Session Initiation Protocol (SIP) Overview

T-110.7100
Applications and Services in Internet
6.10.2009

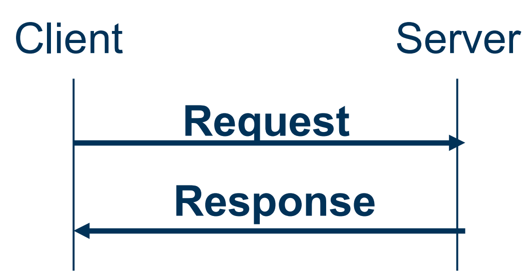
Jouni Mäenpää
NomadicLab, Ericsson

Contents

- SIP introduction, history and functionality
- Key concepts of SIP
- SIP addresses
- SIP messages
- SIP registrations
- SIP routing
- The Session Description Protocol (SDP)
- Extending SIP

SIP General Overview

- SIP is an application-level, end-to-end, client-server, extensible, text based protocol.
- The design base was HTTP and SMTP
- SIP was originally used to establish, modify and terminate multimedia sessions in the Internet (e.g., VoIP)
- SIP messages are either requests or responses.
- SIP messages carry zero or more “bodies”.
- Session Description Protocol (SDP) is the common body for session initiation.
- SIP runs on any transport protocol (UDP, TCP, TLS, SCTP)
  - The spec mandates UDP and TCP. Other transport protocols are optional



History of SIP

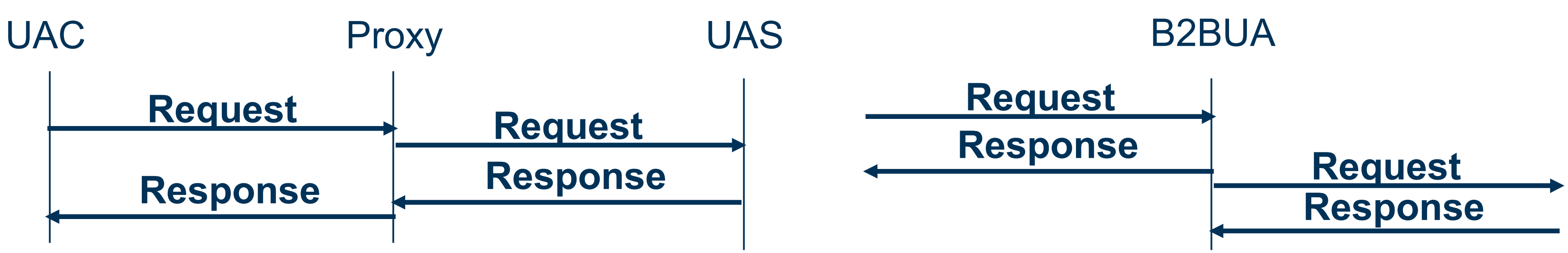
- Specified in the Internet Engineering Task Force (IETF)
- In February 1996, Handley and Schooler submitted an Internet draft on Session *Invitation* Protocol, SIPv1, to the IETF
  - SIPv1 used Session Description Protocol (SDP) to describe sessions
  - Text-based
  - UDP-based transport
- Also in February 1996, Henning Schulzrinne submitted an Internet draft specifying the Simple Conference Invitation Protocol (SCIP) to the IETF
  - SCIP was based on HTTP and was thus text-based
  - Defined its own format for session descriptions
  - TCP-based transport
- Presentations of SIPv1 and SCIP at the 35th IETF meeting in March 1996
- In December 1996, a new draft on Session *Initiation* Protocol (SIPv2) was submitted by Handley, Schulzrinne and Schooler
  - SIPv2 merged SIPv1 and SCIP
  - SIPv2 is based on HTTP, and is thus text-based
  - Uses both UDP and TCP
  - Uses SDP to describe multimedia sessions
- In December 1997, a decision to split SIP into a base spec and extensions
- In February 1999, SIP reached the proposed standard level and was published as RFC 2543
- In June 2002, the current SIP specification, RFC 3261, was published

Overview of SIP Functionality

- SIP provides the following functionality:
  - User location (not geographical location)
    - Determination of the end system to be used for communication
  - User availability
    - Determination of the willingness of the other party to engage in communications
  - User capabilities
    - Determination of the media parameters to be used
  - 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

