Introduction to SIP

- SIP introduction
- SIP and other most used VoIP signalling protocols
- TCP/IP protocol family, Basic terms and relations to SIP
- Basics of SIP-operation

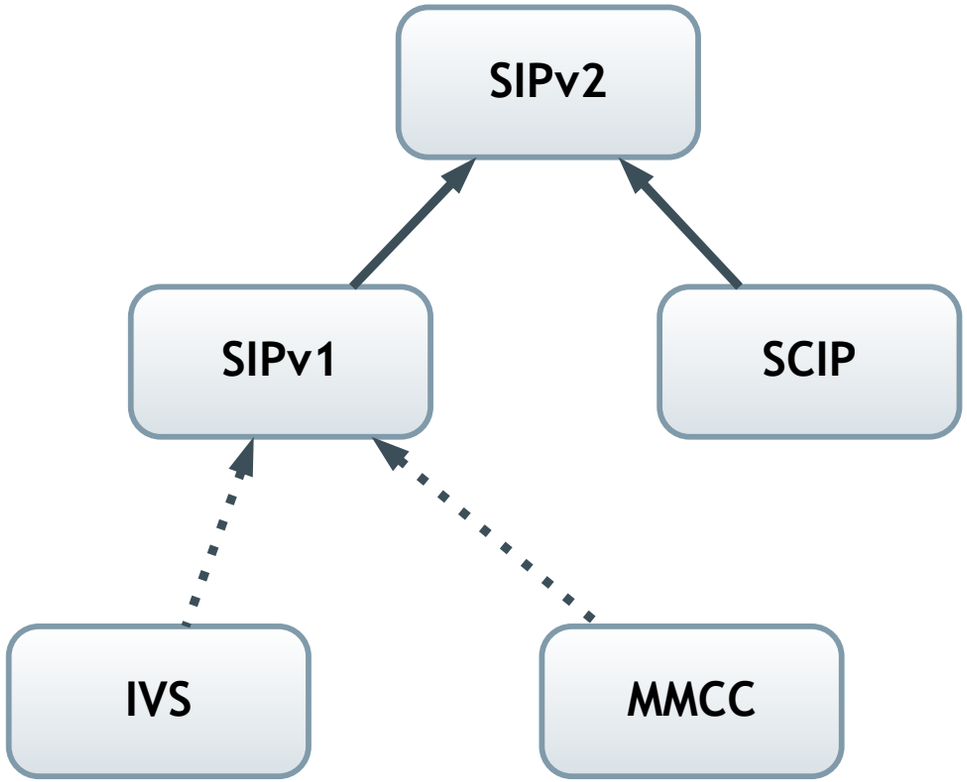
Introduction to SIP

- The Session Initiation Protocol (SIP) is an application-layer control protocol that can establish, modify and terminate multimedia sessions or calls.
- SIP sessions involve one or more participants and can use unicast or multicast communication.

SIP

- SIP = Session Initiation Protocol
- Signalling protocol for initiating, modifying and terminating sessions in IP networks
- Used in VoIP
- The base of IMS (IP Multimedia Subsystem) architecture

Evolution towards SIP



SIPv2 - Session Initiation Protocol
SIPv1 - Session Invitation Protocol
SCIP - Simple Conference Invitation Protocol
MMCC - Multimedia Conference Control
IVS - INRIA Videoconferencing System

History of SIP

- 1996 – Internet Draft, Version 1.0
  - Originally developed by the IETF MMUSIC Working Group
- 1998 – Internet Draft, Version 2.0
- 1999 (March) – Proposal Standard
- 1999 (April) – Published as RFC 2543
- 1999 (September) – SIP working group established
- 2002 – RFC 3261
- Currently exist a lot of standard RFC extensions to RFC 3261

Design Principles of SIP

- Based on HTTP and SMTP
- Transport protocol neutrality (UDP, TCP, SCTP)
- Request routing – direct or proxy-routed
- Separation of signalling and media description
- Extensibility
- Personal mobility (different terminal, same identifier)

User plane vs. Control plane

- User Plane
  - Provides user's information transfer along with associated controls (e.g. flow control, error control)
  - Protocol: RTP, RTCP
- Control Plane
  - Performs connection control functions (e.g. connection setup, maintaining terminations)
  - Protocol: SIP, SDP,…

VoIP Signalling protocols

- Open standards/specifications (that can anyone study and use in his implementations/products)
  - SIP – by IETF, that's what we are going to talk about
  - H.323 – by ITU-T, until recently the biggest rival of SIP
  - H.248 / MEGACO – MEdia GAteway COntrol protocol
  - IAX – Inter Asterisk Exchange protocol, not a standard indeed, proprietary but open-source

VoIP Signalling protocols

- Proprietary (developed / licensed by single vendor)
  - Skype
    - YES it IS a VoIP signalling protocol too!
    - No one (except the author) really knows how Skype operates in real.
    - Seems to use Peer-to-Peer distributed model, with some elements of centralized control
    - True proprietary one
      - Totally closed specification
      - including heavy anti-reverse-engineering protection
      - Totally closed for third party developers, may include strange code!
      - Widely used, carries significant part of nowadays VoIP traffic over public internet.

VoIP Signalling protocols

- Proprietary...continue
  - Skinny Client Control Prot.
    - Cisco's proprietary VoIP protocol (No to be confused with SS7-SCCP)
    - Connects Cisco VoIP phones to the Cisco Call Manager server.
    - The Cisco Call Manager is an H.323 proxy that communicates with Skinny clients (the Cisco phones)
    - Much less overhead than with H.323.
    - SCCP is a "lite" client that reduces the processing load on the hardware.
    - Not as closed as Skype, some third party devices also support Skinny

VoIP Signalling protocols

- Proprietary...continue
  - Others
    - Nearly every major VoIP-telephony vendor (Alcatel, AT&T, Siemens, ...) uses some kind of its own VoIP signalling protocol.
    - Generally used to connect system-phones with PBX using IP network.
    - Some features adopted from open standards could be seen.

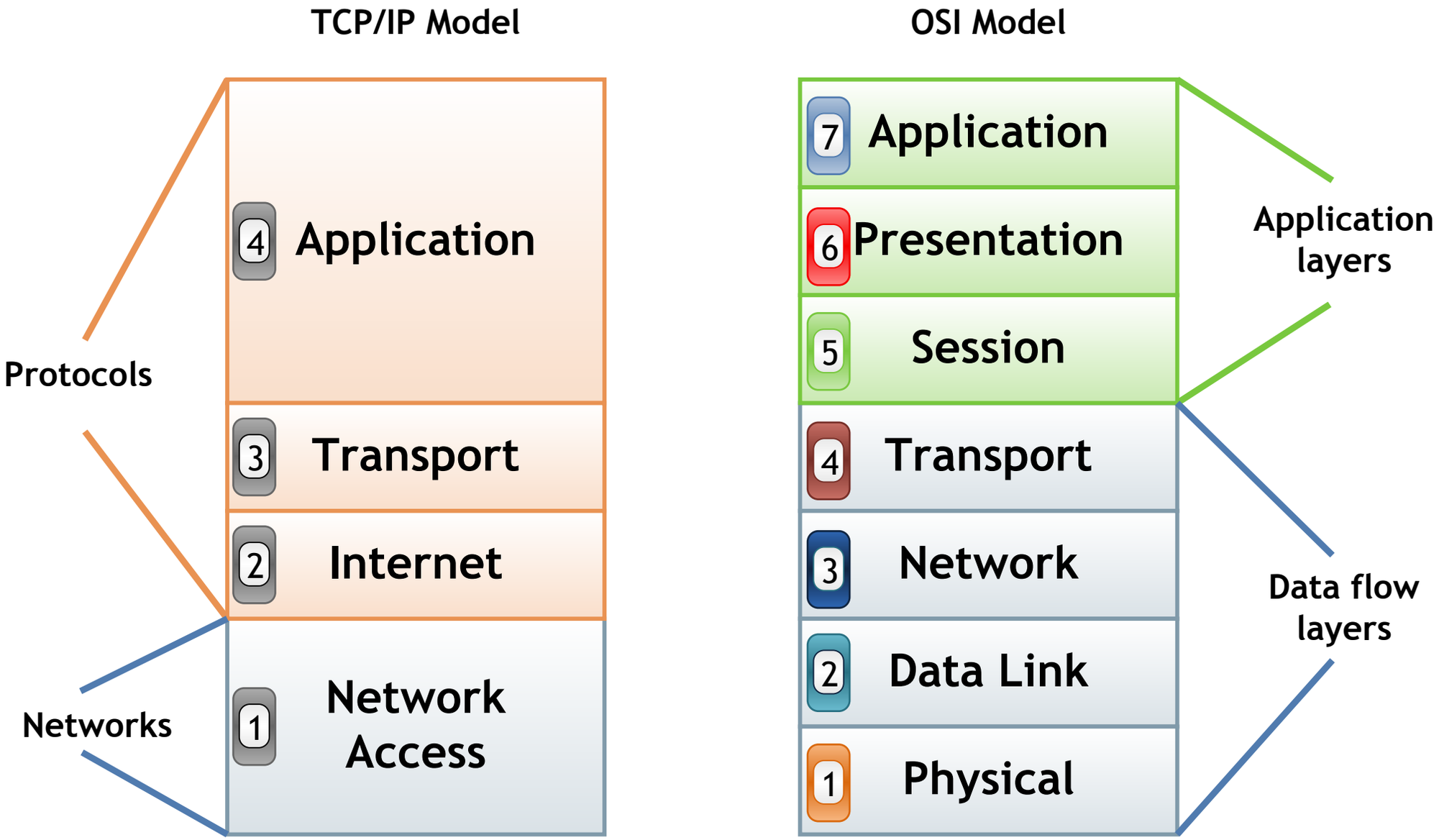
SIP and TCP/IP protocol suite

- SIP is a VoIP signalling protocol
- VoIP transports both Signalling and Payload(user data) using an IP based packet switched network.
- SIP transports its signalling messages using TCP/IP protocol suite transport protocols (UDP, TCP, SCTP)

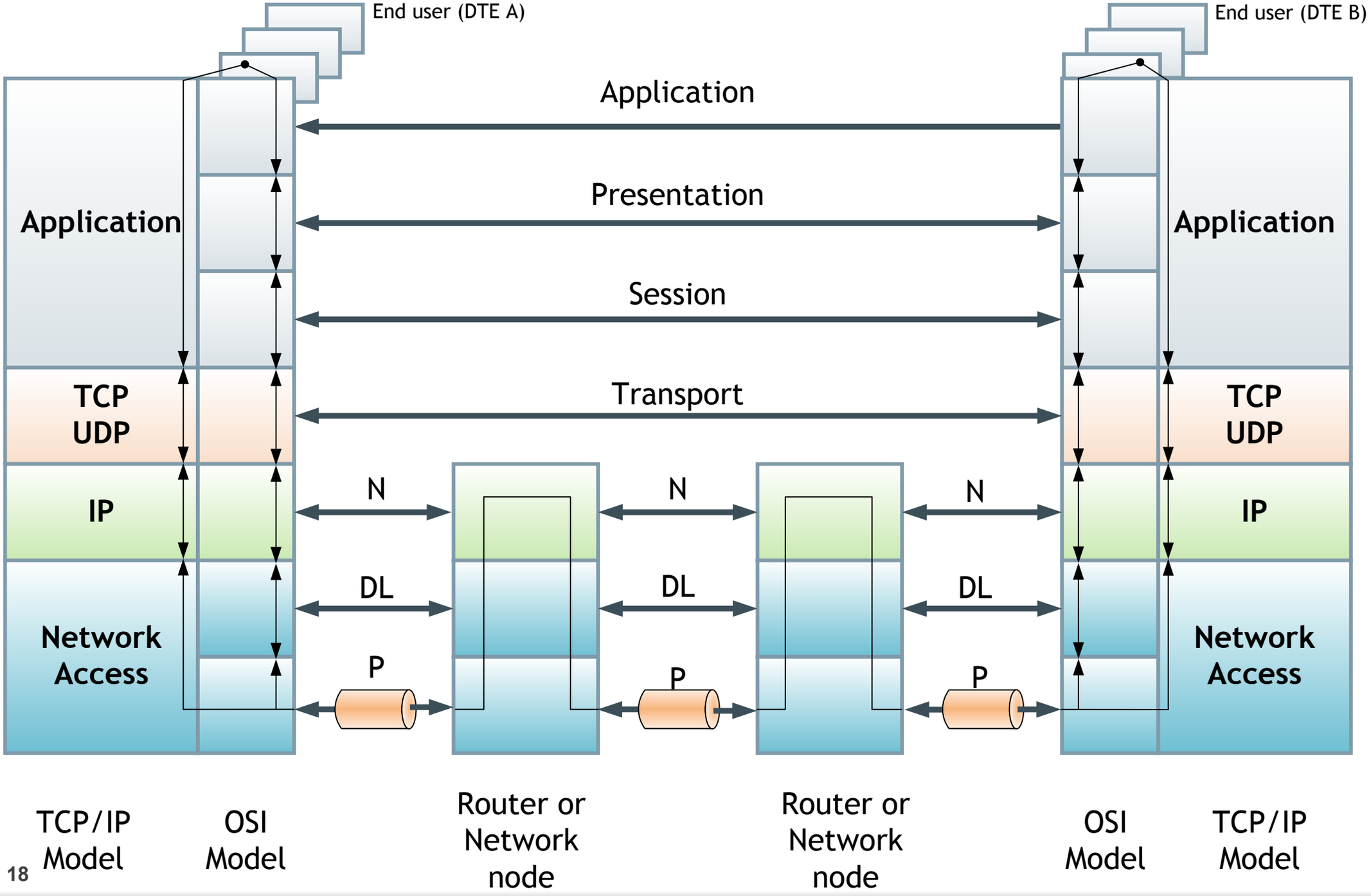
SIP and TCP/IP protocol suite

- From ISO-OSI model perspective SIP is an application layer protocol.
- From IP network perspective SIP entities are processes/ applications running on various IP hosts
- SIP utilizes IP transport protocols (UDP,TCP,SCTP) port multiplexing.

