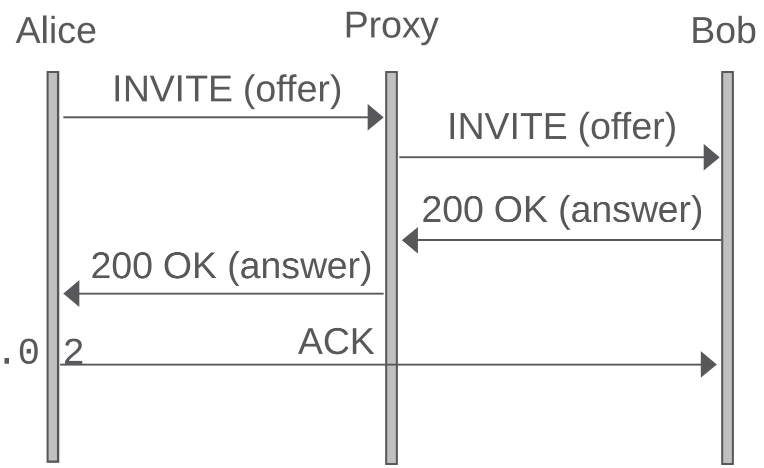
Payload type 0 refers to u-law PCM coded audio sampled at 8 kHz

Payload type 31 refers to H.261 video.

- Bob sends his answer to Alice:

```
v=0
o=Bob 234562566 236376607 IN IP4 192.0.0.2
s=Let's discuss
c=IN IP4 192.0.0.2
t=0 0
m=audio 30000 RTP/AVP 0
a=sendrecv
m=video 30002 RTP/AVP 31
a=sendrecv
```

Bob supports the same video and audio codecs as Alice does.



- As a result, Alice and Bob can have a video conversation

OFFER/ANSWER MODEL – EXAMPLE 2

- Multiple codecs are offered, but only one is accepted
- Alice sends an offer to Bob:
  - Alice offers three audio codecs (PCMU, PCMA and iLBC) and H.261 and MPV video

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.atlanta.example.com
s=
c=IN IP4 host.atlanta.example.com
t=0 0
m=audio 49170 RTP/AVP 0 8 97
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
m=video 51372 RTP/AVP 31 32
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
```

- Bob’s select PCMU audio and drops the video component