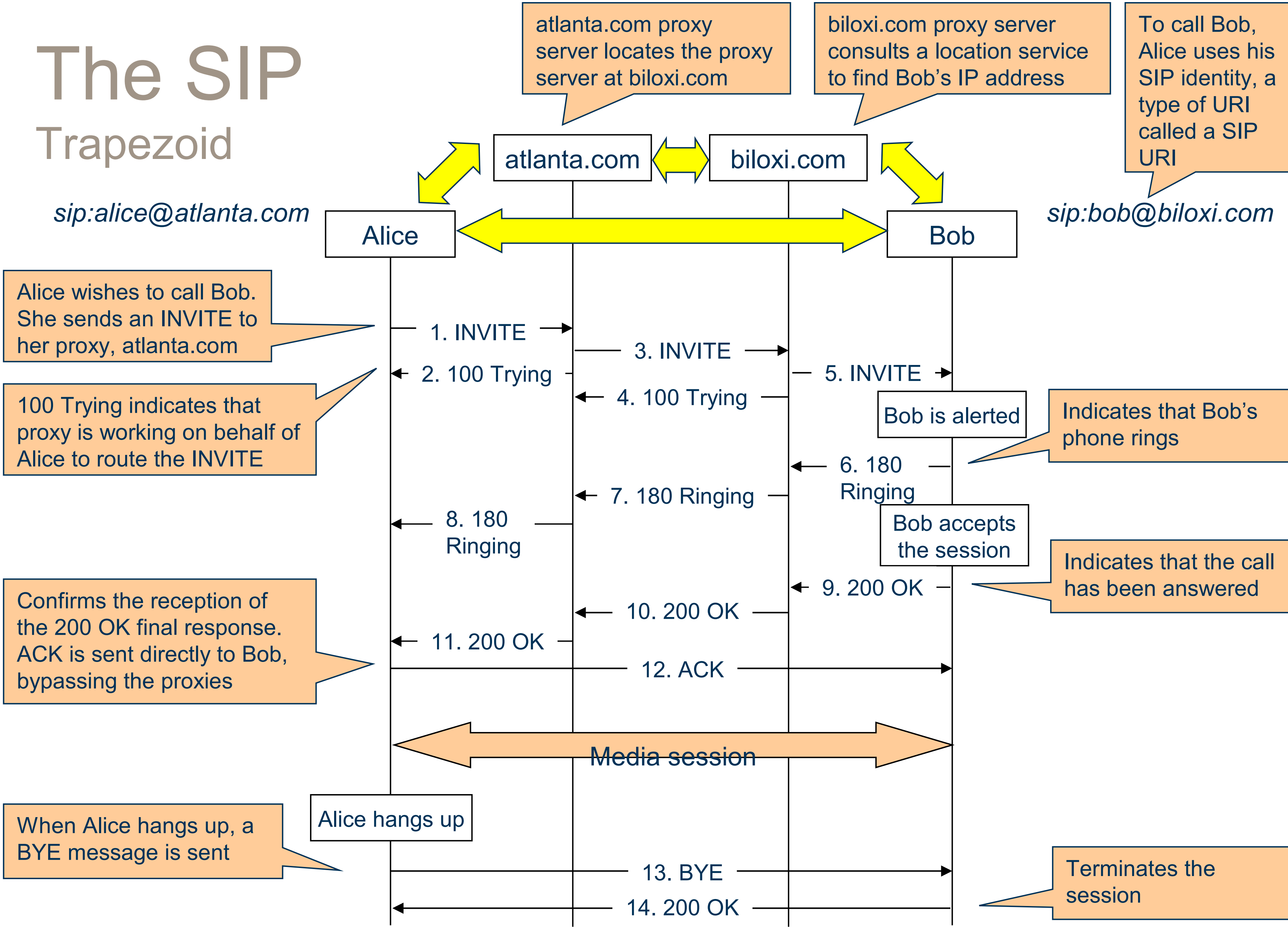
- User Agent (UA): An endpoint
  - User Agent Client (UAC): sends requests, receives responses
  - User Agent Server (UAS): receives requests, sends responses
- Proxy server: A network host that proxies requests and responses, i.e., acts as a UAC and as a UAS.
- Registrar: A special UAS that accepts 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 SIP Proxies

- There are several types of SIP proxies, depending on the state they keep:
- **Stateless proxy:** a proxy that does not keep any state when forwarding requests and responses.
  - A simple message forwarder
- **Transaction stateful proxy, or stateful proxy:** a proxy that stores state during the duration of the transaction.
  - Maintains a server transaction and a client transaction
- **Call stateful proxy:** a proxy that 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