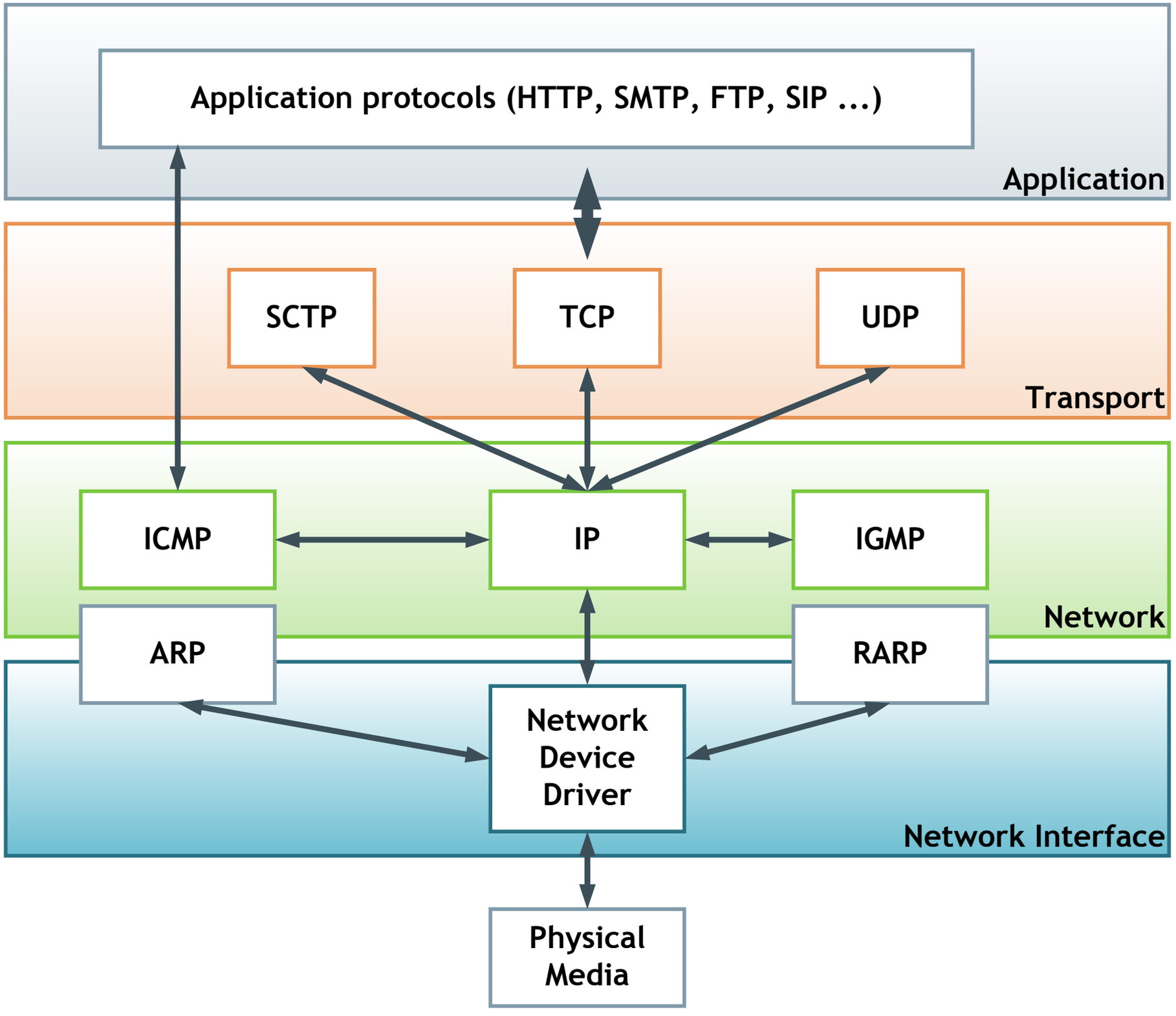
OSI/TCP model comparison



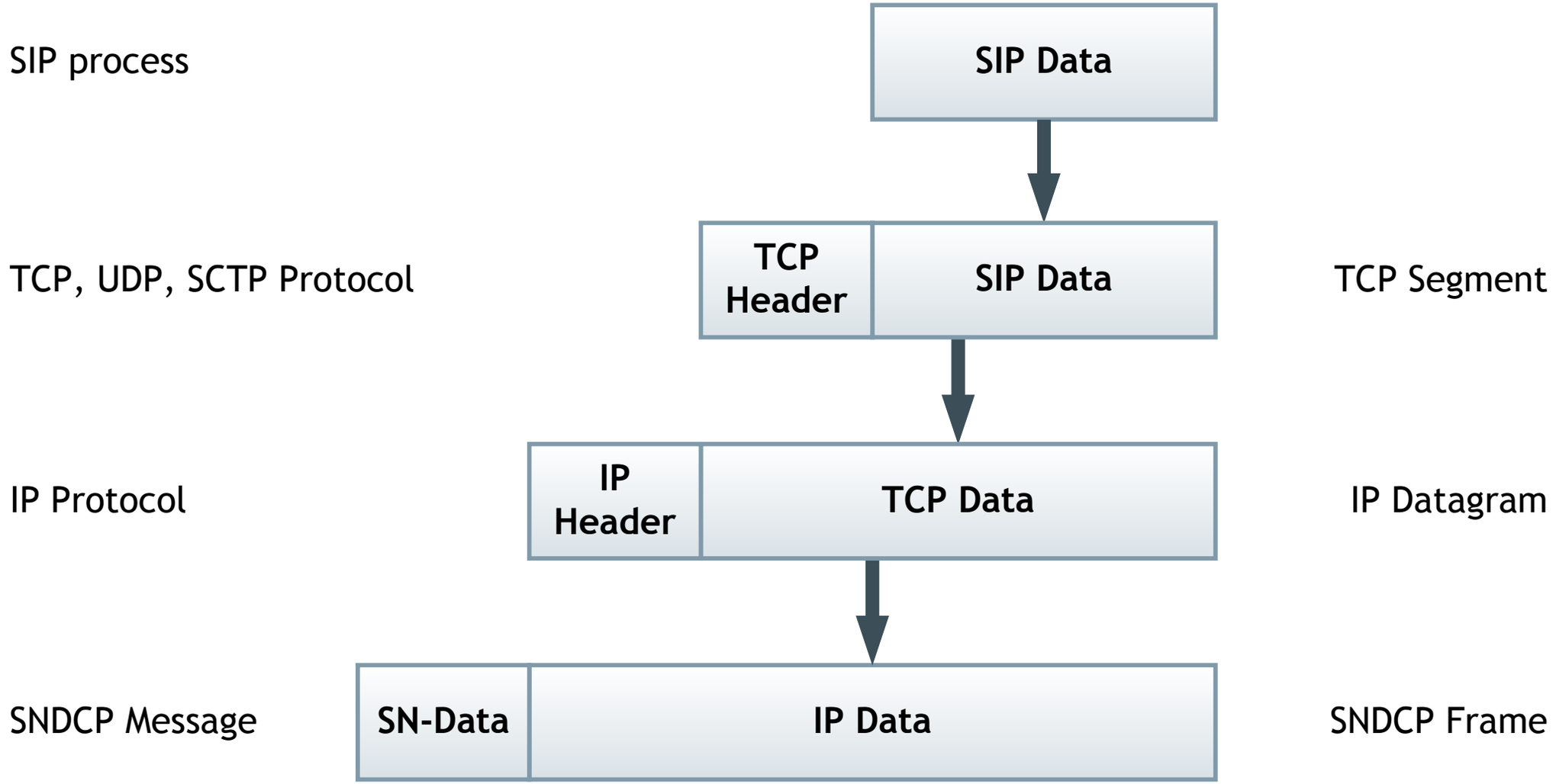
OSI 7-Layer vs. TCP/IP 4-layer-Model



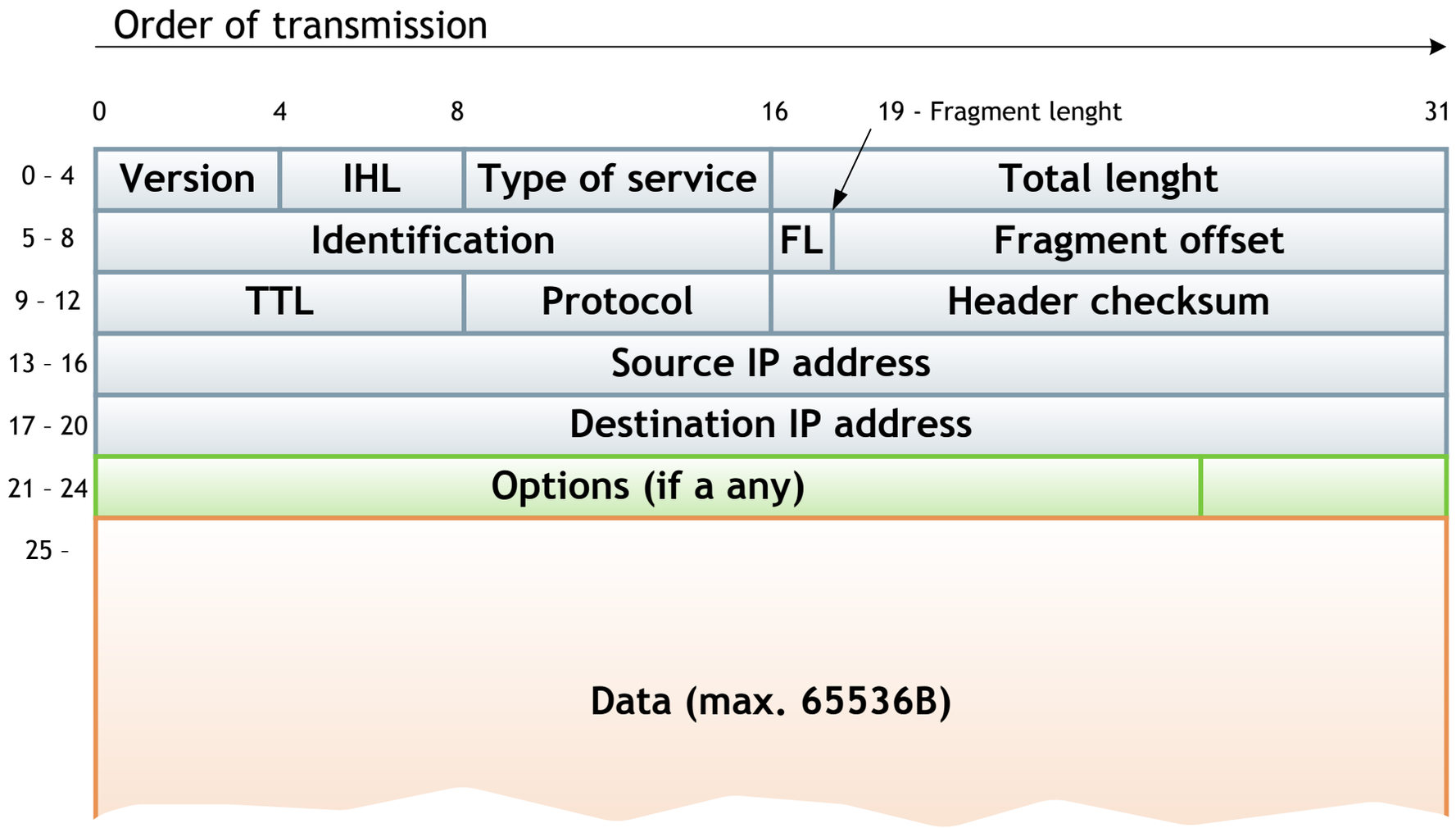
TCP/IP Protocol Architecture



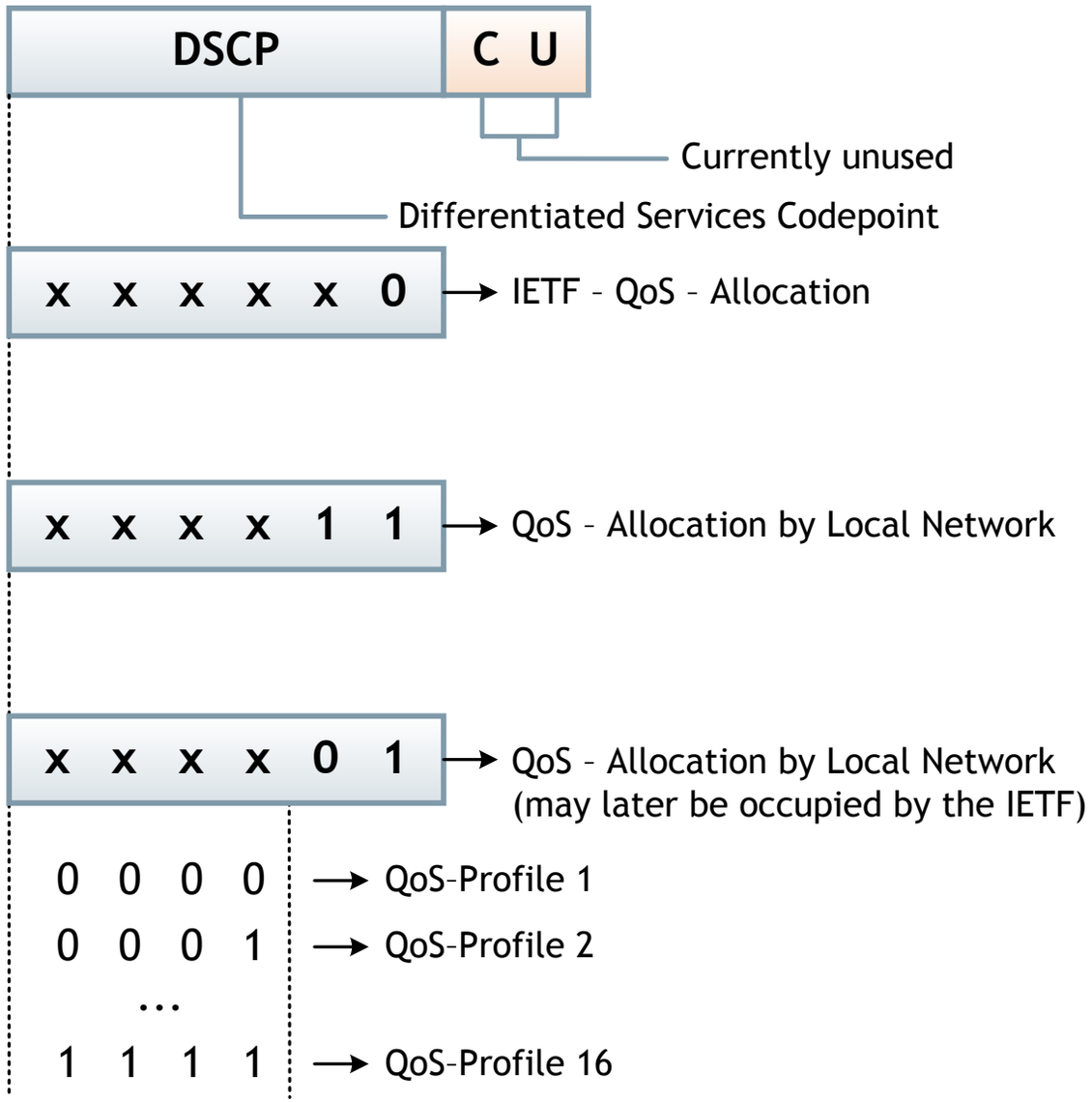
Data encapsulation



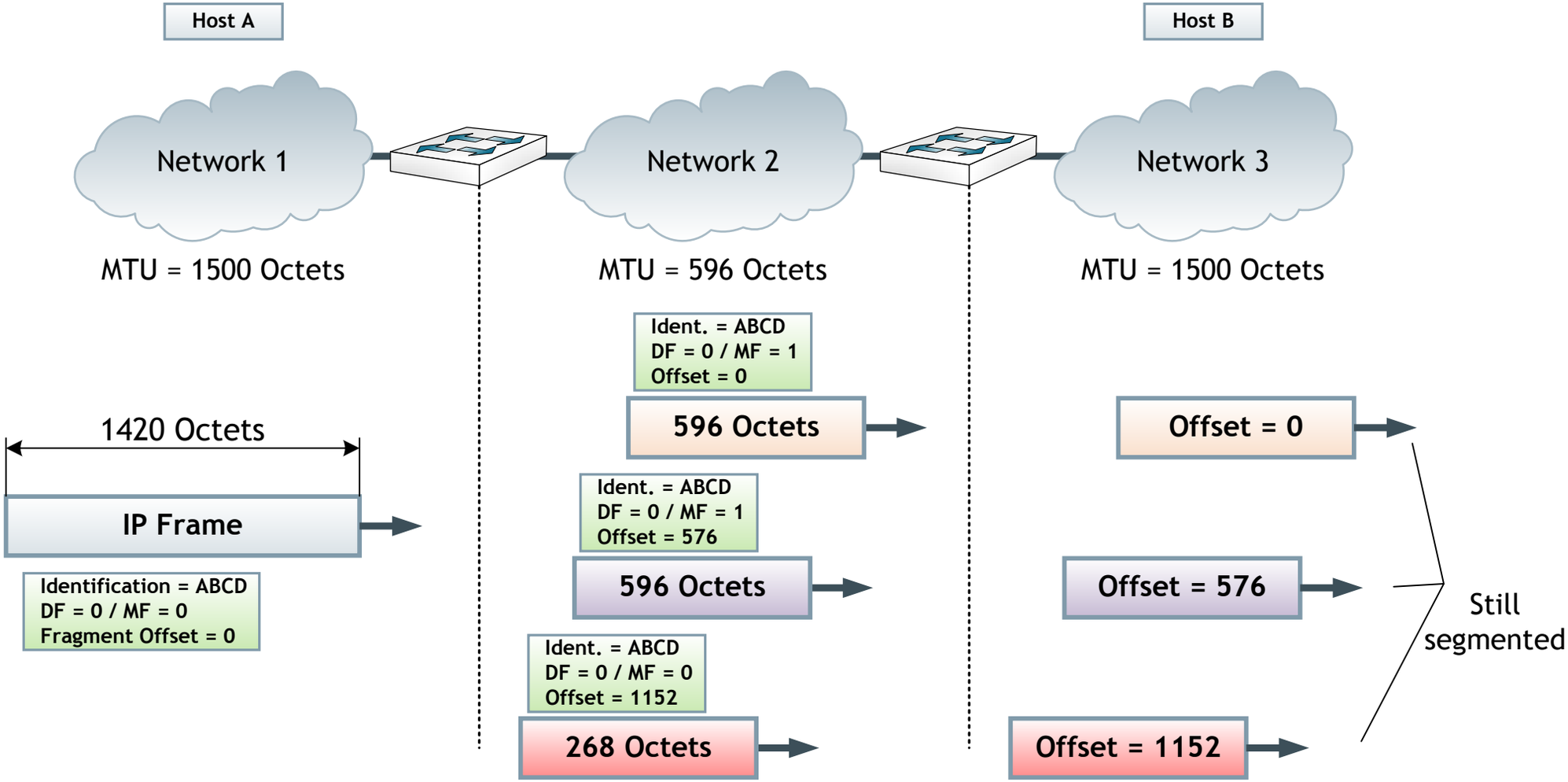
IP packet


Order of transmission
0
4
8
16
19 - Fragment lenght
31
0 - 4
Version
IHL
Type of service
Total lenght
5 - 8
Identification
FL
Fragment offset
9 - 12
TTL
Protocol
Header checksum
13 - 16
Source IP address
17 - 20
Destination IP address
21 - 24
Options (if a any)
25 -
Data (max. 65536B)