```
v=0
o=bob 2808844564 2808844564 IN IP4 host.biloxi.example.com
s=
c=IN IP4 host.biloxi.example.com
t=0 0
m=audio 49174 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 0 RTP/AVP 31
a=rtpmap:31 H261/90000
```

Bob is willing to send and receive PCMU audio

Zero port number indicates a rejected stream

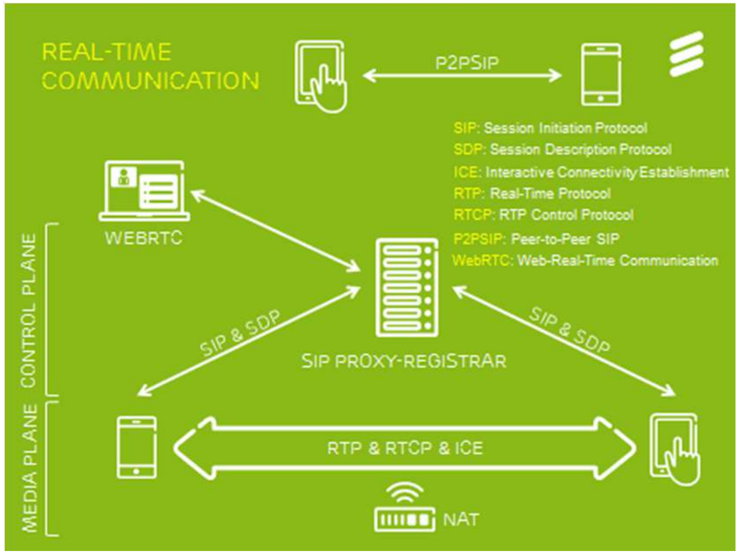
For a rejected stream, at least one media format must be present

WHERE IS SIP USED TODAY?

- 3G IP Multimedia Subsystem (IMS)
  - Voice over Long Term Evolution (VoLTE)
- SIMPLE (SIP Instant Messaging and Presence Leveraging Extensions)
- SIP VoIP/IM clients (some examples)
  - Pidgin (cross-platform)
  - Jitsi (cross-platform)
  - KPhone (Linux)
  - Sipdroid (Android)
  - Linphone (PCs, Android, iOS)
  - Etc.
- Enterprise telephony
  - SIP IP-PBXs
- SIP trunking
  - Interconnecting private enterprise domain to public domain to provide PSTN/PLMN interworking
- Apple FaceTime (iOS)
- Skype
  - Skype Connect – receive Skype calls on and initiate from office phones
- Microsoft Lync Server
- WebRTC (Web Real-Time Communication)

REAL-TIME PROTOCOL (RTP)

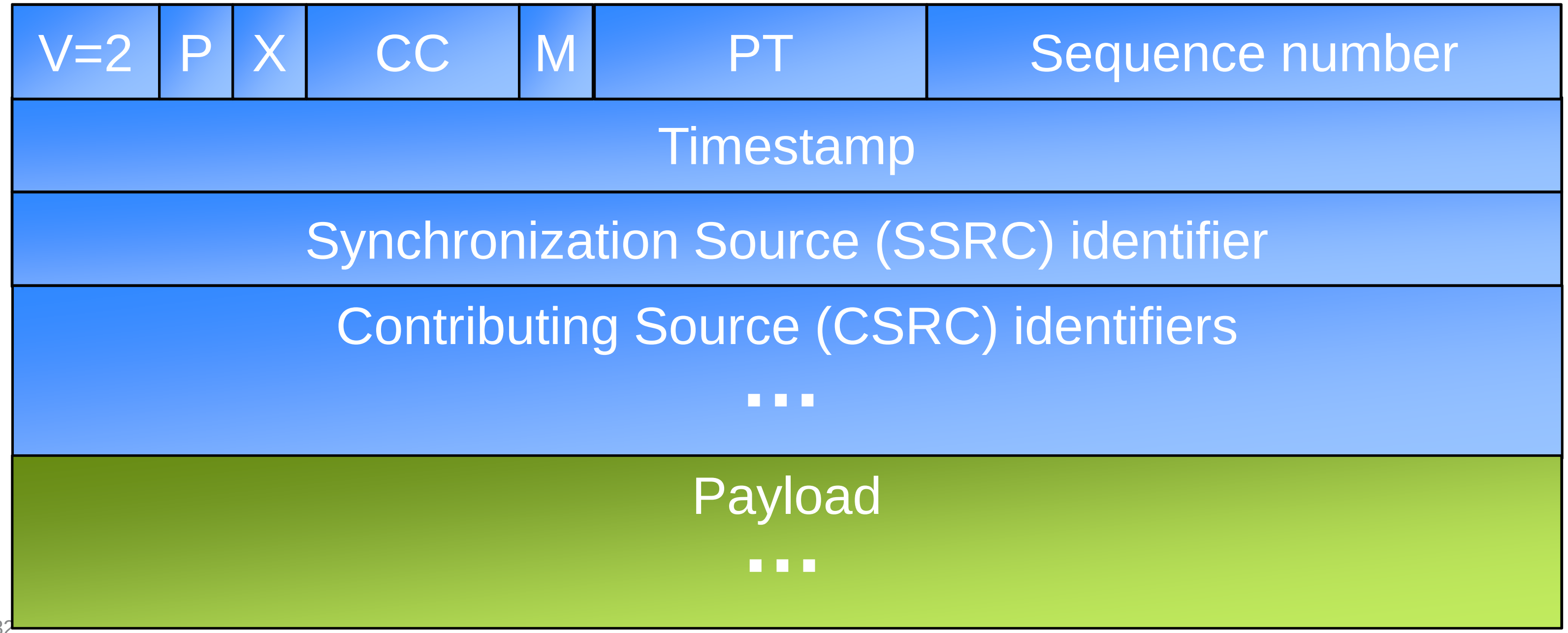
- **RTP** - protocol for transmitting real-time data such as audio and video
  - Primary standard for audio/video transport in IP networks
  - Defines a packet format for delivering audio/video over IP
  - Together with SIP&SDP, one of the core protocols for VoIP
- **RTCP** – RTP Control Protocol
  - Monitoring transmission statistics and quality of service (QoS) feedback
  - Synchronization of streams



- **RTP profiles**
  - RTP can carry a wide range of multimedia formats
  - Information about specific formats does not need to be carried in RTP header
    - Instead, it is provided through profiles and payload formats
  - Profile defines codecs used to encode the payload and their mapping to payload types
  - A profile is accompanied by several payload format specifications
  - Examples of profiles
    - RTP/AVP – profile for audio and video conferences with minimal control
    - Secure RTP (SRTP)
  - Payload formats: G.711, G.729, MP3, OPUS, H.264, VP8

RTP HEADER

- V - RTP version
- P – Padding (if set, the packet contains padding)
- X – extension (if set, the header is followed by a header extension)
- CC – CSRC identifier count
- M – marker bit (interpretation defined by profile)
- PT – Payload type (format of RTP payload)
- Sequence number – used to detect packet loss
- Timestamp – reflects the sampling instant
- SSRC – identifies the source of the stream
- CSRC – list the contributing sources (in case of multiple sources)



ERICSSON