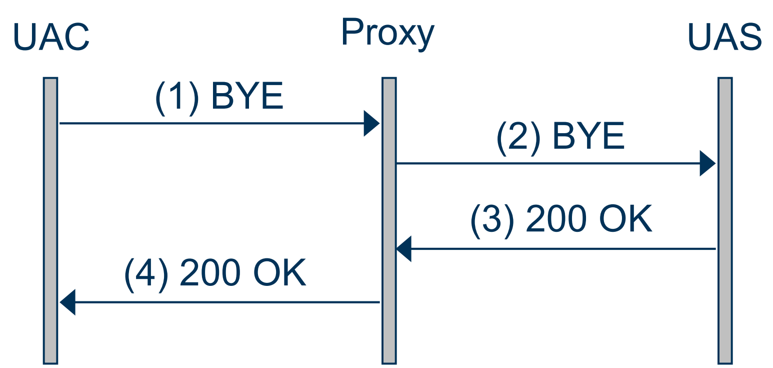
- INVITE: creates a session
- BYE: terminates a session
- ACK: acknowledges a final response for an INVITE request
- CANCEL: cancels an INVITE request
- REGISTER: binds a public SIP URI to a contact address
- OPTIONS: queries a server for capabilities
- SUBSCRIBE: installs a subscription for a resource
- NOTIFY: informs about changes in the state of the resource
- PUBLISH: publication of presence information
- MESSAGE: delivers an instant message
- REFER: used for call transfer, call diversion, etc.
- PRACK: acknowledges a provisional response for an INVITE request
- UPDATE: changes the media description (e.g., SDP) in an existing session
- INFO: used to transport mid-session information

SIP Addresses

- SIP uses Uniform Resource Identifiers (URIs). At least, SIP URIs and SIPS URIs are supported, although others (such as TEL URL) are commonly supported.
  - 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, and may include username, may include port numbers, may include parameters
  - sip:user:password@host:port;uri-parameters
- Non SIP/TEL URIs are also valid under certain circumstances: HTTP, IM, PRES, MAILTO…

SIP Transactions (1/2)

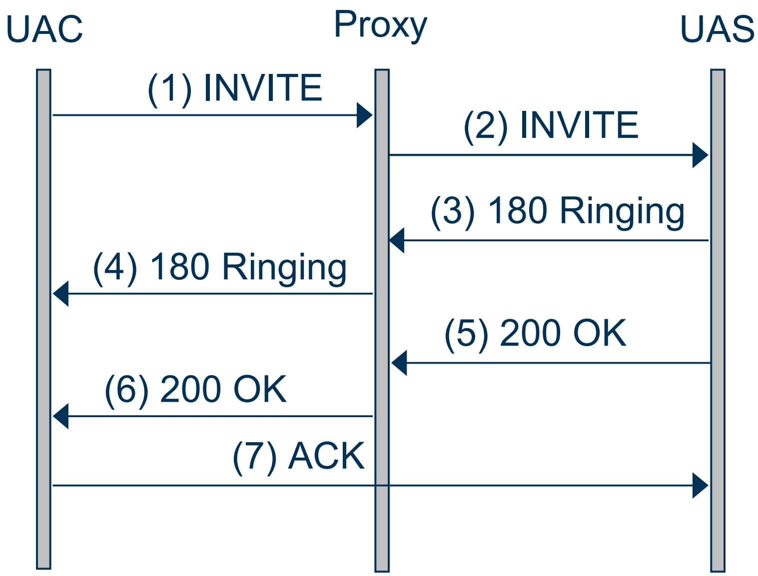
- SIP Transaction
  - Occurs between a client and a server
  - Each transaction consists of a request that invokes a particular method, or function, on the server, and at least one response
  - Comprises all messages from the first request sent up to a final response
  - May contain zero or more provisional responses before the last final response
- There are three types of transactions
  - Regular transactions: any transaction with method other than INVITE, ACK or CANCEL
  - INVITE-ACK transactions
  - CANCEL transactions