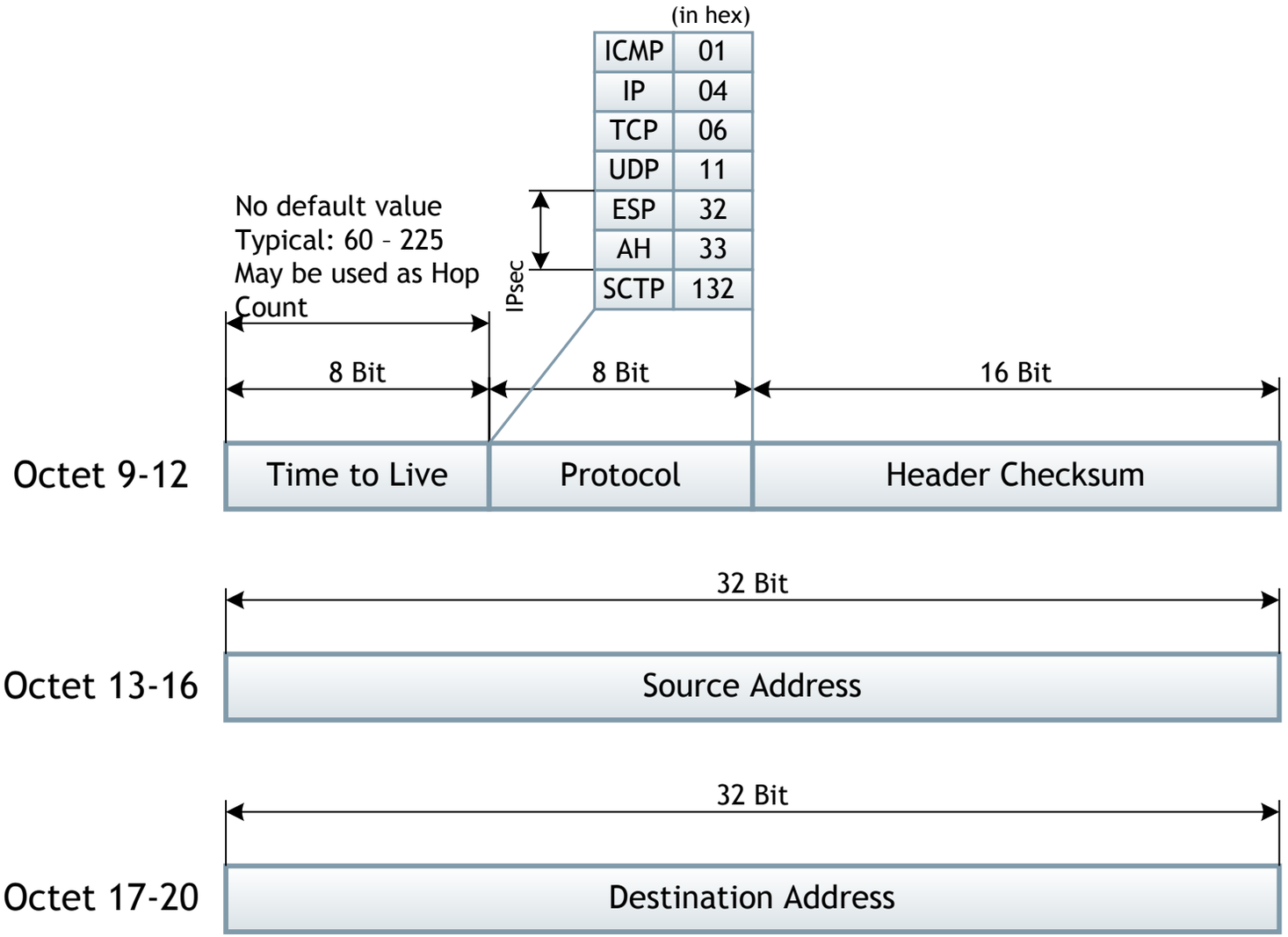
TOS-Field / Differentiated services



Fragmentation control in IP



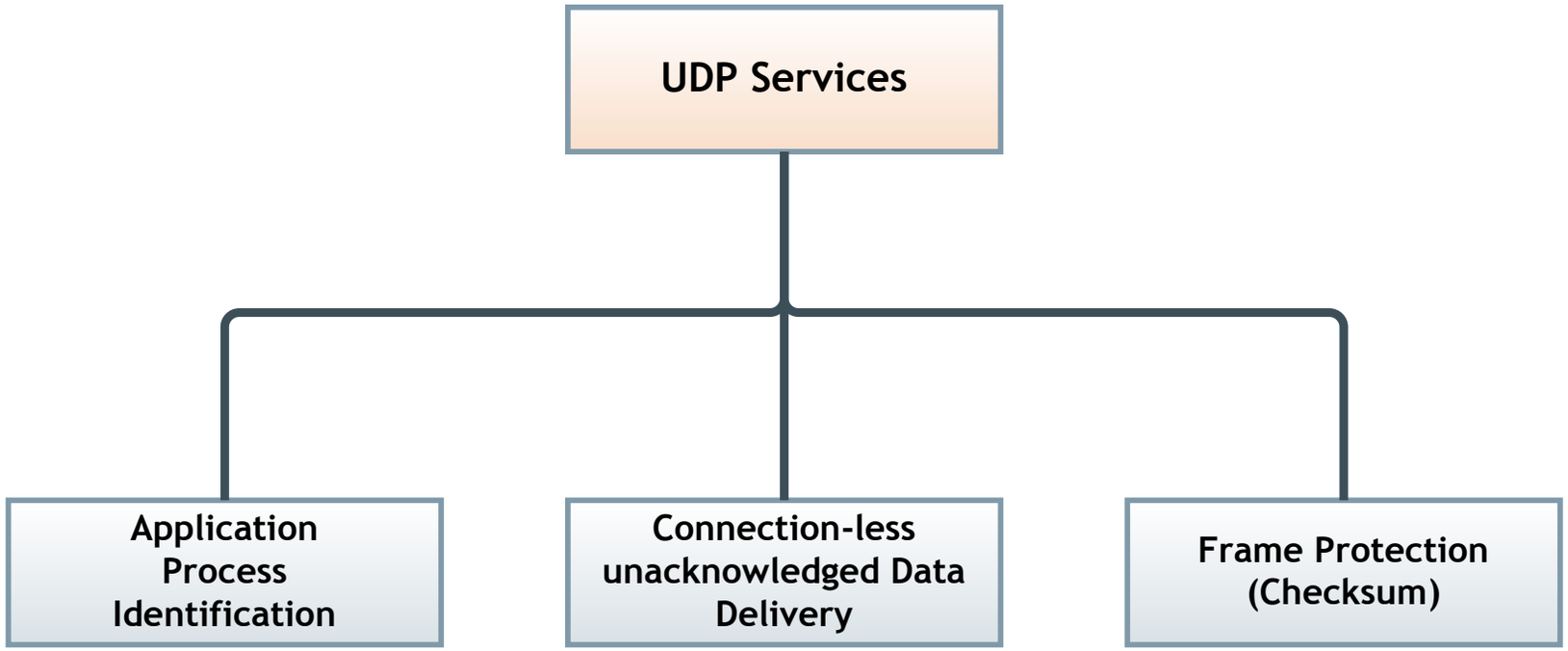
The IP Header / Octet 9 - 20



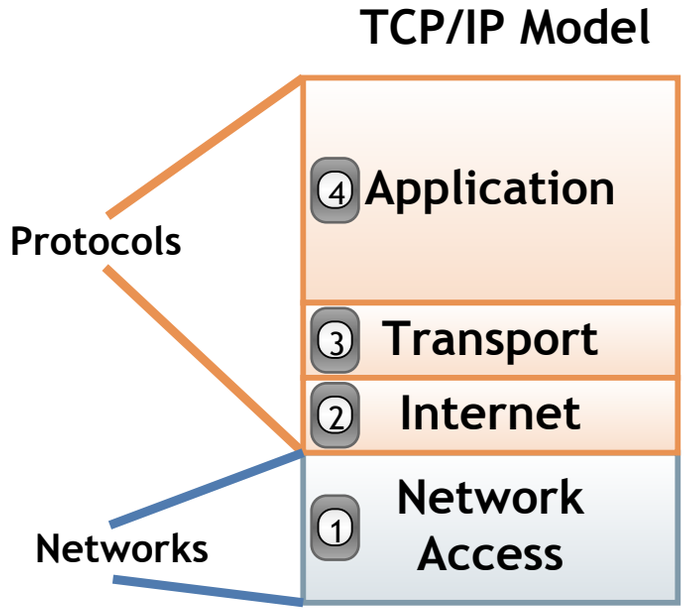
UDP

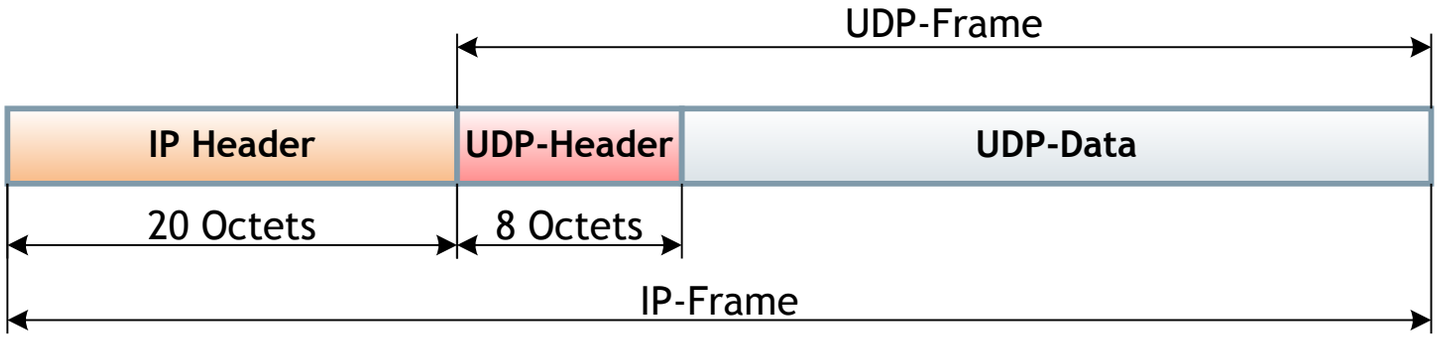
- User Datagram Protocol - provides datagram mode of packet-switched computer communication in IP networks.
- Internet Protocol (IP) is used as the underlying protocol (Protocol number 17 in IP header).
- Provides a procedure for application programs to send messages with a minimum of protocol mechanism.
- Delivery and duplicate protection are NOT guaranteed.
- UDP uses the same port based communication model as TCP/ SCTP – application sockets.
- Numbers of UDP ports do not have to correspond with numbers of TCP ports.

Services of UDP

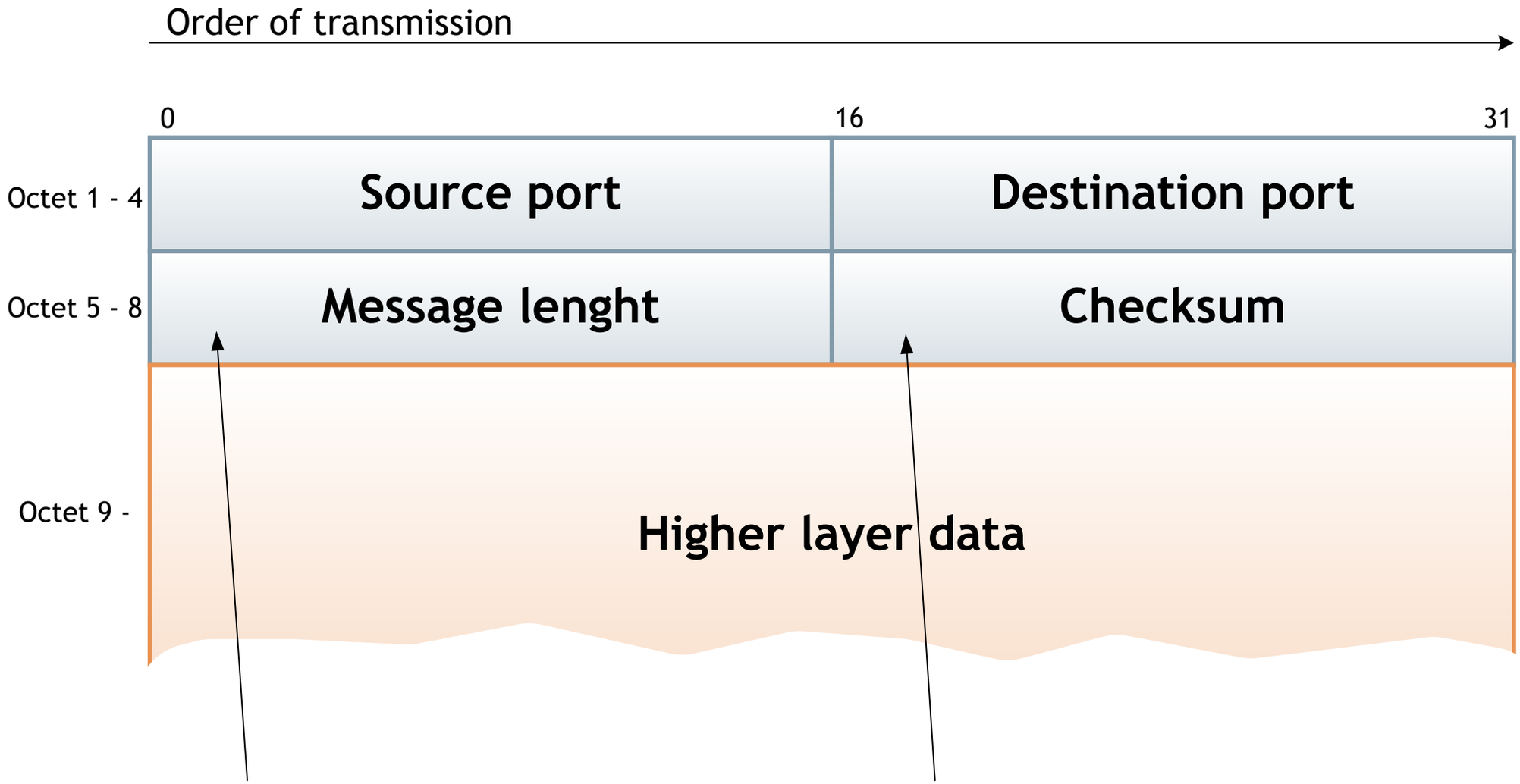


Details of the User Datagram Protocol (UDP)





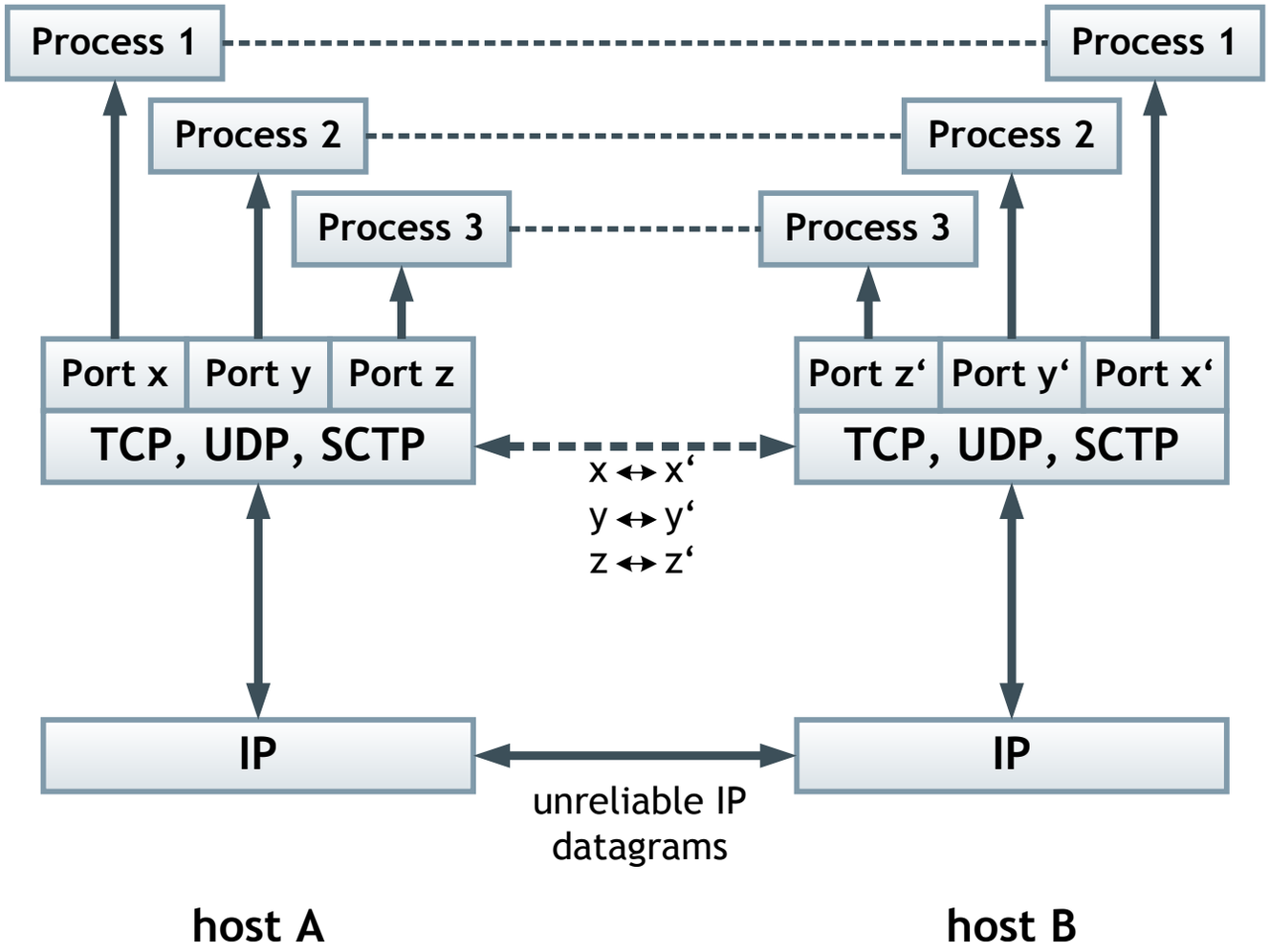
UDP datagram



field includes the 8-byte header and the data, measured on octets

field is optional and stored as zero if not computed (a computed zero is stored as all ones)

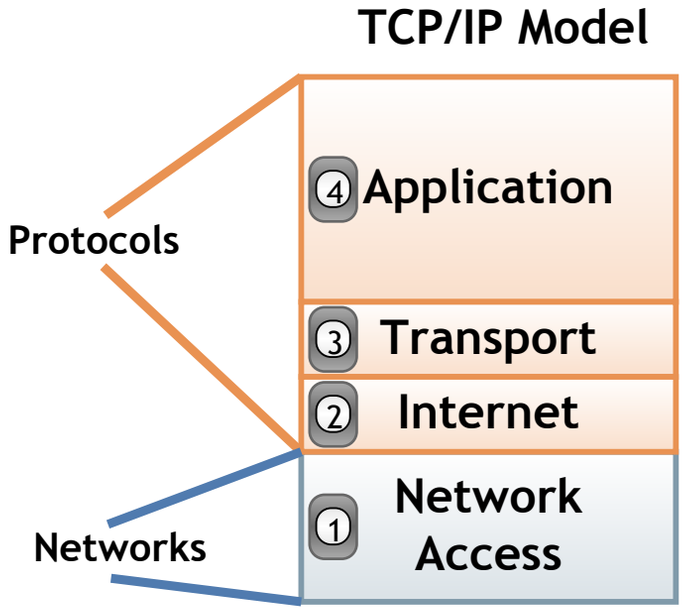
Communication model

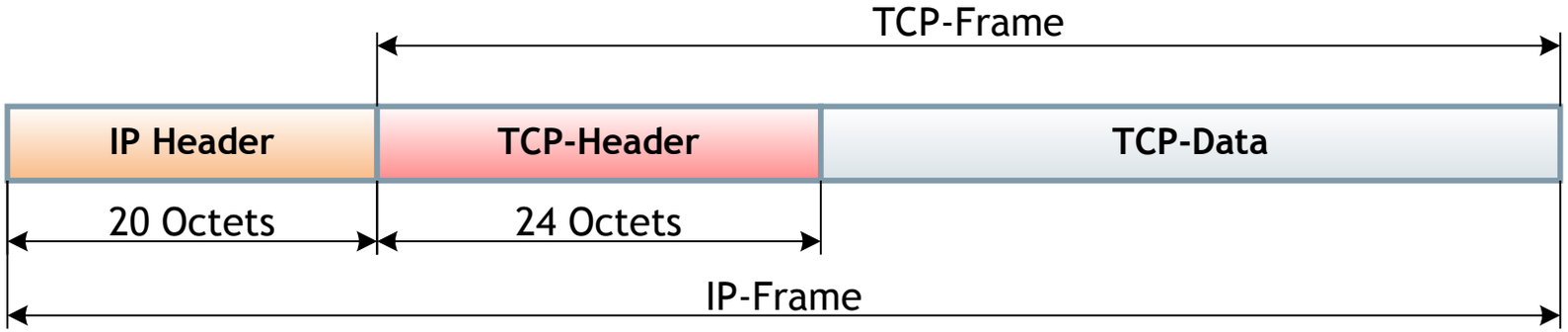


Transmission Control Protocol (RFC 793)

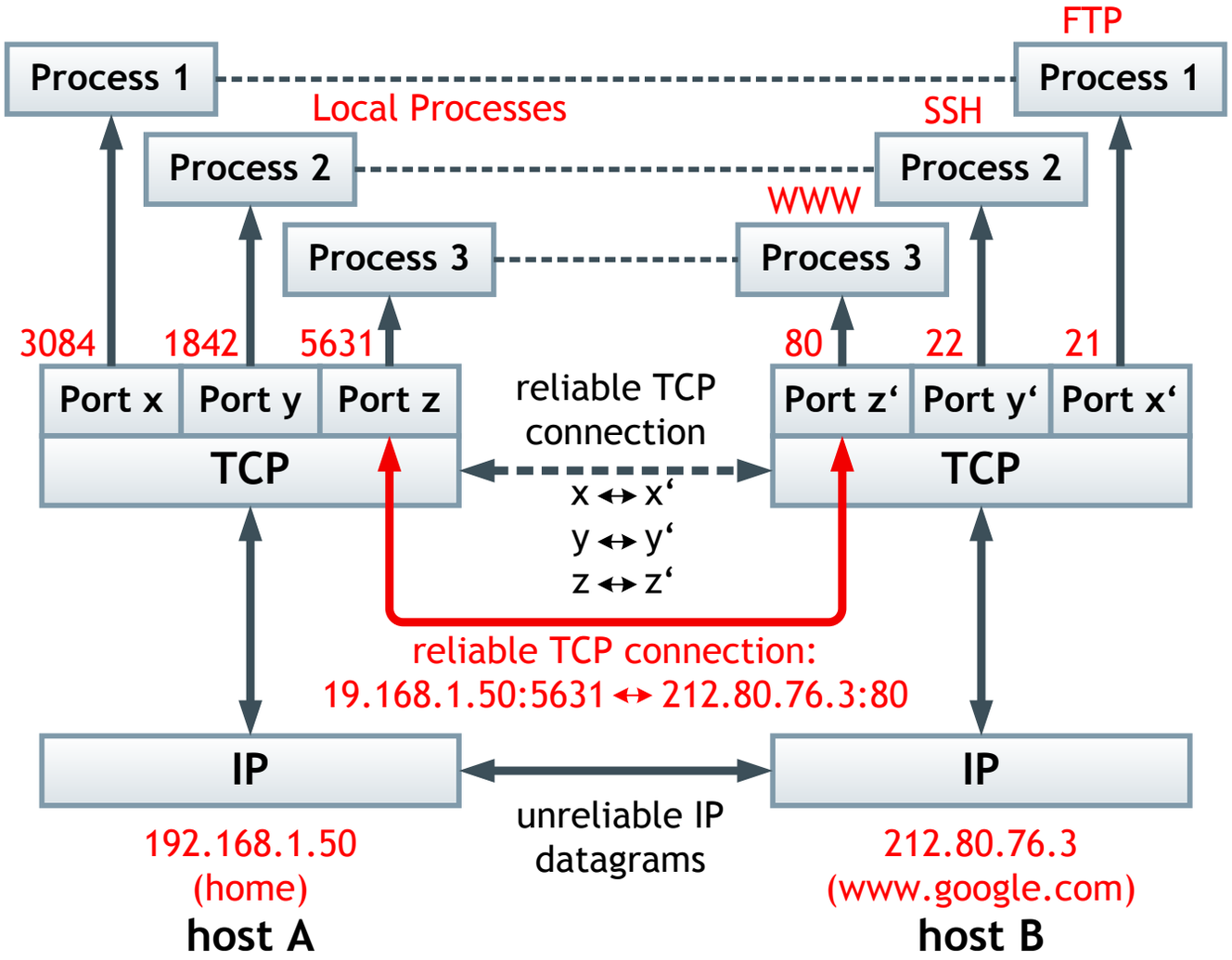
- Because IP provides only unreliable means of transporting data, the main purpose of TCP is to provide reliable logical connection between two communicating processes. TCP provides:
  - Basic Data Transfer
  - Reliability
  - Flow Control
  - Multiplexing
  - Full duplex logical connection

Details of the Transmission Control Protocol (TCP)