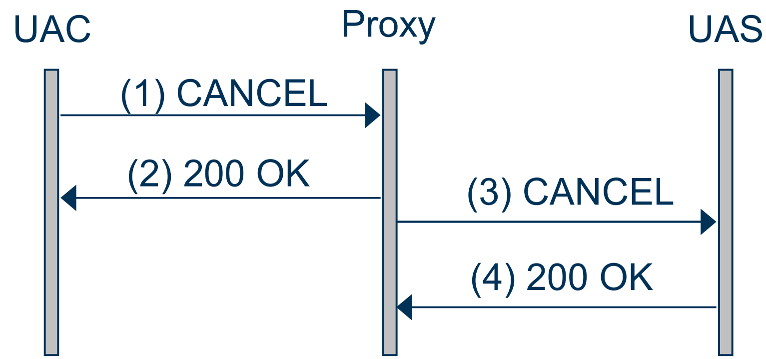


Regular transaction

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

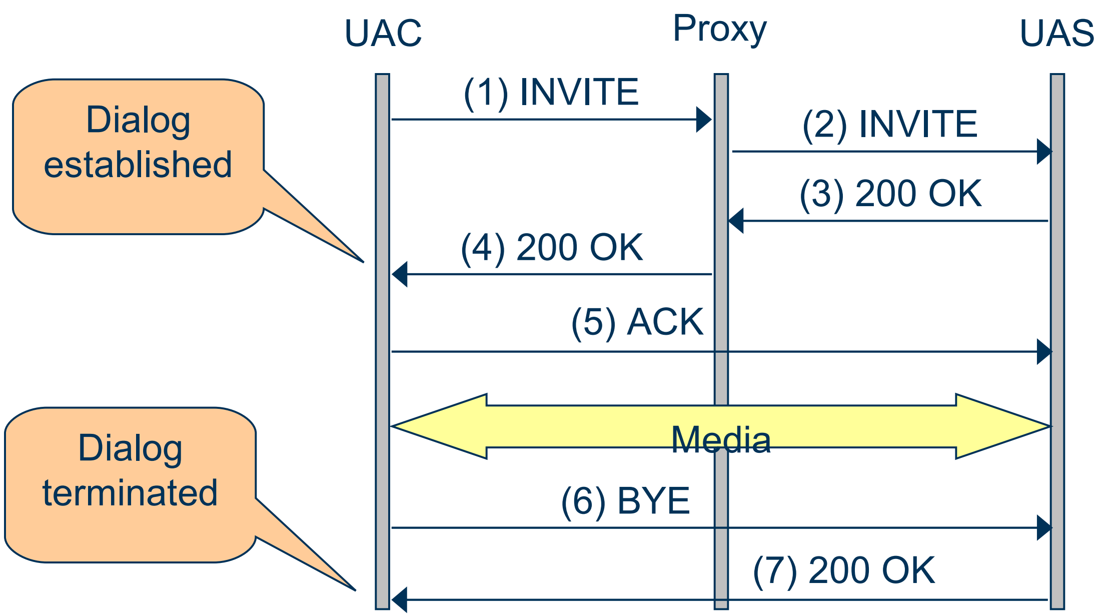




- A CANCEL transaction cancels a previous transaction
  - Connected to a previous transaction
  - Similar to regular transactions with the exception that the final response is generated by the next SIP hop (proxy) instead of the UAS

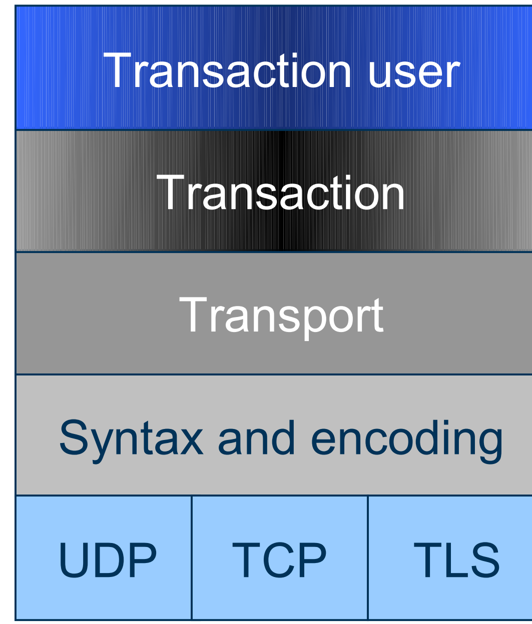
SIP Dialogs

- 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

SIP Version

Reason phrase

Status code

Status Line

```
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
```

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)

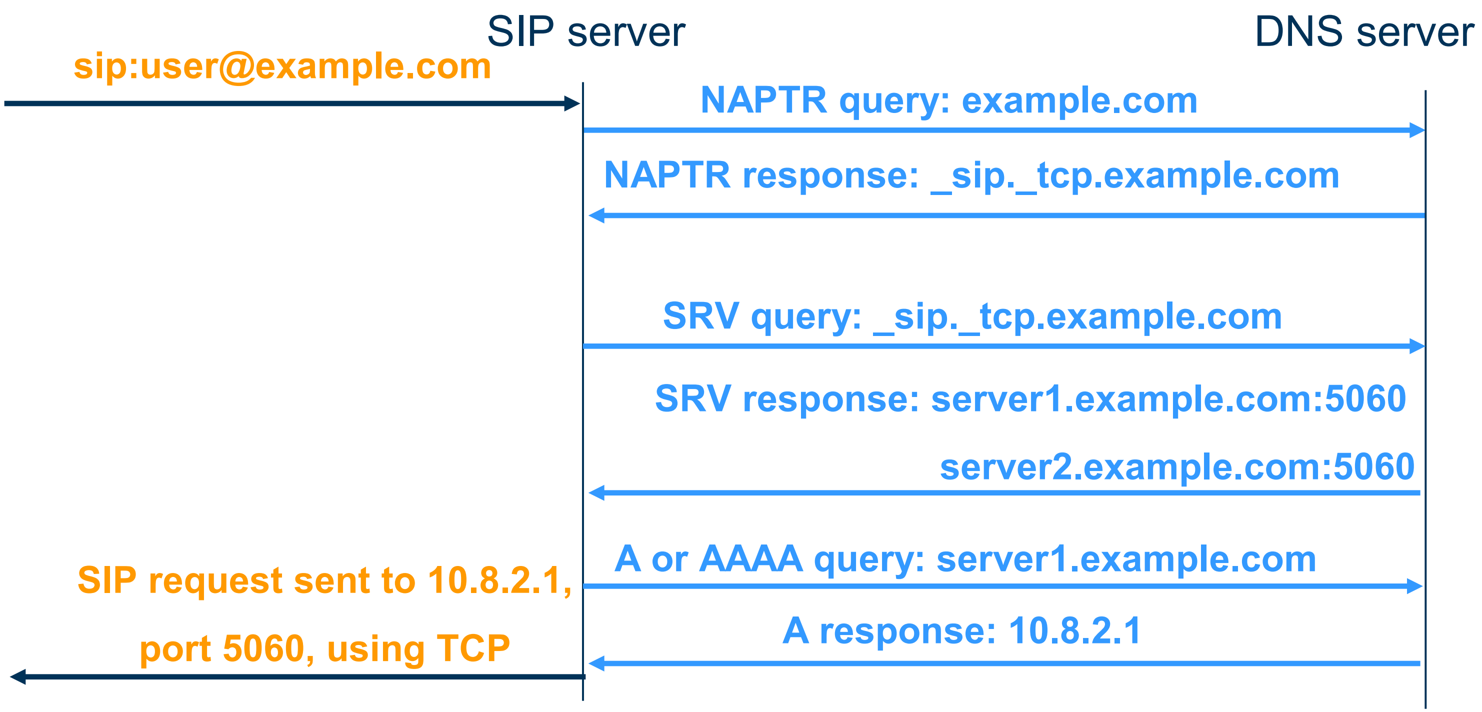
- SIP defines two types of responses
  - Final responses convey the result of the request processing, and are sent reliably
  - Provisional responses provide information on the progress of the request processing, but are not sent reliably in the core protocol (RFC 3261)
- Status codes ranges:
  - 100 – 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 System

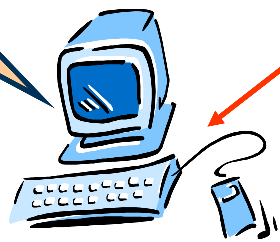
- SIP clients use DNS to route requests and find the next hop to route the request
  - By looking into a NAPTR (Naming Authority Pointer) record in DNS
  - By looking into a SRV (Services) record in DNS
  - By looking into A (IPv4) or AAAA (IPv6) records in DNS
- Example
  - Assumption: no transport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



SIP Registration

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.

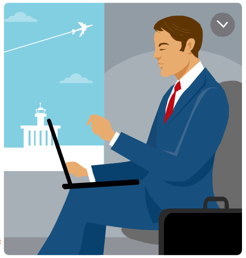
Public user identity *sip:bob.doe@home1.net* is bound to the contact address *sip:bob@play.home1.net*



SIP server
and registrar

1 REGISTER sip:home1.net SIP/2.0
From: <sip:bob.doe@home1.net>
To: <sip:bob.doe@home1.net>
Contact: <sip:bob@play.home1.net>

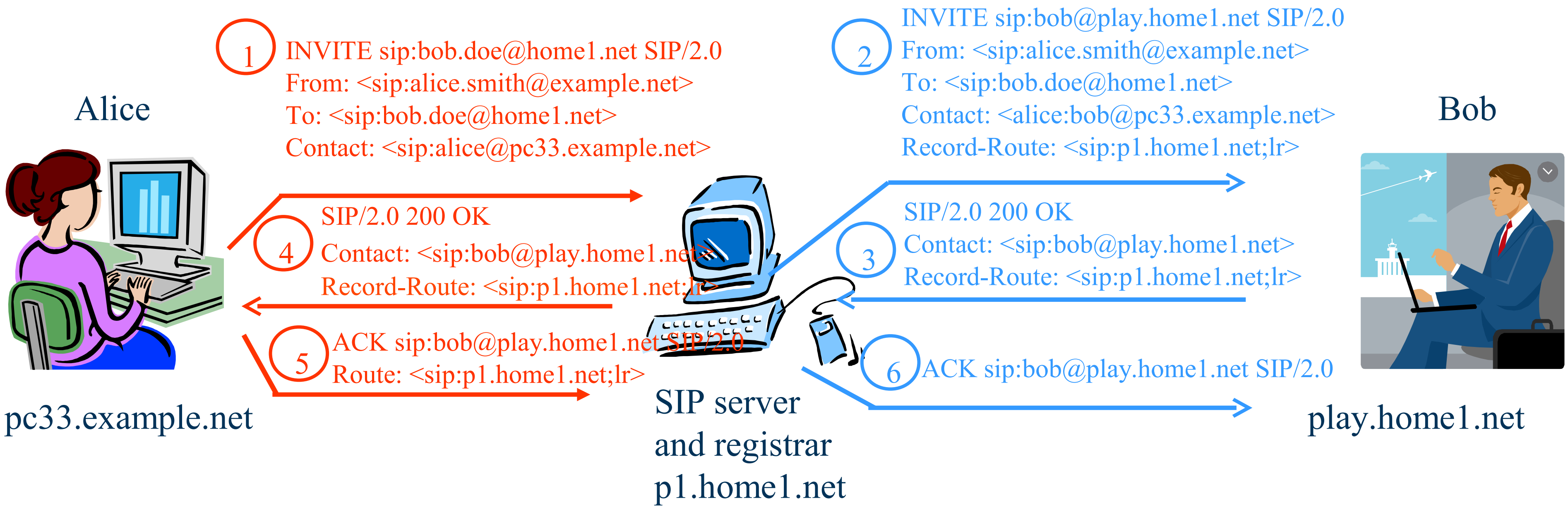
2 SIP/2.0 200 OK
From: <sip:bob.doe@home1.net>
To: <sip:bob.doe@home1.net>
Contact: <sip: bob@play.home1.net>