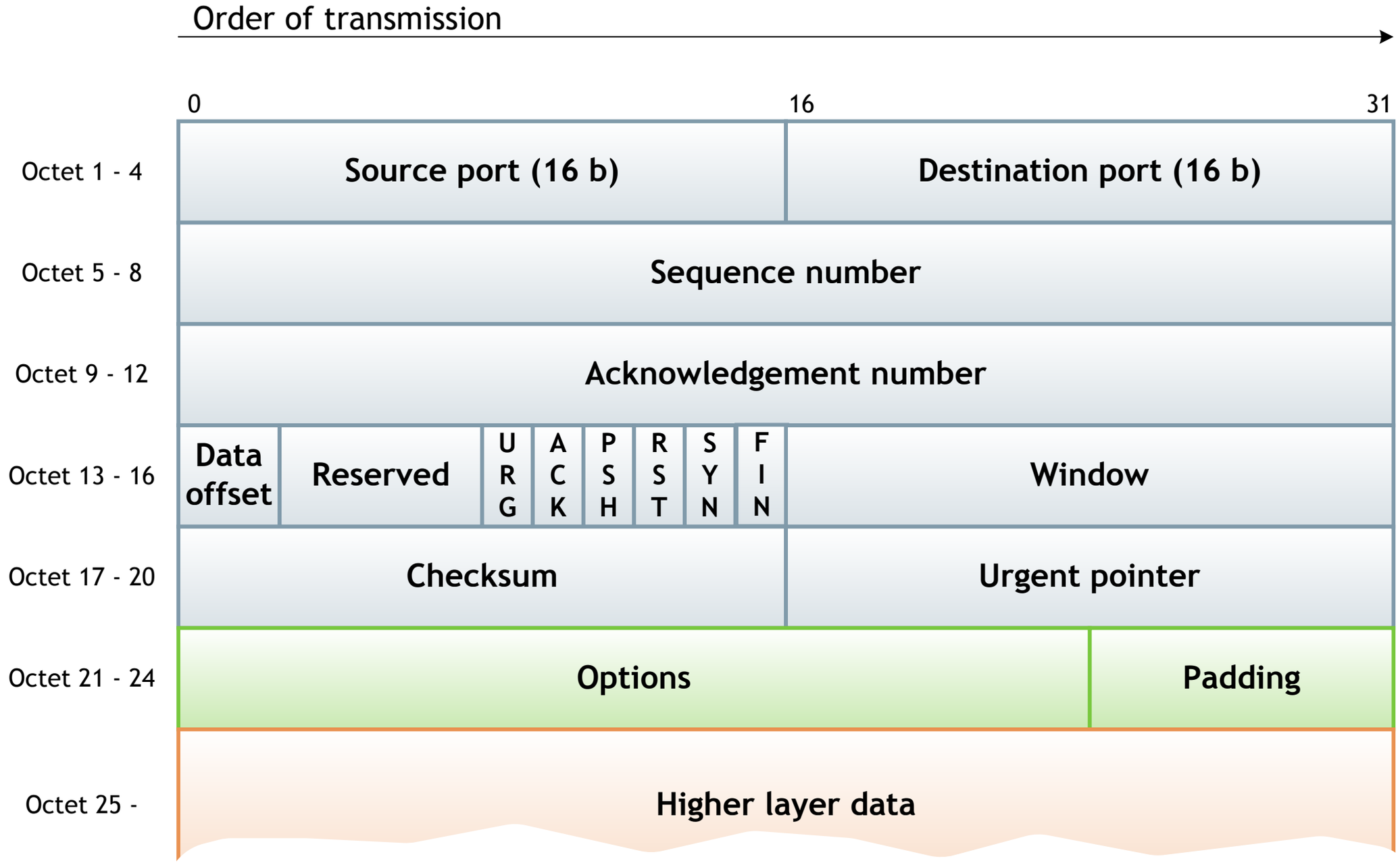




TCP Multiplexing

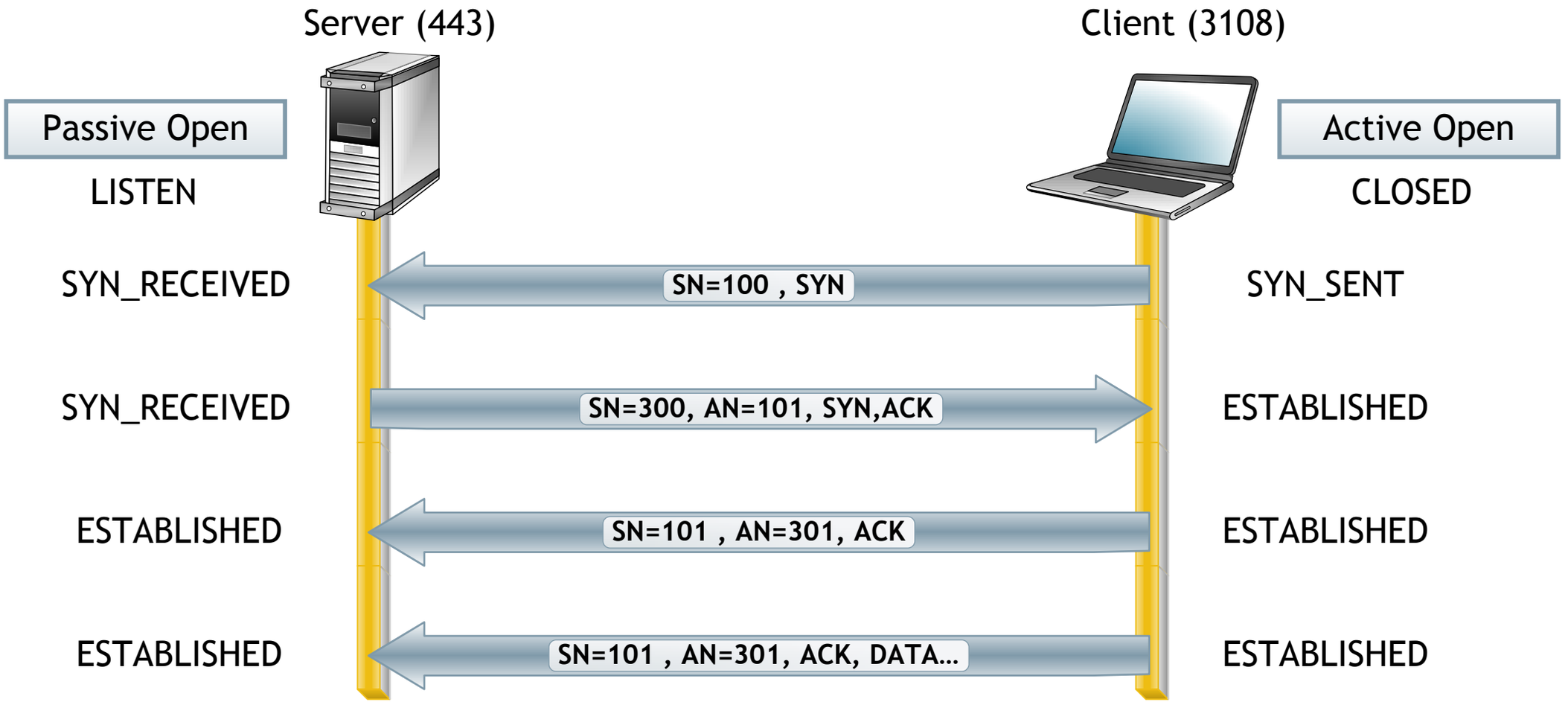


TCP datagram



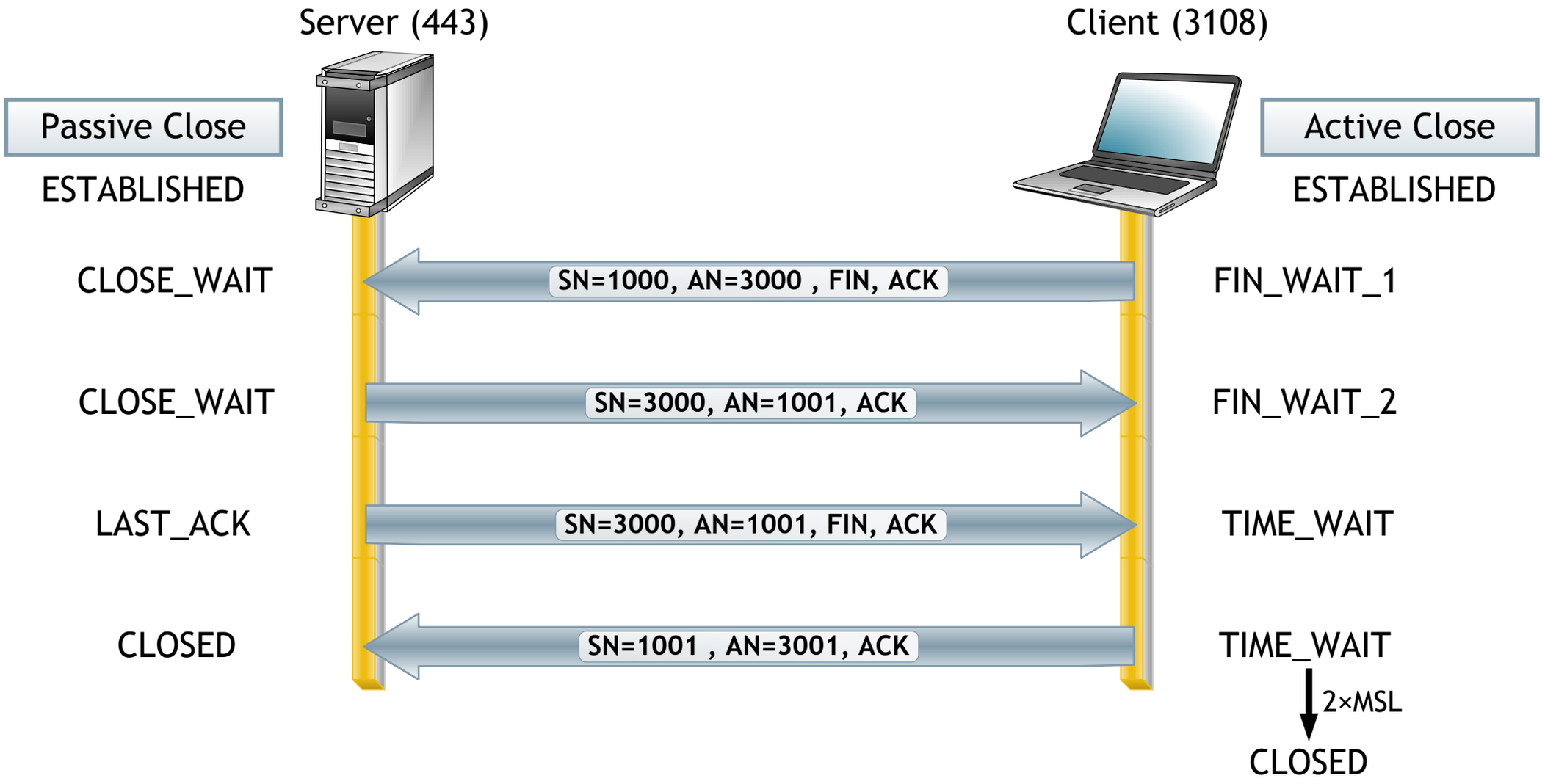
TCP connection establishment

- So called TCP three-way handshake i.e. connection is established after three TCP segments are transferred.



TCP connection release

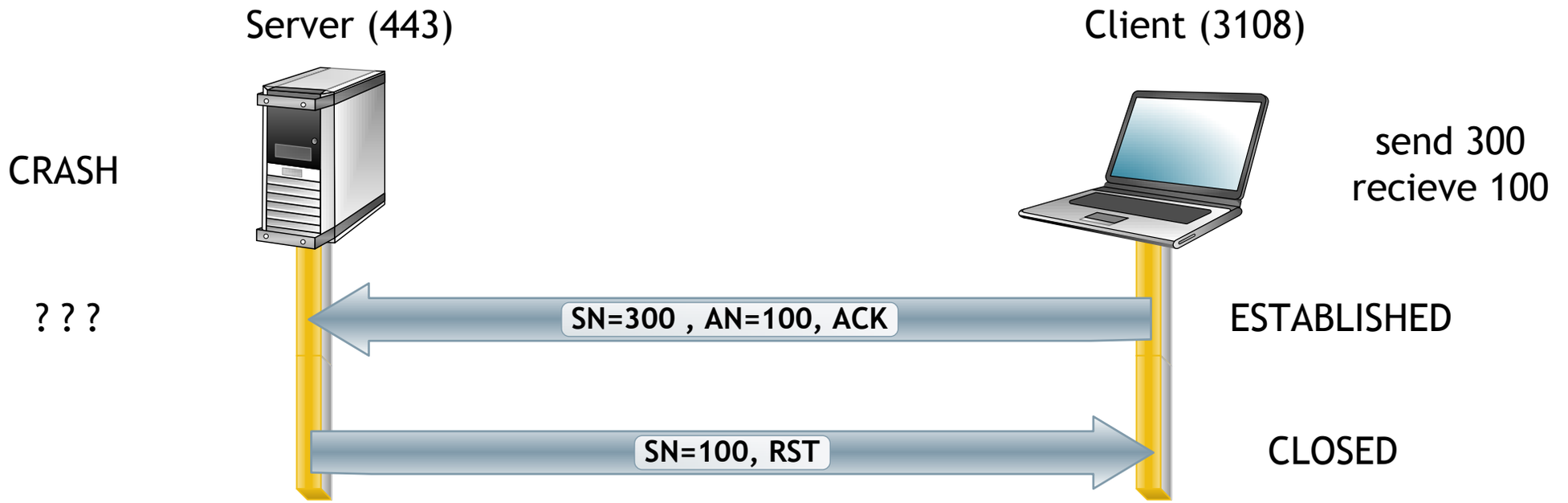
- It is needed to send out four TCP segments befor we have to terminate two independent connections (sockets).



TCP connection reset

This procedure is used when refusing TCP connection – often in abnormal situations. For instance:

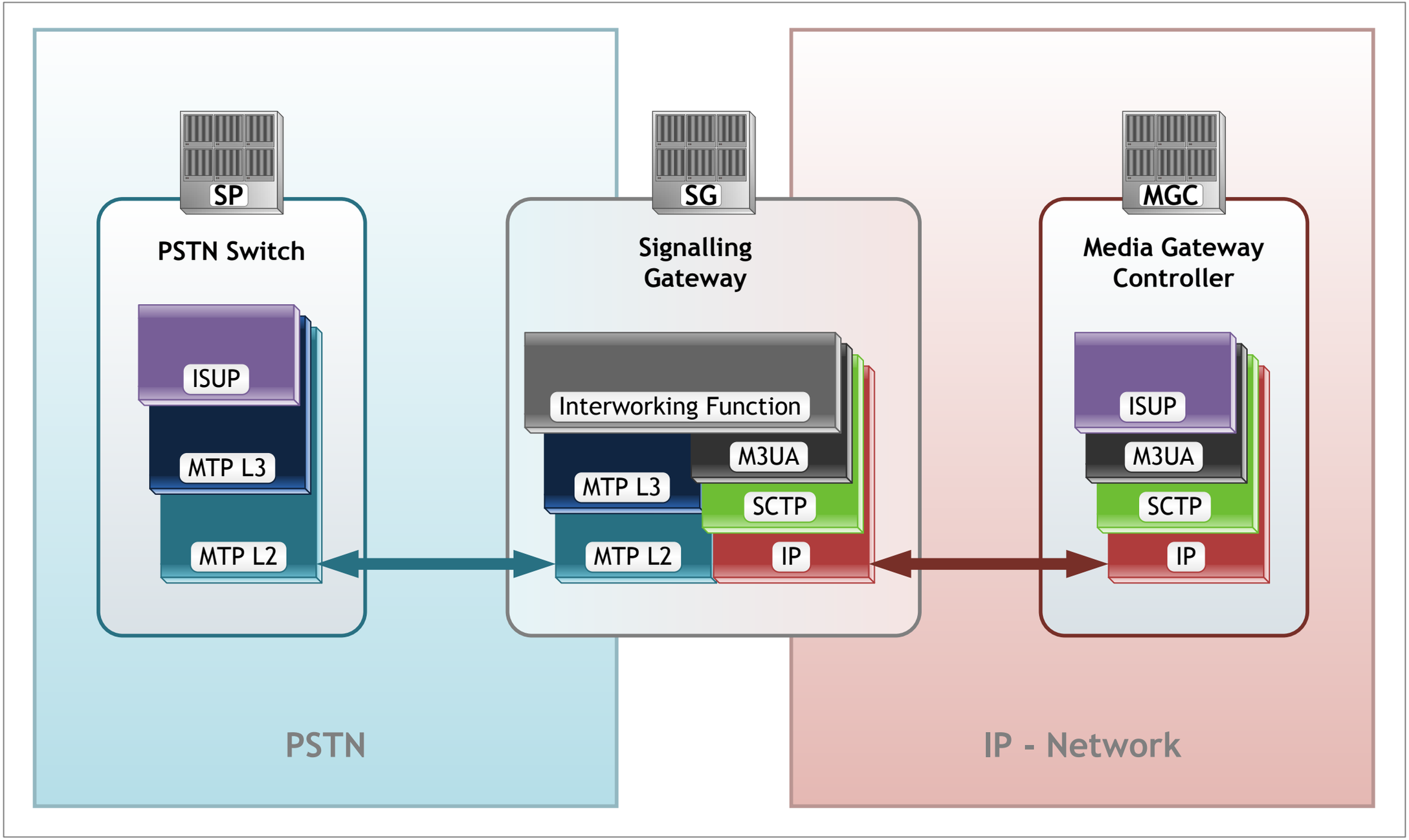
- Client wants to establish connection on a port where no process is running
- There was a crash on one side but the other side continues in sending TCP segments etc…



SCTP – Stream Control Transmission Protocol

- Reliable transport protocol operating on top of a potentially unreliable connectionless packet service such as IP.
- Offers
  - Acknowledged error-free non-duplicated transfer of datagrams (messages).
  - Detection of data corruption, loss of data and duplication of data (using checksums and sequence numbers).
  - A selective retransmission mechanism.
- Originally designed to provide a generalpurpose transport protocol for message-oriented applications.
- Transport protocol for signalling data. (IETF SIGTRAN w.g. released the SCTP standard draft document(RFC2960) in Oct. 2000.

SCTP authentic purpose



SCTP vs. TCP

- SCTP
  - Multi-homing
  - Several streams (Multistreaming) within a single connection (Association)
  - SCTP stream represents a sequence of messages
  - Resistance to flooding(DOS/syn-flood), masquerade and blind attacks
- TCP
  - A stream is referred to as a sequence of bytes
  - Multiple independent streams require multiple TCP connections

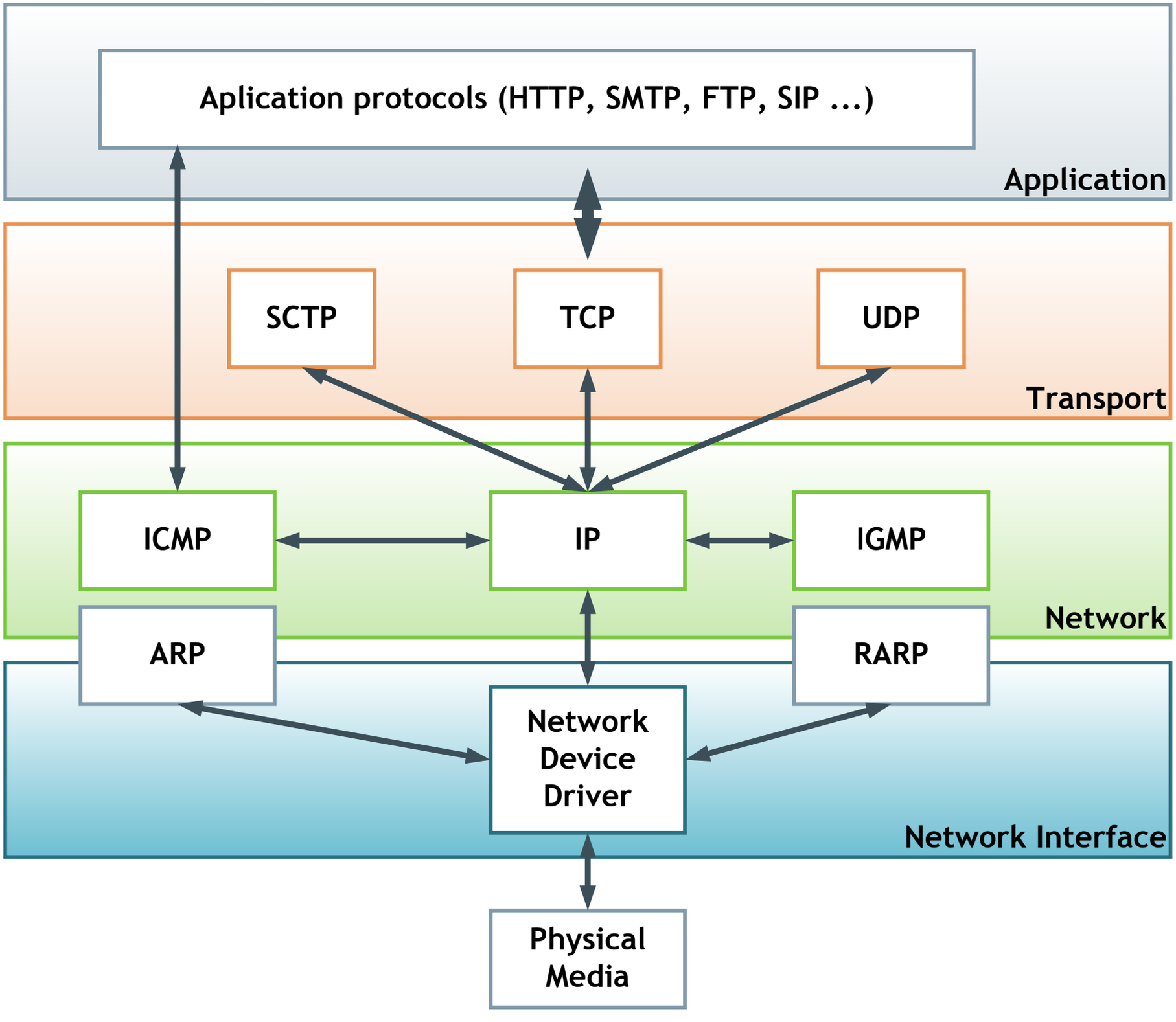
TCP and SCTP

- Both utilize similar congestion avoidance algorithms
- Both are sensitive to MITM attacks (no self security mechanisms)
- Strong security could be achieved using IPsec and TLS.