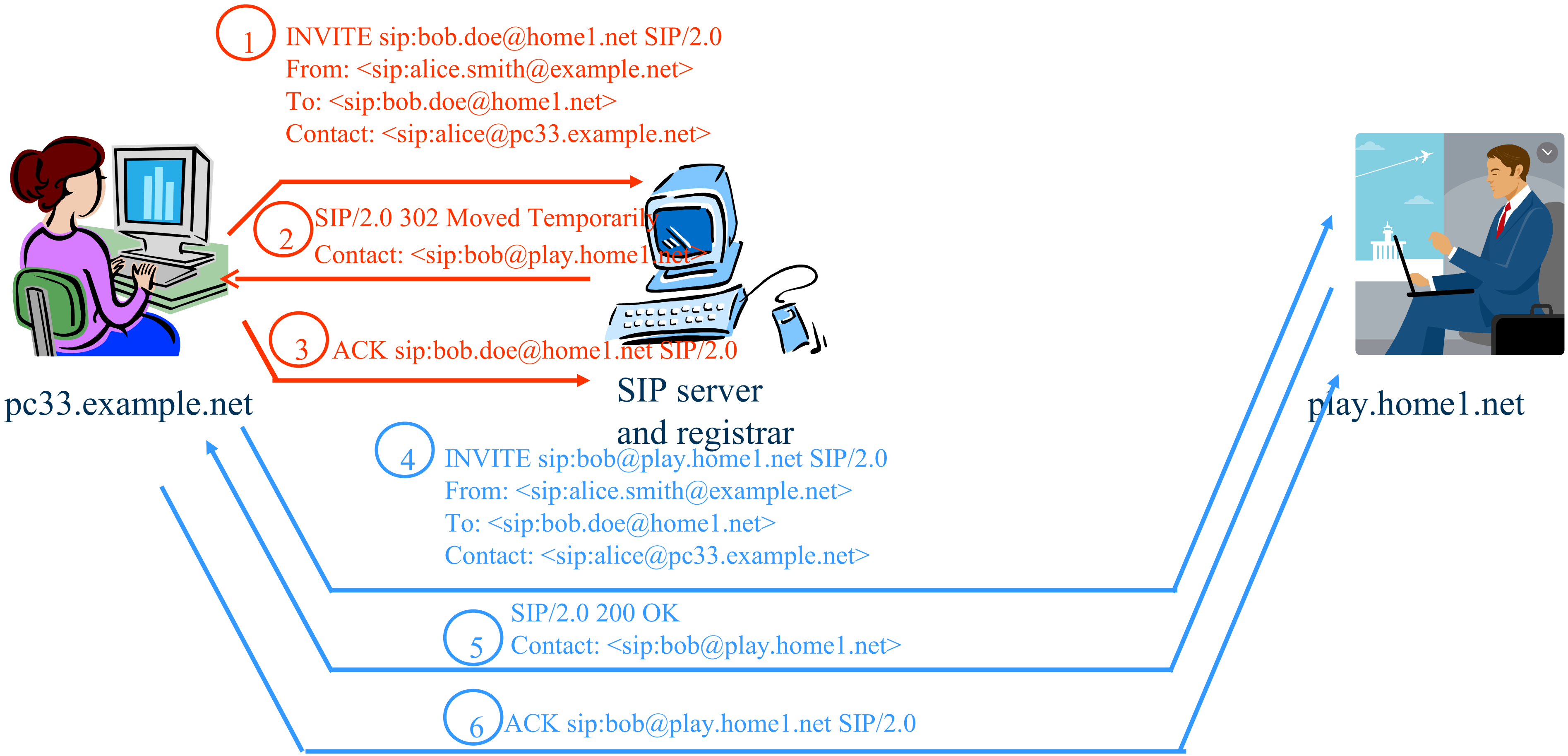
play.home1.net

- 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

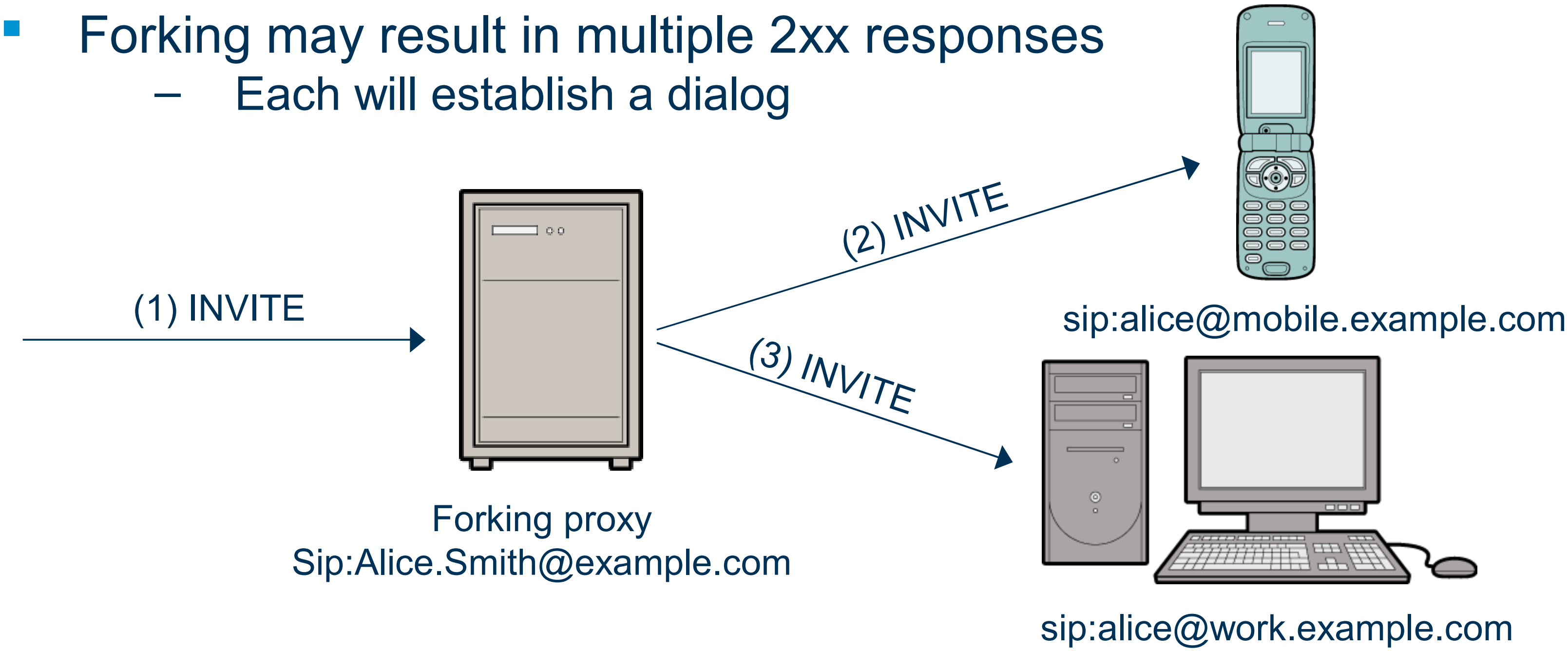


Routing: SIP Server in Redirect Mode



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



ERICSSON

Session Description Protocol (SDP)

- A session description is a description of the session to be established
- 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, and 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=Alice.Smith@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 in the Internet
  - However, 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, which 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) who then 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

Application Areas of SIP

- SIP was originally developed to establish multimedia sessions on the internet (audio, video)
  - Mainly multicast sessions
  - Also unicast sessions
- But has evolved to support other aspects of communications:
  - Voice over IP calls between SIP terminals
  - Voice over IP calls between Gateways (SIP-T)
  - Gaming sessions
  - Instant messaging
  - Presence
  - Multimedia conferences
  - Machine-to-machine communication (e.g., vending machine notifications)