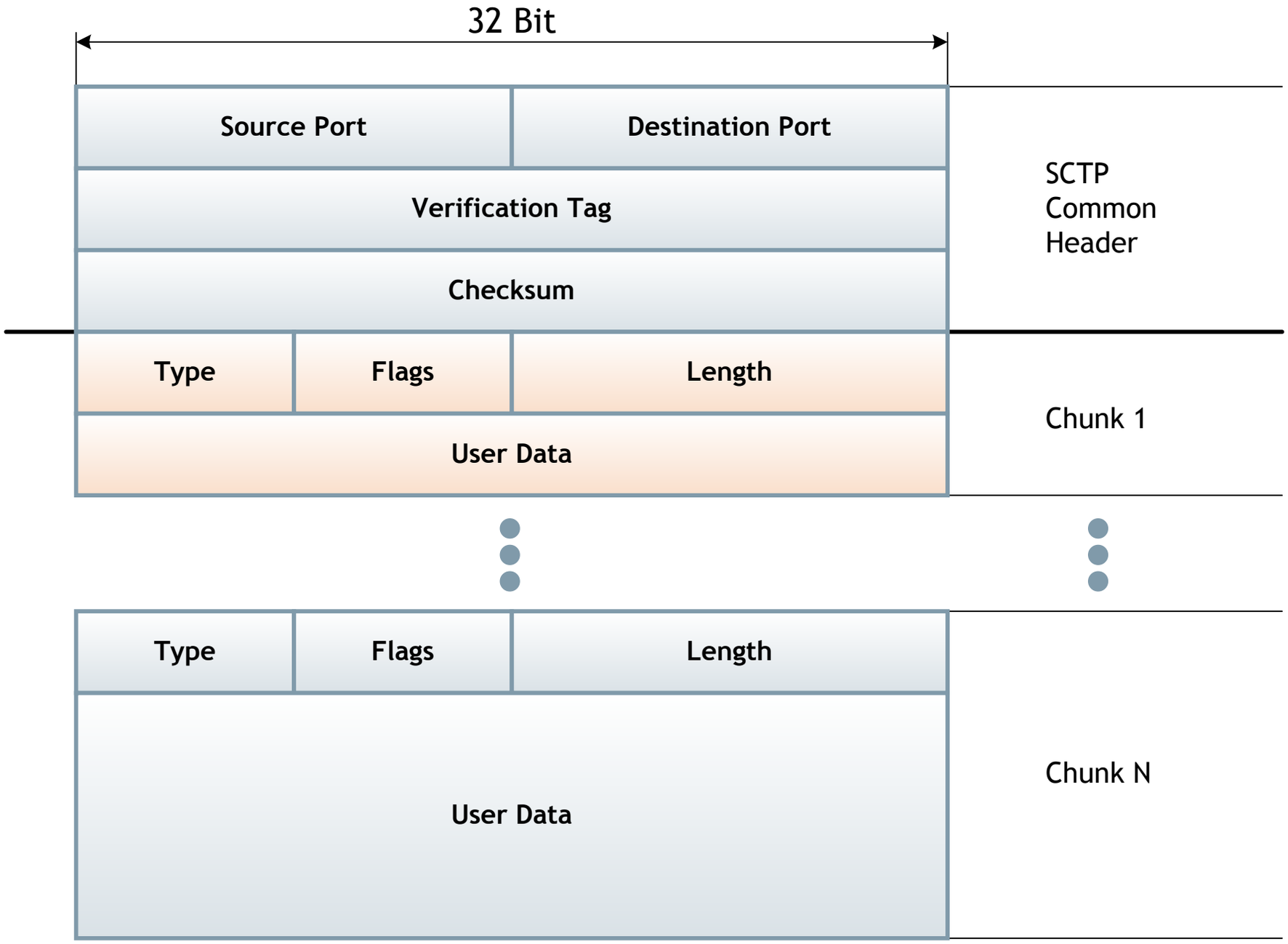
SCTP usage

- SCTP can be used as the transport protocol for applications where:
  - Monitoring of connection
  - Detection of loss of session is required.
- The SCTP path/session failure detection mechanisms, especially the heartbeat, will actively monitor the connectivity of the session.

TCP/IP Protocol Architecture



SCTP PDU-packet



CHUNK

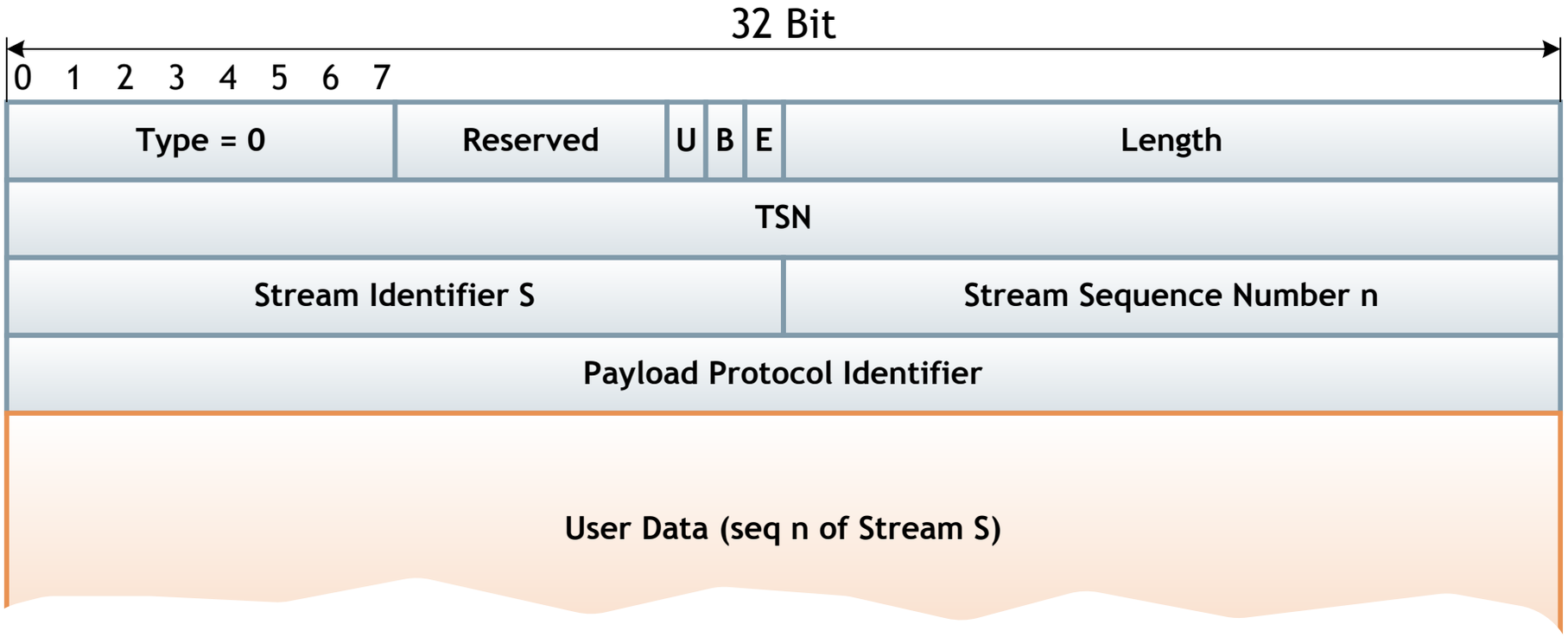
- Each chunk begins with a chunk type field, used to distinguish data chunks and different types of control chunks,
- Followed by chunk specific flags and a chunk length field needed because chunks have a variable length. The value field contains the actual payload of the chunk.

CHUNK types

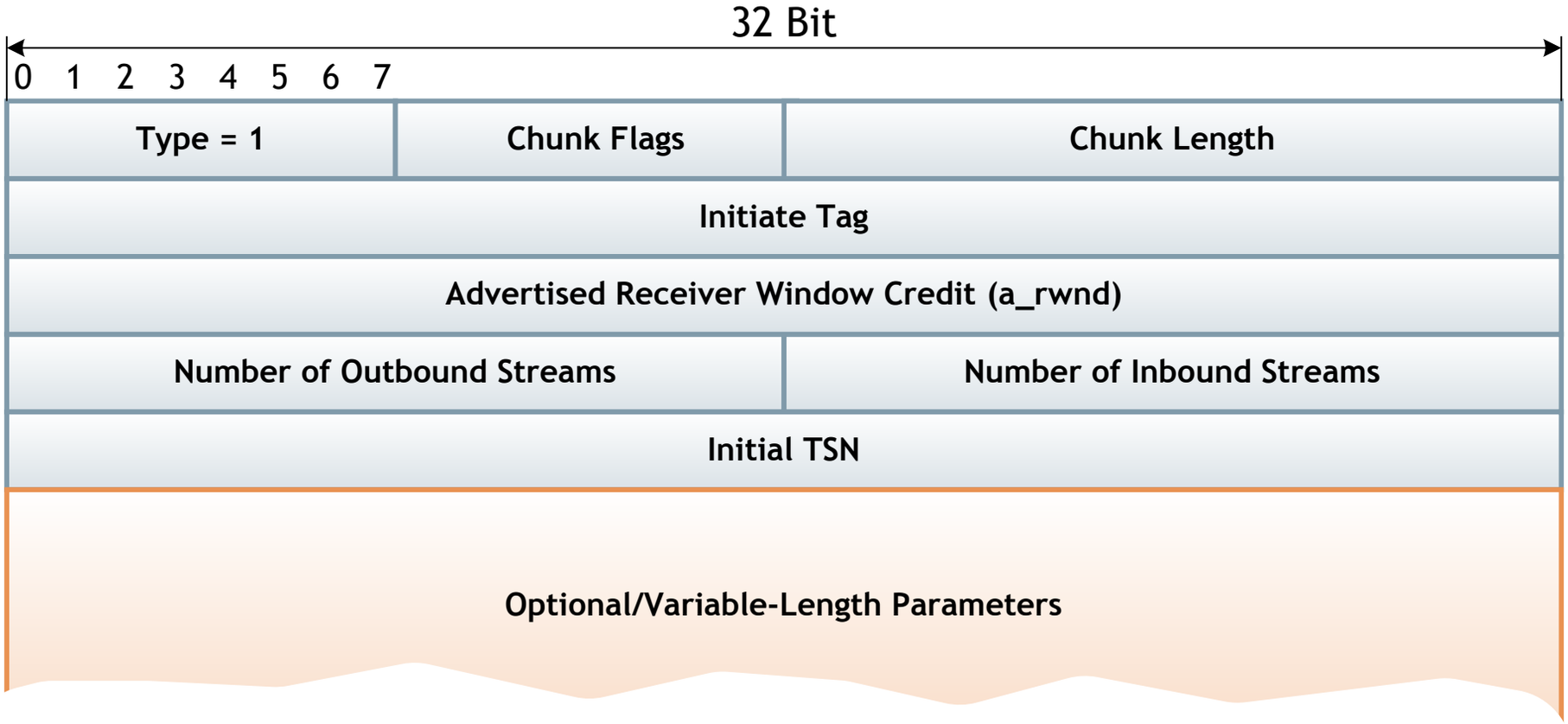
- So far there are 13 chunk types defined for standard use.

ID Value	Chunk Type
0	Payload Data (DATA)
1	Initiation (INIT)
2	Initiation Acknowledgement (INIT ACK)
3	Selective Acknowledgement (SACK)
4	Heartbeat Request (HEARTBEAT)
5	Heartbeat Acknowledgement (HEARTBEAT ACK)
6	Abort (ABORT)
7	Shutdown (SHUTDOWN)
8	Shutdown Acknowledgement (SHUTDOWN ACK)
9	Operation Error (ERROR)
10	State Cookie (COOKIE ECHO)
11	Cookie Acknowledgement (COOKIE ACK)
12	Reserved for Explicit Congestion Notification Echo (ECNE)
13	Reserved for Congestion Window Reduced (CWR)
14	Shutdown Complete (SHUTDOWN COMPLETE)
15 to 62	reserved by IETF
63	IETF-defined Chunk Extensions
64 to 126	reserved by IETF
127	IETF-defined Chunk Extensions
128 to 190	reserved by IETF
191	IETF-defined Chunk Extensions
192 to 254	reserved by IETF
255	IETF-defined Chunk Extensions

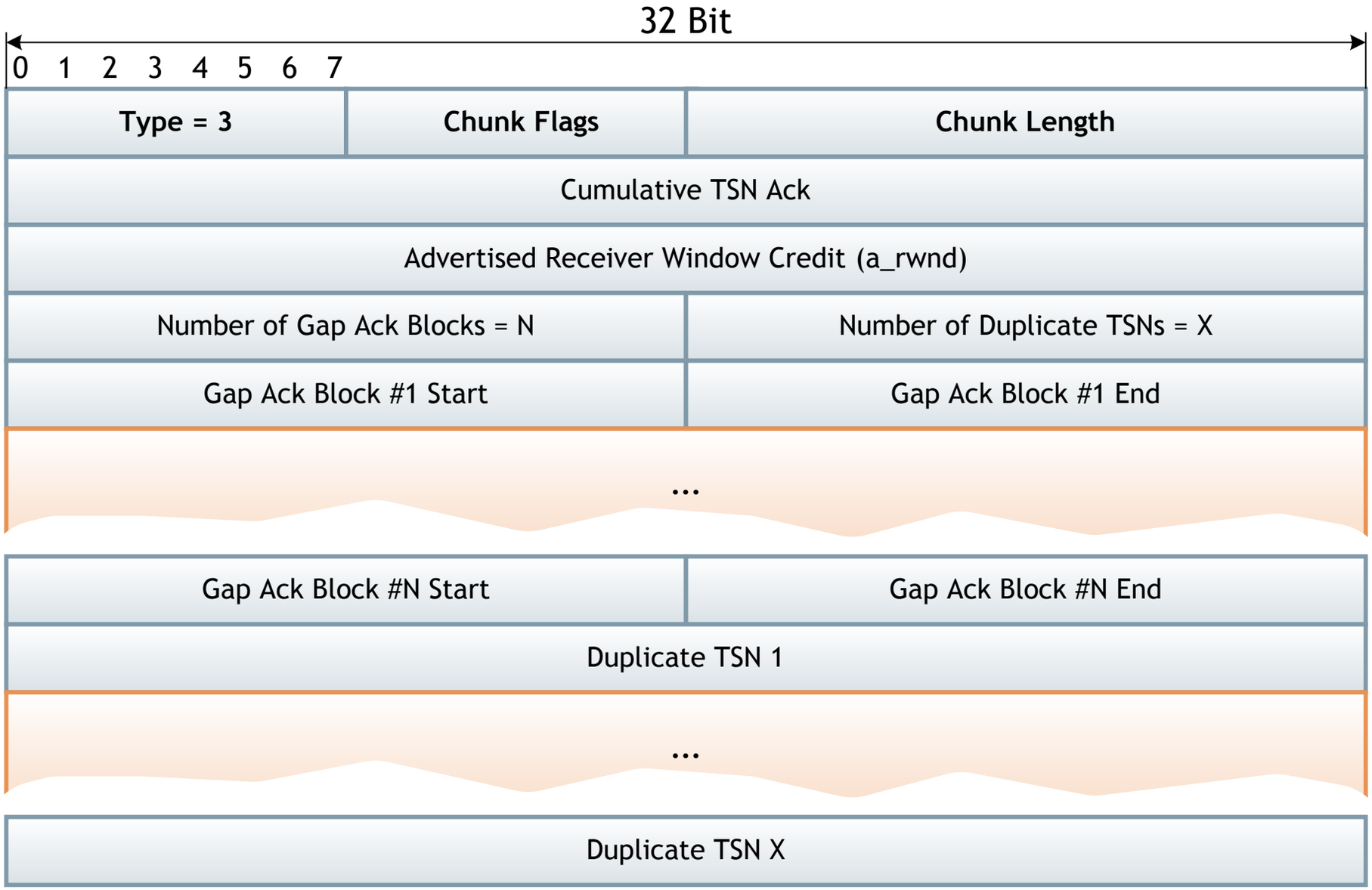
Payload (DATA) chunk



Initiation (INT) chunk



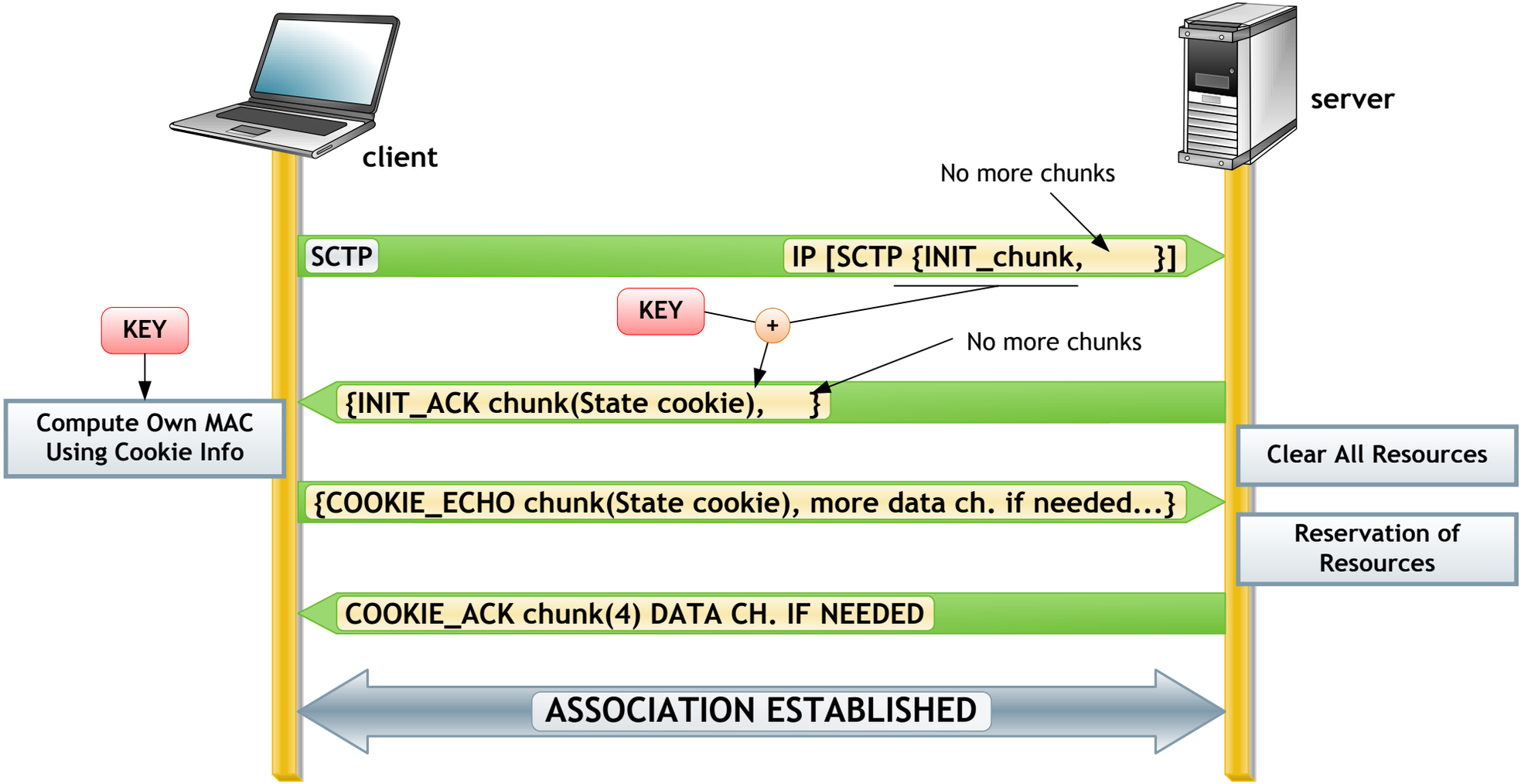
Selective Acknowledgement (SACK) chunk



SCTP association establishment

- The initialization of an association is completed on both sides after the exchange of four messages.
- The passive side (server) does not allocate resources until the third of these messages has arrived and been validated.
- This all is called four way handshake.

Four way handshake



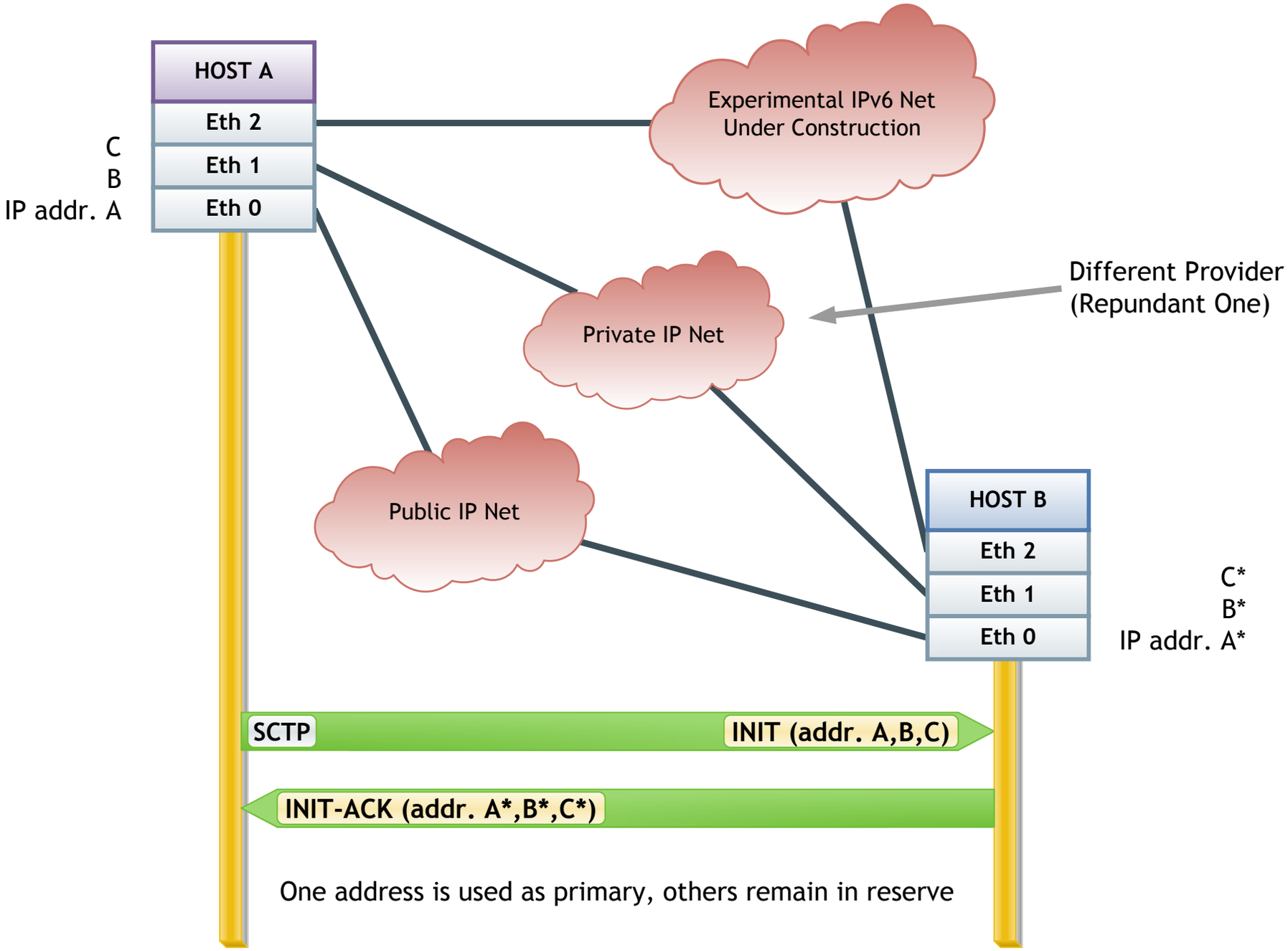
Association termination

- Graceful Termination of an Association
  - Regular association termination (using SHUTDOWN chunks) ensuring that no data is lost.
  - Both sides wait for all data to be sent and acknowledged.
- Abortion of the Association
  - One side suddenly terminates the association sending ABORT chunk
  - Both sides clear the process without waiting for ACK. of data that may be eventually still on the fly.

Important SCTP characteristics

- Flow Control
- Selective Acknowledgement
- Multihoming
- Flow Control for Multihomed Endpoints
- Congestion Control
- Slow Start and Congestion Avoidance
- Path MTU Discovery

Multihoming



SIP and its transport protocols resume

- UDP
  - Original, default and most used.
  - UDP transport support is mandatory for all SIP devices
  - Connectionless transport service, sequencingless, unreliable, with no retransmission.
  - SIP messages could be simply lost by IP layer with no indication to upper layer
  - Drawbacks are compensated by
    - Fastness
    - Minimum protocol overhead

SIP and UDP continue...

- Using UDP transport it is usual, to send simply more same SIP messages (e.g. 2-4) just for sure. Assuming that some could be lost and duplicated ones will be ignored by peer SIP entity.

- UDP itself is unable to keep NAT/state-firewall holes open, so that artificial signalization is to be generated by a host behind a NAT device (e.g. REGISTER every 85 sec.)

SIP and TCP

- TCP as transport protocol for SIP was added later to provide more reliability to signalling when needed.
- Provides reliable, connection oriented transport service with sequencing.
- Messages lost by an IP layer will be retransmitted by TCP layer automatically

SIP and TCP cont.

- Positives:
  - Reliability
  - Sequence delivery
  - TCP is able to keep NAT/firewall hole open.
- Negatives:
  - “Head of line blocking” problem
  - Much more protocol overhead than UDP
  - Some devices may not support TCP transport

SIP and SCTP

- Relatively new, still in development.
- Same reliability features as TCP, even with some improvements.
- Eliminates “Head of line blocking” problem
- Allows permanent transmission-path/signalling connection state monitoring
- Supports multi-homing (same device could be reachable by several IP addresses / routes / networks...)

SIP and SCTP cont.

- SCTP in general claims to be a main signalling transport protocol in future IP networks.
- It is assumed that SCTP importance will arise with extensive IMS/SIP based networks deployment.
- Easier parsing at application layer – no need of boundaries (Content-Length headers) etween different.

SIP and SCTP cont.

- SCTP advantages over TCP are seamless in “zero-loss-networks“ (backbone networks usually are nearly zero-loss)

TCP/SCTP and SIP

- Both transport protocols are insecure
- Security could be achieved using TLS - (Transport Layer Security)
- The TLS transport protocol provides reliable and private transport mechanism.

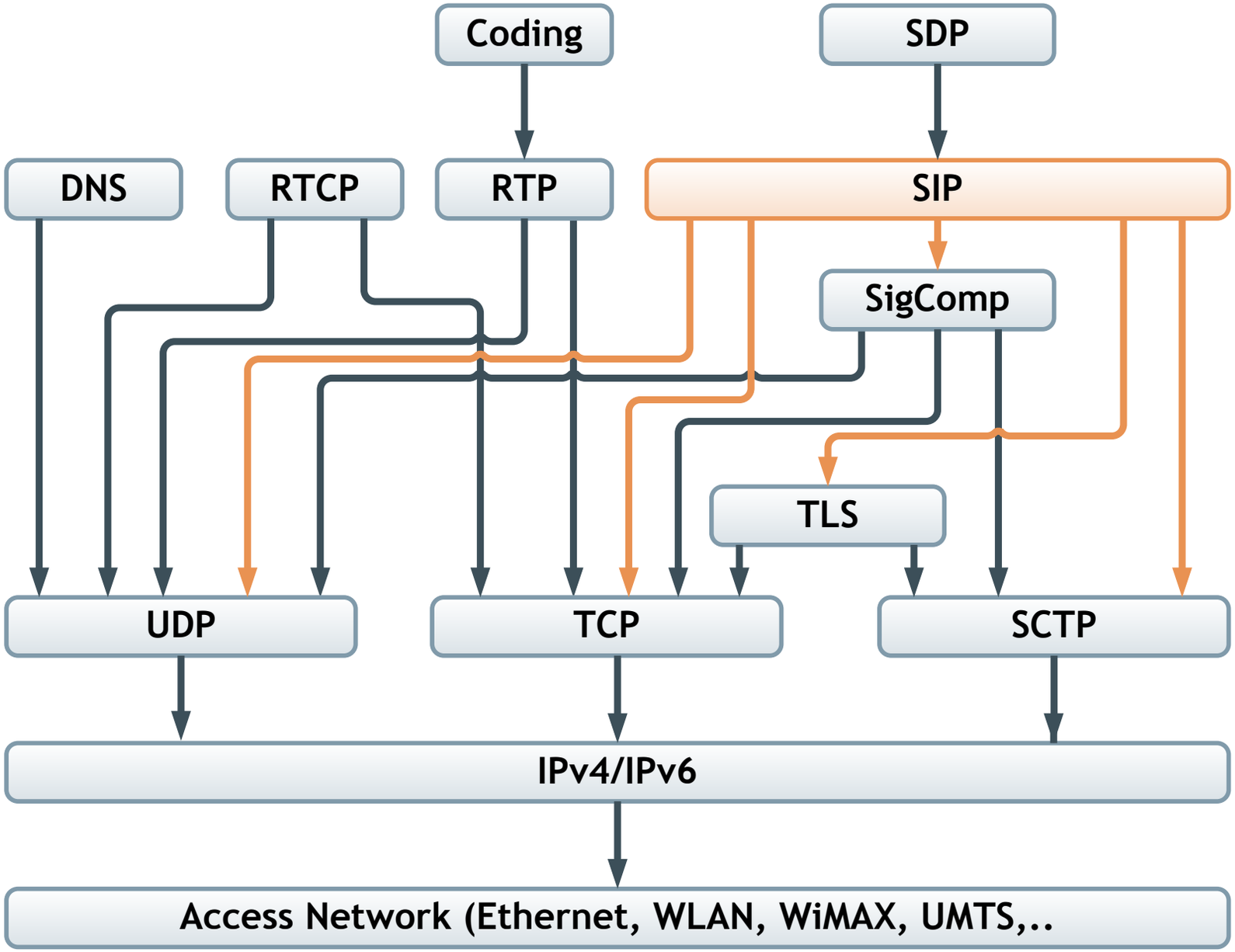
IP = best effort network

- IPv4 has no methods of QOS insurance
- Both traffic sources and their priorities are equal (first come, first served)
- IP is a best effort network = “it will give you as much as it can, but does not guarantee anything”
- Several methods of QOS insurance in IPv4 have been developed and deployed but with quite shaky results (traffic shaping....)

Congestion - the biggest scourge of IP network

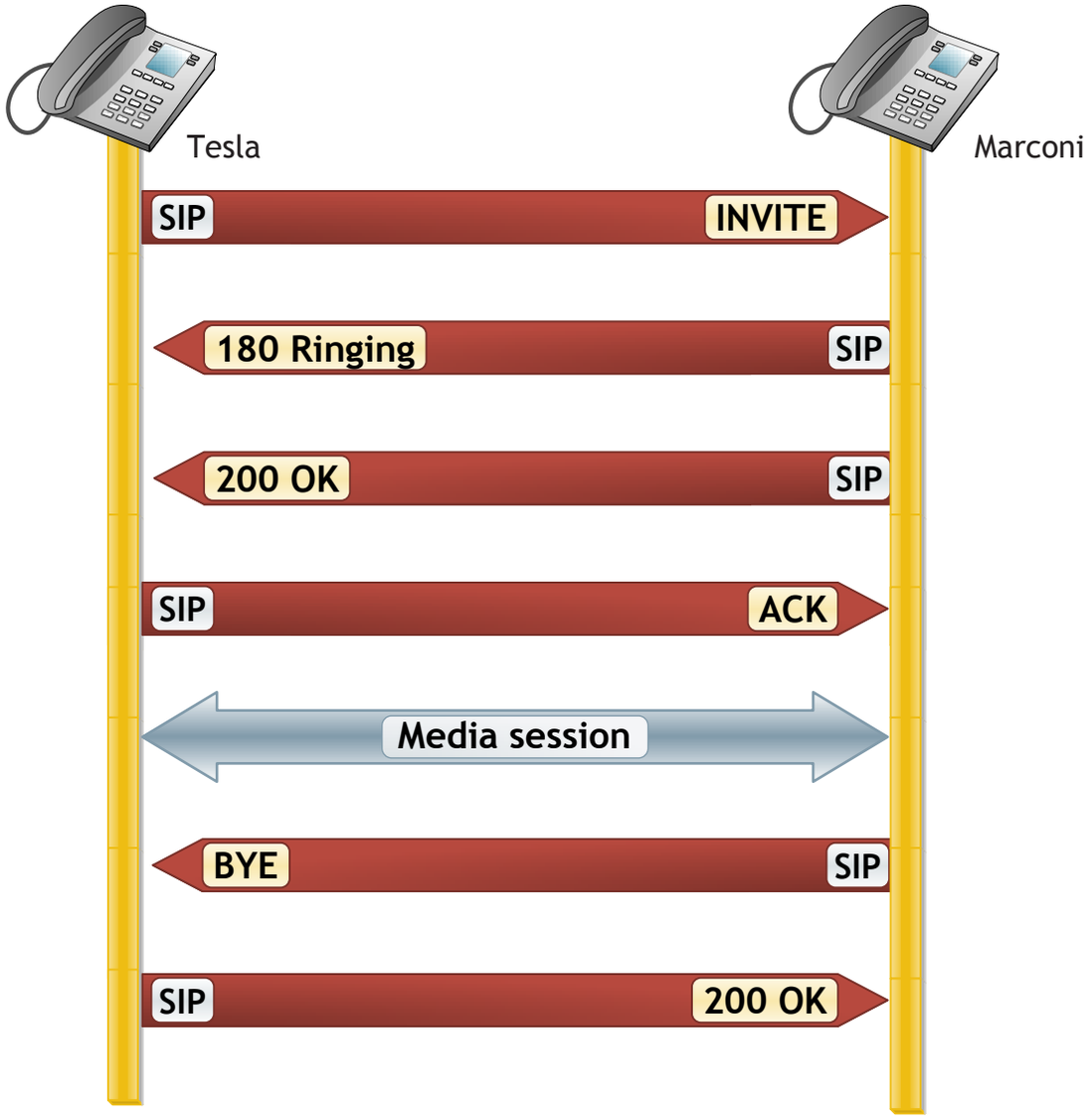
- IP network should newer be overloaded
- No overload facing mechanisms
- Starts to behave unpredictably when overloaded
- When designing an IP connection/network it is generally a good idea to over design it.
- There is an unwritten law that IP network should be at least one level faster than it seems well (when you think that 10M Ethernet would be sufficient, use 100M and so on) to void congestion/overload completely.

SIP protocol stack ones more



SIP operation

- SIP communication model is based on Requests/ Responses exchanges between SIP entities



INVITE

```
INVITE sip:marconi@radio.org SIP/2.0
Via: SIP/2.0/UDP lab.high-voltage.org:5060;branch=z9hG4bKfw19b
Max-Forwards: 70
To: G. Marconi <sip:Marconi@radio.org>
From: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341
Call-ID: 123456789@lab.high-voltage.org
CSeq: 1 INVITE
Subject: About That Power Outage...
Contact: <sip:n.tesla@lab.high-voltage.org>
Content-Type: application/sdp
Content-Length: 158

v=0
o=Tesla 2890844526 2890844526 IN IP4 lab.high-voltage.org
s=Phone Call
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

RINGING

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP lab.high-voltage.org:5060;branch=z9hG4bKfw19b
;received=100.101.102.103
To: G. Marconi <sip:marconi@radio.org>;tag=a53e42
From: Nikola Tesla <sip:n.tesla@high-voltage.org>>;tag=76341
Call-ID: 123456789@lab.high-voltage.org
CSeq: 1 INVITE
Contact: <sip:marconi@tower.radio.org>
Content-Length: 0
```

OK

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP lab.high-voltage.org:5060;branch=z9hG4bKfw19b
;received=100.101.102.103
To: G. Marconi <sip:marconi@radio.org>;tag=a53e42
From: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341
Call-ID: 123456789@lab.high-voltage.org
CSeq: 1 INVITE
Contact: <sip:marconi@tower.radio.org>
Content-Type: application/sdp
Content-Length: 155
v=0
o=Marconi 2890844528 2890844528 IN IP4 tower.radio.org
s=Phone Call
c=IN IP4 200.201.202.203
t=0 0
m=audio 60000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

ACK

```
ACK sip:marconi@tower.radio.org SIP/2.0
Via: SIP/2.0/UDP lab.high-voltage.org:5060;branch=z9hG4bK321g
Max-Forwards: 70
To: G. Marconi <sip:marconi@radio.org>;tag=a53e42
From: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341
Call-ID: 123456789@lab.high-voltage.org
CSeq: 1 ACK
Content-Length: 0
```

BYE and OK

```
BYE sip:n.tesla@lab.high-voltage.org SIP/2.0
Via: SIP/2.0/UDP tower.radio.org:5060;branch=z9hG4bK392kf
Max-Forwards: 70
To: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341
From: G. Marconi <sip:marconi@radio.org>;tag=a53e42
Call-ID: 123456789@lab.high-voltage.org
CSeq: 1 BYE
Content-Length: 0
```