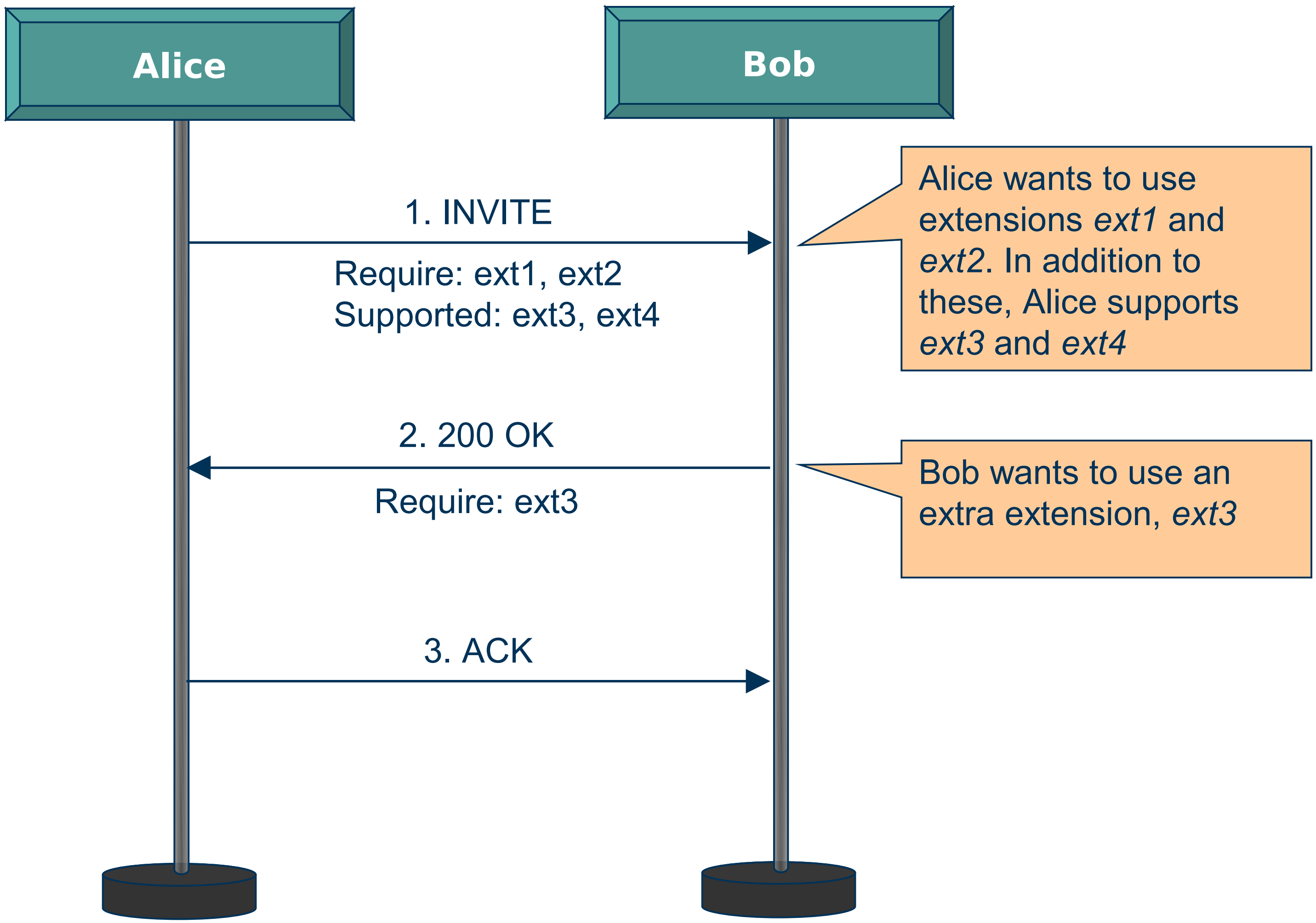
Extending SIP

- Global interoperability possible since the core functionality of SIP as specified in RFC 3261 is present in every implementation
  - A given SIP application can always assume that another SIP application is able to understand the core protocol
- However, many implementations require functionality beyond the core protocol
  - Thus, extensions are required
  - SIP is flexible and easy to extend
- Use of extensions can be negotiated during session establishment
  - Two things are negotiated: the extensions the remote party supports and the extensions that will actually be employed in the session

SIP Extension Negotiation Mechanism

- Three header fields: Require, Supported and Unsupported
- When a dialog is being established, the UAC lists
  - The names of the extensions it wants to use in a Require header field
  - The names of the extensions it supports in a Supported header field
  - The Unsupported header field is used in error responses
- The UAS can also request extra extensions
- Proxy-Require header field can be used to require support of extensions from proxies
- The extensions that a proxy or another UA supports can be queried by using an OPTIONS method
- The names of extensions are referred to as option tags

SIP Extension Negotiation Mechanism



ERICSSON