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP tower.radio.org:5060;branch=z9hG4bK392kf
;received=200.201.202.203
To: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341
From: G. Marconi <sip:marconi@radio.org>;tag=a53e42
Call-ID: 123456789@lab.high-voltage.org
CSeq: 1 BYE
Content-Length: 0
```

SIP addressing

- SIP URLs are used within SIP messages to indicate the originator (From), current destination (Request-URI) and final recipient (To) of a SIP request, and to specify redirection addresses (Contact).
- Some examples for SIP URLS:

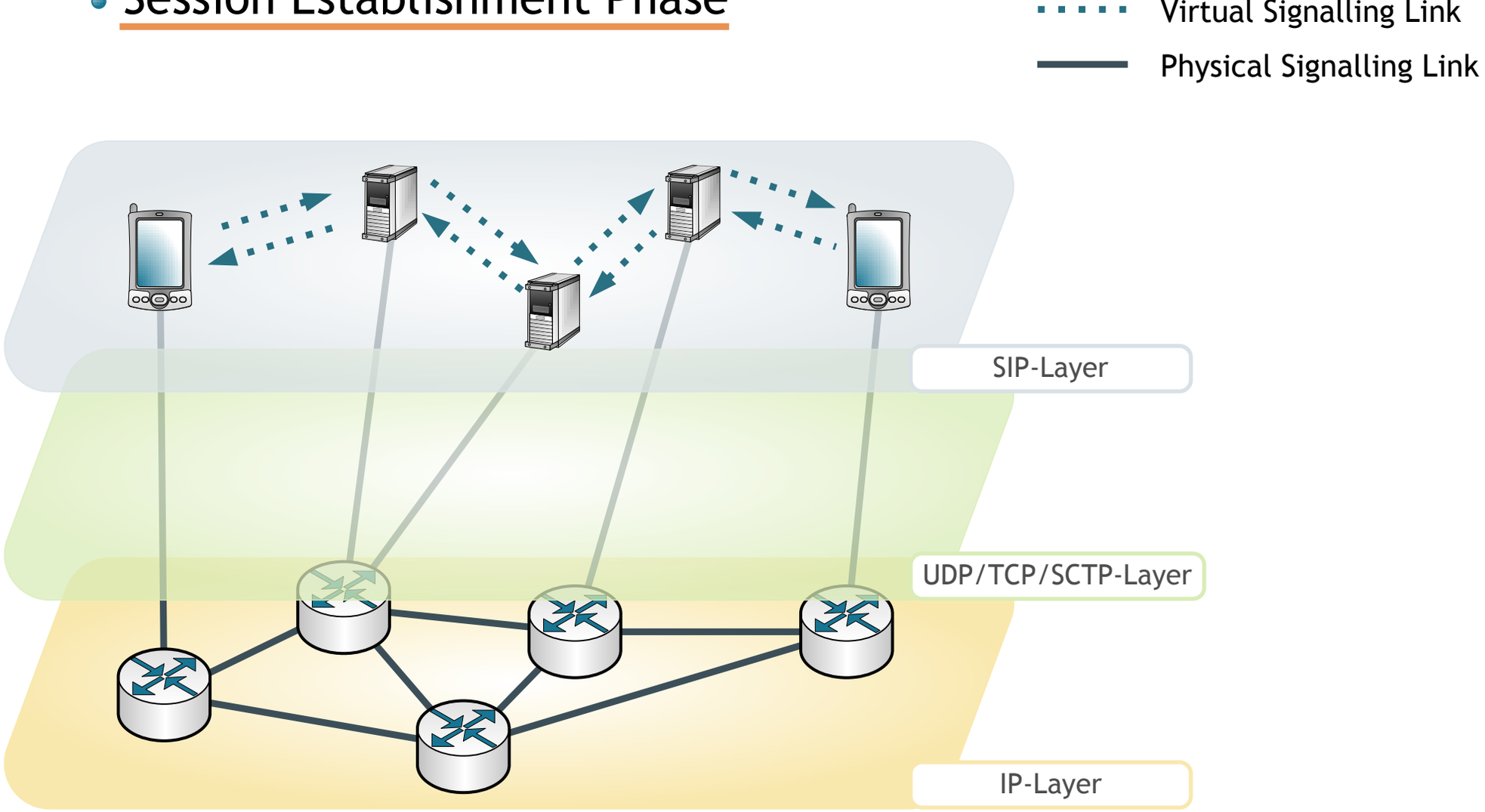
```
sip:1212@gateway.com sip:alice@10.1.2.3
sip:alice@example.com
sip:alice%40example.com@gateway.com
```

Philosophy of SIP-Operation

- Session Establishment Phase
- During the session establishment phase, the two peers may require a number of SIP-proxy servers to route and handle session establishment requests
- SIP-layer requires physical transport of the SIP messages through lower layers

Philosophy of SIP-Operation

- Session Establishment Phase

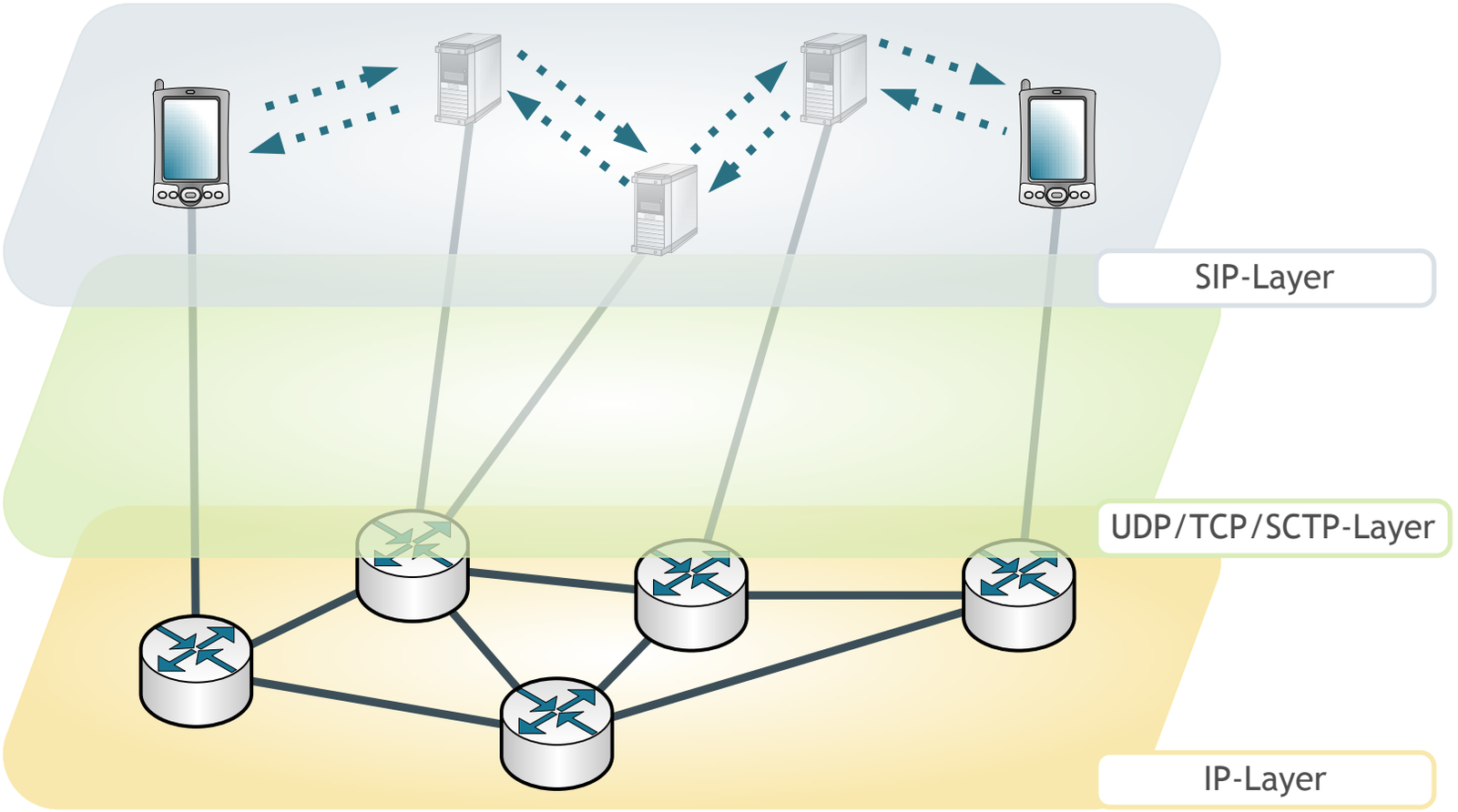


Philosophy of SIP-Operation

- Session Completion Phase
- SIP-proxy servers drop out of the communication chain
- After the setup of the communication channel, there is no more involvement required of the proxies

Philosophy of SIP-Operation

- Session Completion Phase

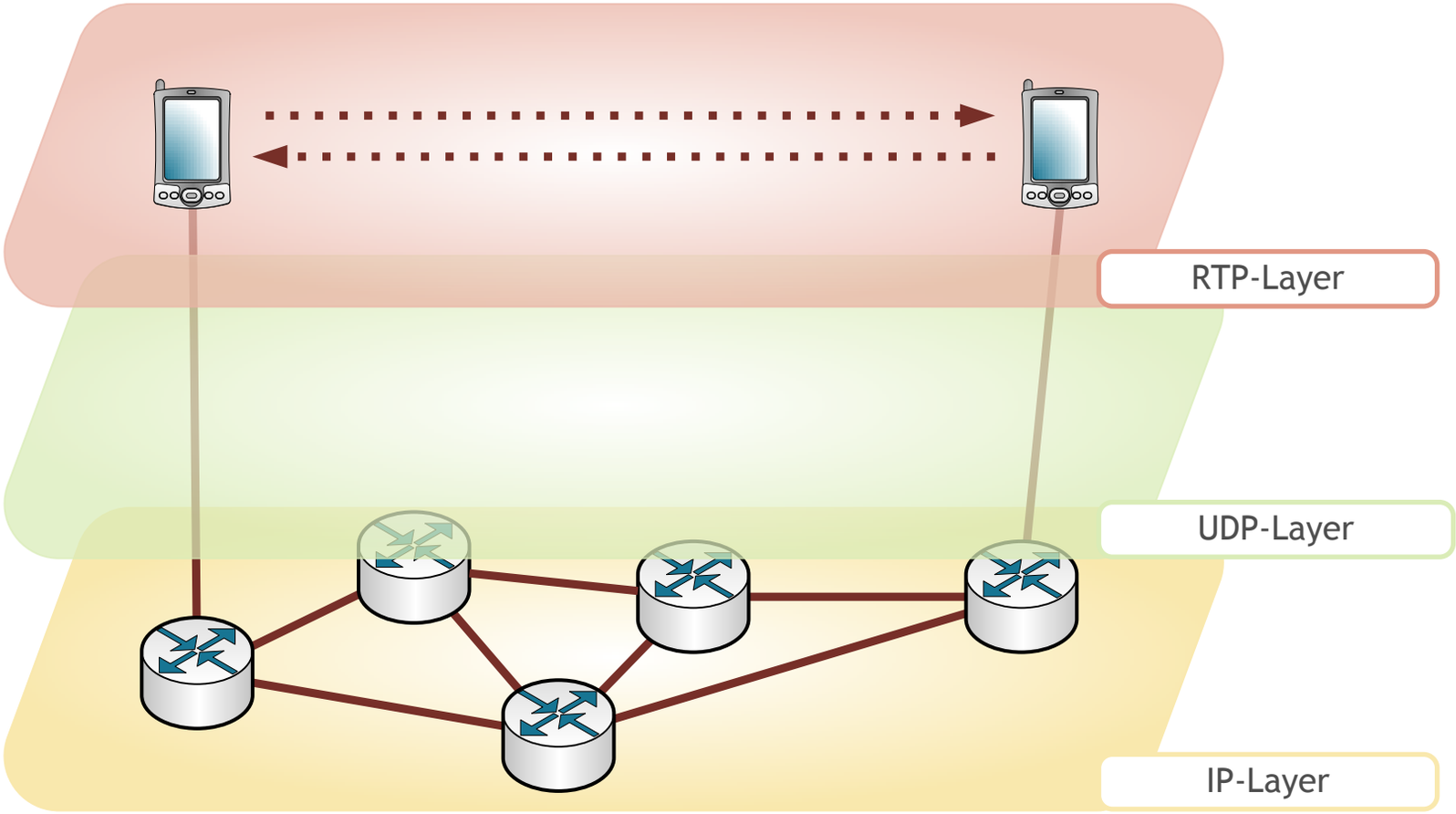


Philosophy of SIP-Operation

- Session Active Phase
- Two peers exchange data, e.g. embedded into RTP-frames
- Data frames are packed into UDP-frames and IP-frames of which every single one can take a different route between the two peers
- Note that there is no SIP-proxy involved in RTP transport

Philosophy of SIP-Operation

- Session Active Phase



Summary

- SIP is an application protocol
- SIP is used to initiate, modify and tear-down media sessions.
- SIP uses UDP/TCP/SCTP or TLS as transport layer
- SIP does not care about established media session any more, it is only used when session modification or tear-down is needed

SIP methods

- SIP requests or methods are considered “verbs” in the protocol, since they request a specific action to be taken by another user agent or server.

SIP methods

- original six SIP methods:
  - INVITE, REGISTER, BYE, ACK, CANCEL, OPTIONS
- methods are described in separate RFCs. :
  - REFER, SUBSCRIBE, NOTIFY, MESSAGE, UPDATE, INFO, PRACK

INVITE

- The INVITE method is used to establish media sessions between user agents. In telephony, it is similar to a Setup message in ISDN or an initial address message (IAM) in ISUP.
- An INVITE usually has a message body containing the media information of the caller.
- The message body can also contain other session information such as quality of service (QoS) or security information.

INVITE

- If an INVITE does not contain media information, the ACK contains the media information of the UAC.
- If the media information contained in the ACK is not acceptable, then the called party must send BYE to cancel the session
- CANCEL cannot be sent because the session is already established.

INVITE

- Mandatory Headers in an INVITE Request
  - Call-ID
  - CSeq
  - From
  - To
  - Via
  - Contact
  - Max-Forwards

INVITE

- A UAC that originates an INVITE to establish a dialog creates a globally unique Call-ID for the duration of the call.
- A CSeq count is initialized (which need not be set to 1, but must be an integer)
- Cseq is incremented for each new request for the same Call-ID.
- The To and From headers are populated with the remote and local addresses.

Tags

- A From tag is included in the INVITE by UAC
- UAS includes a To tag in any responses
- To tag from 200 OK response to an INVITE is used in the To header field of the ACK and all future requests within the dialog.
- The combination of the
  - To tag,
  - From tag,
  - and Call-ID is the unique identifier for the dialog.

Re-INVITE

- An INVITE sent for an existing dialog references the same Call-ID as the original INVITE and contains the same To and From tags.
- Called Re-INVITE, is used to change the session characteristics or refresh the state of the dialog.
- The CSeq command sequence number is incremented so that a UAS can distinguish the Re-INVITE from a retransmission of the original INVITE.

Re-INVITE

- If a re-INVITE is refused or fails, the session continues as if the re-INVITE had never been sent!
- A re-INVITE must not be sent by a UAC until a final response to the initial INVITE has been received instead, an UPDATE request can be sent.

REGISTER

- The REGISTER method is used by a user agent to notify a SIP network of its current Contact URI (IP address) and the URI that should have requests routed to this Contact
- Registration is not required to enable a user agent to use a proxy server for outgoing calls.
- Registration is necessary, for receiving the incoming calls.

Types of Registrar Actions and Contact Headers

Request Headers	Registrar Action
Contact: * Expires: 0	Cancel all registrations
Contact: sip:galvani@bologna.edu.it; expires=30	Add Contact to current registrations; registration expires in 30 minutes
Contact: sip:galvani@bologna.edu.it Expires: 30	Add Contact to current registrations; registration expires in 30 minutes
Contact: sip:galvani@bolognauni.edu; expires=45	Add all Contacts to registrations in preference order listed; first one expires in 45 minutes, second in 30 minutes
Contact: sip:l.galvani@bologna.it Expires: 30	
Contact:mailto:galvani@bologna.edu.it; q=0.1	Add Contacts to current registrations using specified preference SIP requests
No Contact header present	should be proxied; SIP URI expires in 60 minutes (default) mailto URL does not expire
No Contact header present	Return all current registrations in response

Third-party registration request

```
REGISTER sip:registrar.athens.gr SIP/2.0
Via: SIP/2.0/UDP 201.202.203.204:5060;branch=z9hG4bK313
Max-Forwards:70
To: sip:euclid@athens.gr
From: <sip:secretary@academy.athens.gr>;tag=543131
Call-ID: 2000-July-07-23:59:59.1234@201.202.203.204
CSeq: 1 REGISTER
Contact: sip:euclid@parthenon.athens.gr
Contact: mailto:euclid@geometry.org
Content-Length: 0
```

Mandatory Headers in a REGISTER Request

- Call-ID
- CSeq
- From
- To
- Via
- Max-Forwards

BYE

- The BYE method is used to terminate an established media session.
- In telephony similar to a release message.
- A session is considered established if:
  - INVITE has received a success class response (2xx)
  - an ACK has been sent

BYE

- BYE is sent only by user agents participating in the session, never by proxies or other third parties.
- BYE is an end-to-end method, so responses are only generated by the other user agent.
- A user agent responds with a 481Dialog/Transaction Does Not Exist to a BYE for an unknown dialog.

Mandatory Headers in a BYE Request

- Call-ID
- CSseq
- From
- To
- Via
- Max-Forwards

ACK

- The ACK method is used to acknowledge final responses to INVITE requests.
- Final responses to all other requests are never acknowledged.
- The Cseq number is never incremented for an ACK,
- CSeq method is changed to ACK.

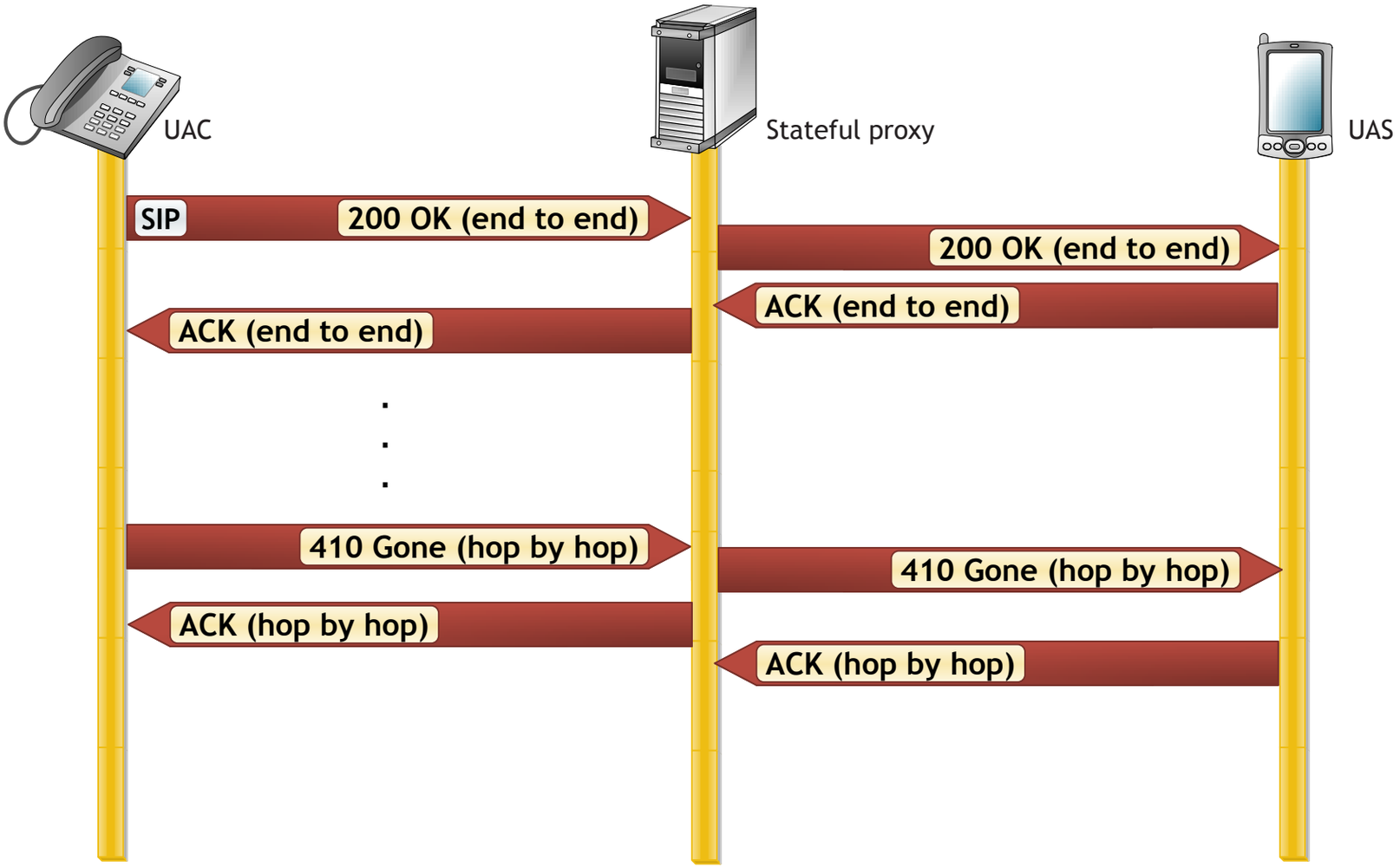
ACK

- An ACK may contain an application/sdp message body.
- This is permitted if the initial INVITE did not contain a SDP message body
- If the INVITE contained a message body, the ACK may not contain a message body!
- The ACK may not be used to modify a media description that has already been sent in the initial INVITE
- re-INVITE must be used for this purpose.

ACK

- For 2xx responses, the ACK is end-to-end
- For all other final responses it is done on a hop-by-hop basis (assumes that stateful proxies are involved).
- The end-to-end nature of ACKs to 2xx responses allows a message body to be transported.
- An ACK generated in a hop-by-hop acknowledgment will contain just a single Via header with the address of the proxy server generating the ACK.

End-to-end vs. hop-by-hop



ACK and brach-ID

- A stateful proxy receiving an ACK message must determine whether or not the ACK should be forwarded downstream to another proxy or user agent or not.
- Compare the branch ID for a match pending transaction branch Ids.
  - If there is not an exact match, the ACK is proxied toward the UAS.
  - Otherwise, the ACK is for this hop and is not forwarded by the proxy.

Mandatory Headers in a ACK Request

- Call-ID
- CSseq
- From
- To
- Via
- Max-Forwards

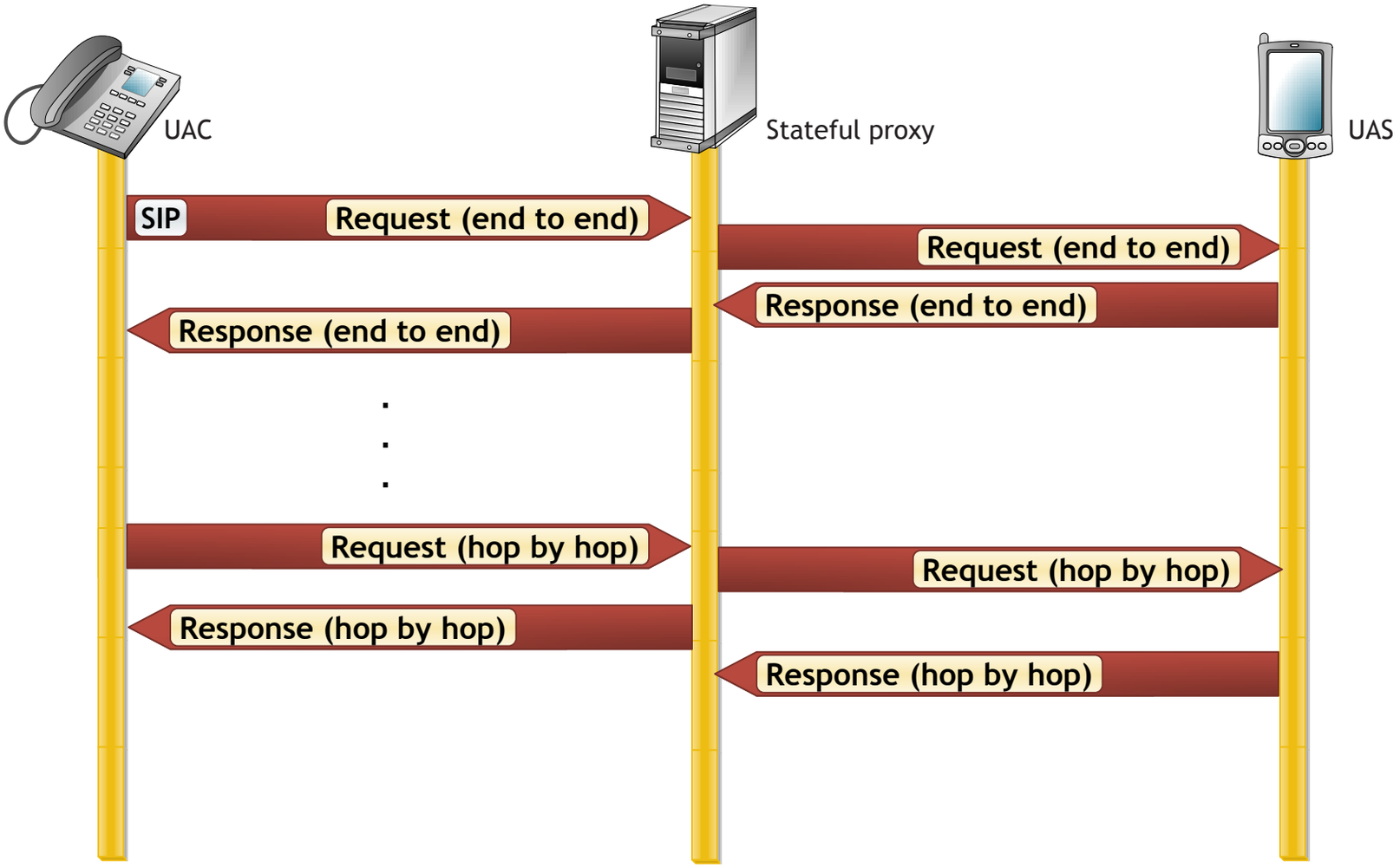
CANCEL

- The CANCEL method is used to terminate pending searches or call attempts.
- Can be generated by either user agents or proxy servers provided that a 1xx response containing a tag has been received, but no final response has been received.

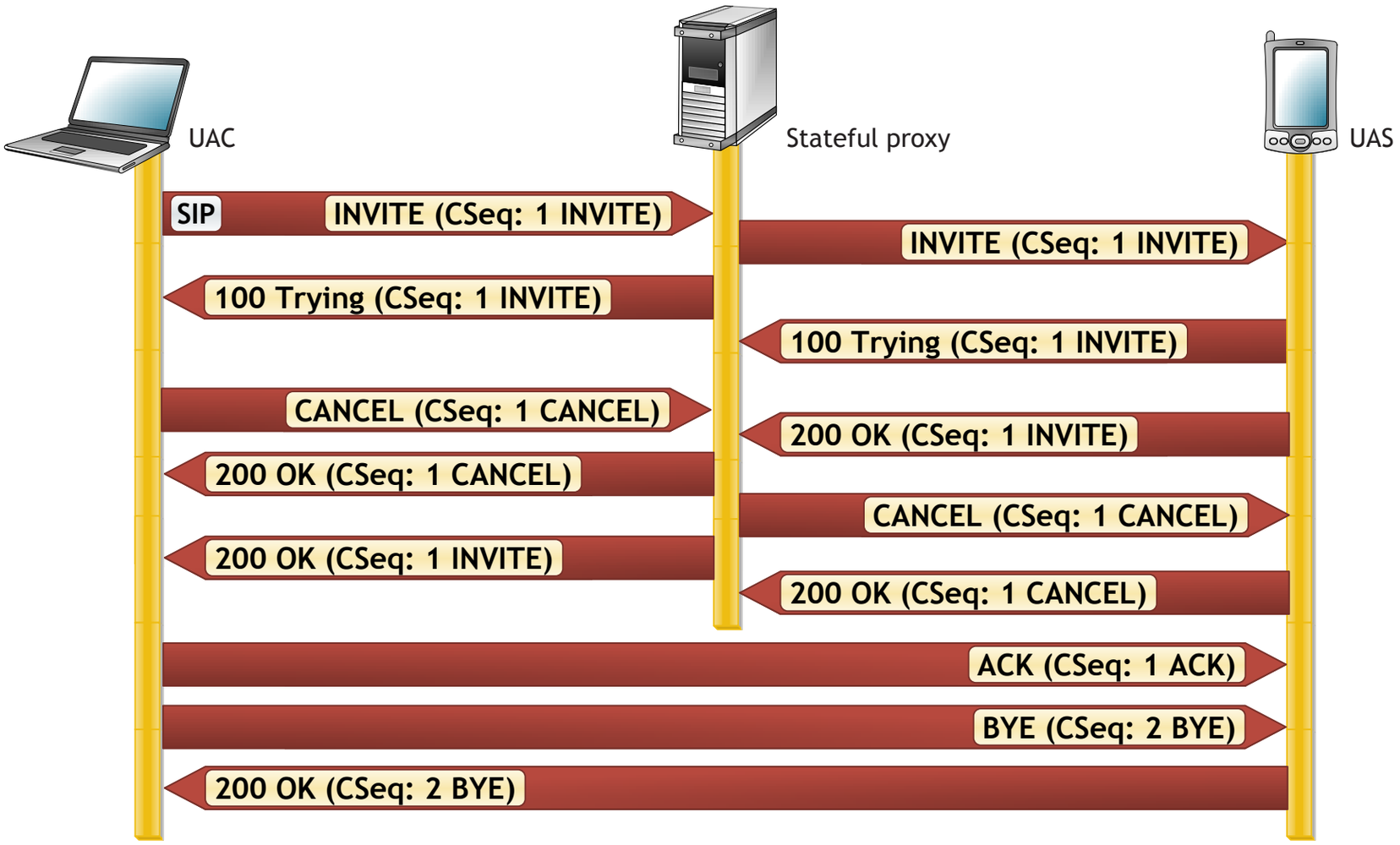
CANCEL

- UA uses the method to cancel a pending call attempt it had earlier initiated.
- A forking proxy can use the method to cancel pending parallel branches after a successful response has been proxied back to the UAC.
- CANCEL is a hop-by-hop request and receives a response generated by the next stateful element.

CANCEL is hop by hop request



Race condition in call cancellation



Mandatory Headers in a CANCEL Request

- Call-ID
- CSseq
- From
- To
- Via
- Max-Forwards

OPTIONS

- The OPTIONS method is used to query a user agent or server about its capabilities and discover its current availability
- The response to the request lists the capabilities of the user agent or server.
- A user agent or server responds to the request as it would to an INVITE

OPTIONS

- A success class (2xx) response can contain Allow, Accept, Accept-Encoding, Accept Language, and Supported headers indicating its capabilities.
- OPTIONS request may not contain a message body

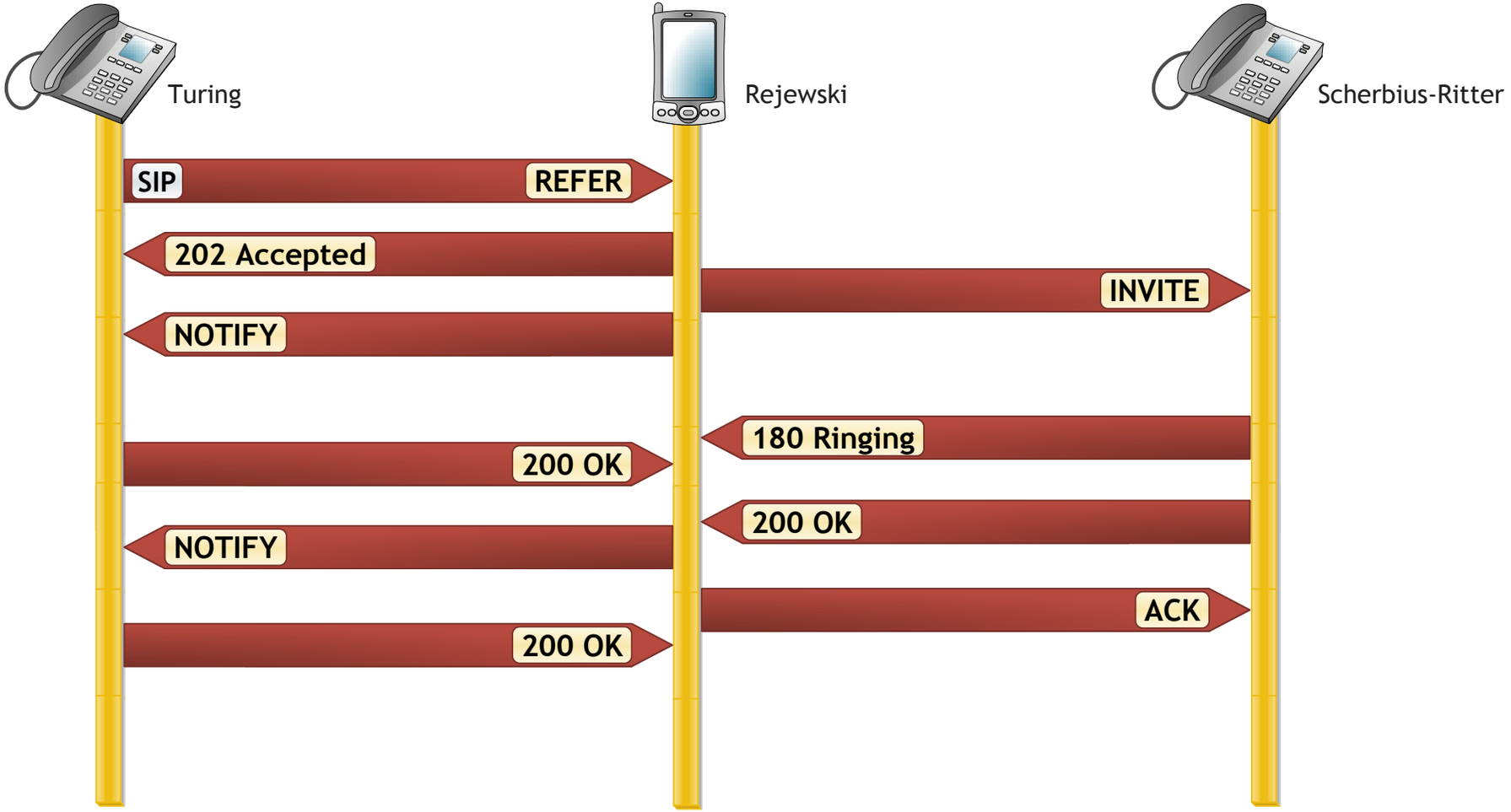
OPTIONS

```
OPTIONS sip:user@carrier.com SIP/2.0
Via: SIP/2.0/UDP cavendish.kings.cambridge.edu.uk
;branch=z9hG4bK1834
Max-Forwards:70
To: <sip:user@proxy.carrier.com>
From: J.C. Maxwell <sip:james.maxwell@kings.cambridge.edu.uk>
;tag=34
Call-ID: 9352812@cavendish.kings.cambridge.edu.uk
CSeq: 1 OPTIONS
Content-Length: 0
SIP/2.0 200 OK
Via: SIP/2.0/UDP cavendish.kings.cambridge.edu.uk;tag=512A6
;branch=z9hG4bK0834 ;received=192.0.0.2
To: <sip:user@proxy.carrier.com>;tag=432
From: J.C. Maxwell <sip:james.maxwell@kings.cambridge.edu.uk>
;tag=34
Call-ID: 9352812@cavendish.kings.cambridge.edu.uk
CSeq: 1 OPTIONS
Allow: INVITE, OPTIONS, ACK, BYE, CANCEL, REFER
Accept-Language: en, de, fr
Content-Length: ...
Content-Type: application/sdp
v=0
etc...
```

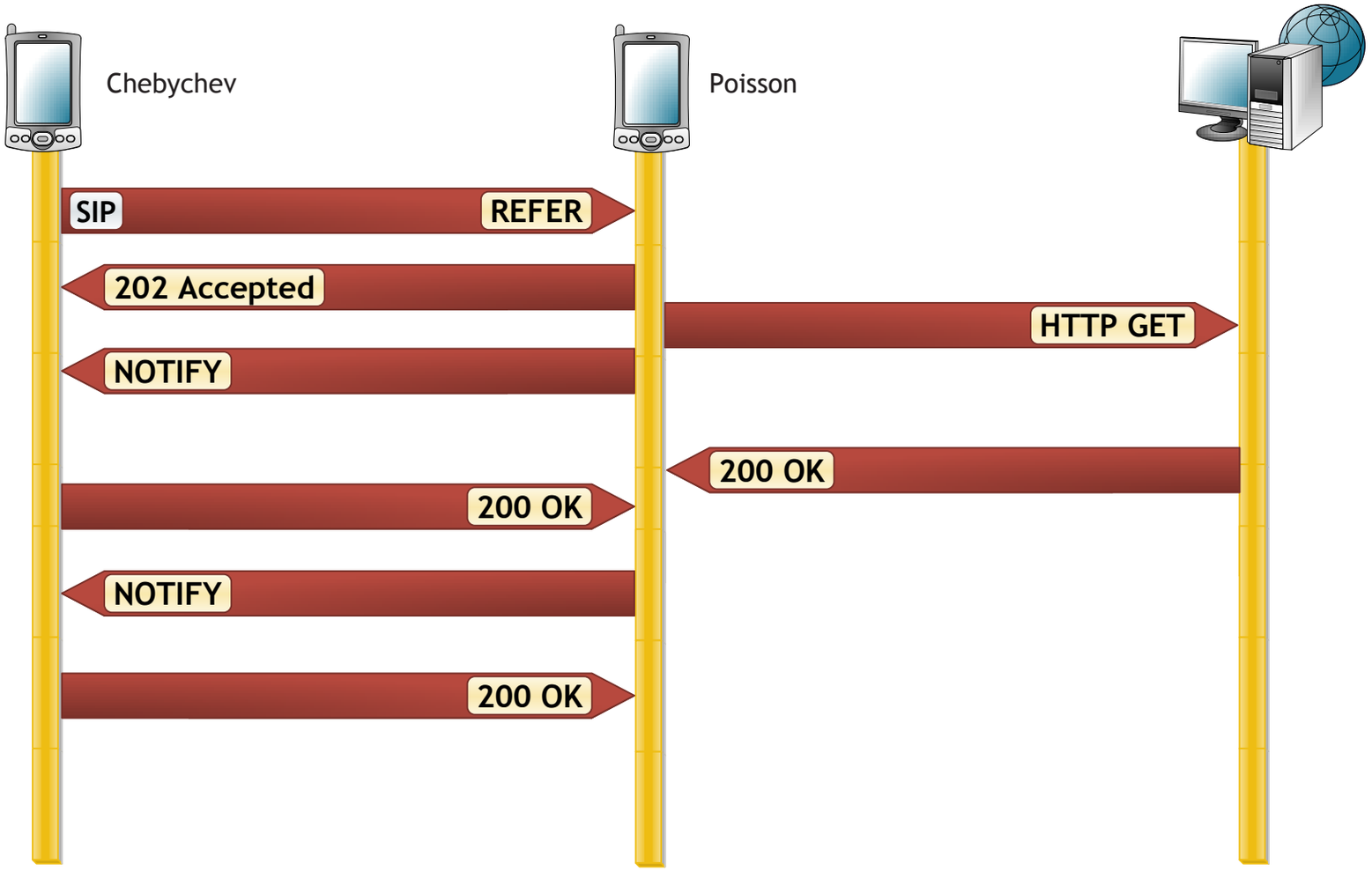
REFER

- The REFER method (RFC 3515) is used by a user agent to request another user agent to access a URI or URL resource
- Can be sent either inside or outside an existing dialog.

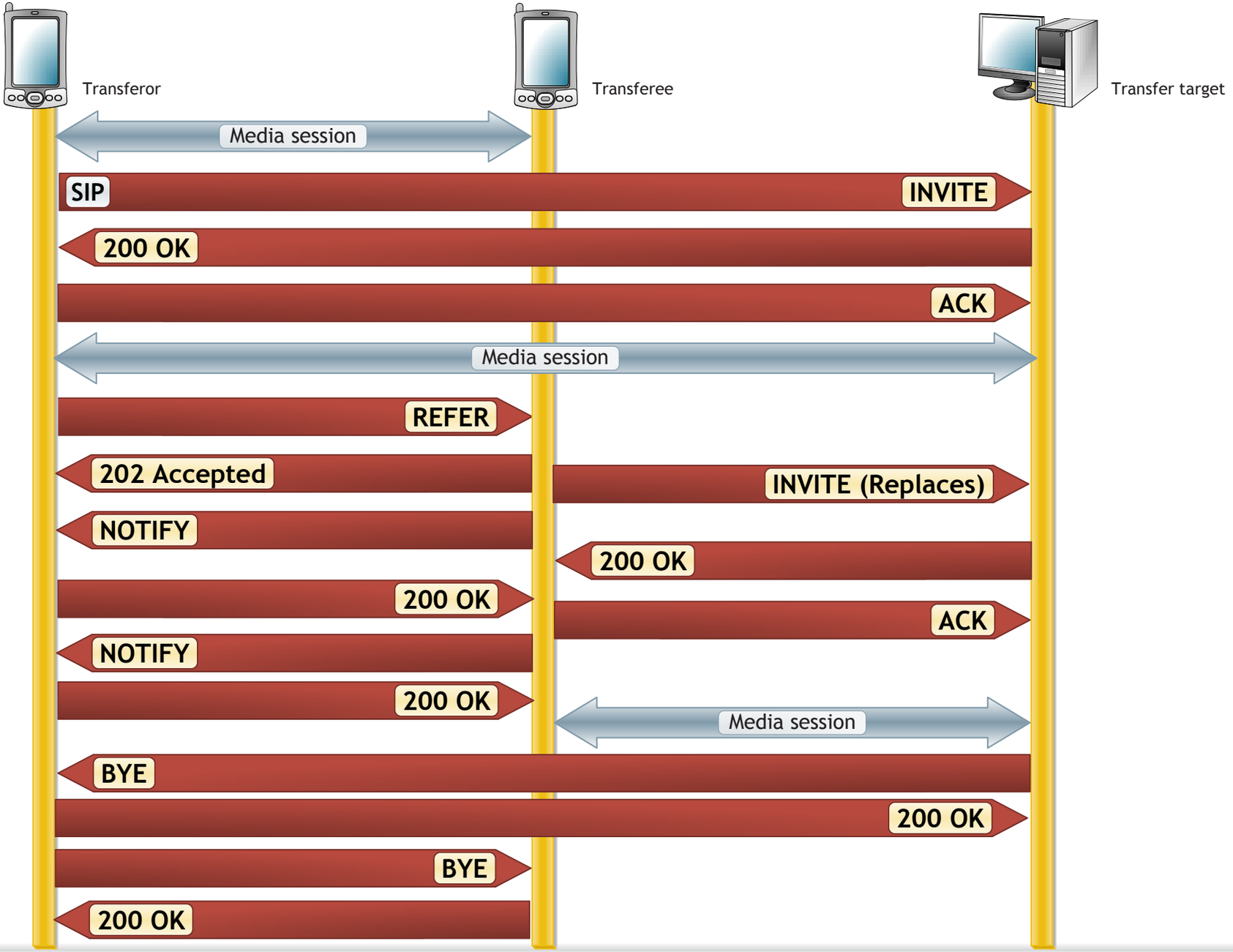
REFER example call flow



REFER example used to push Web page.



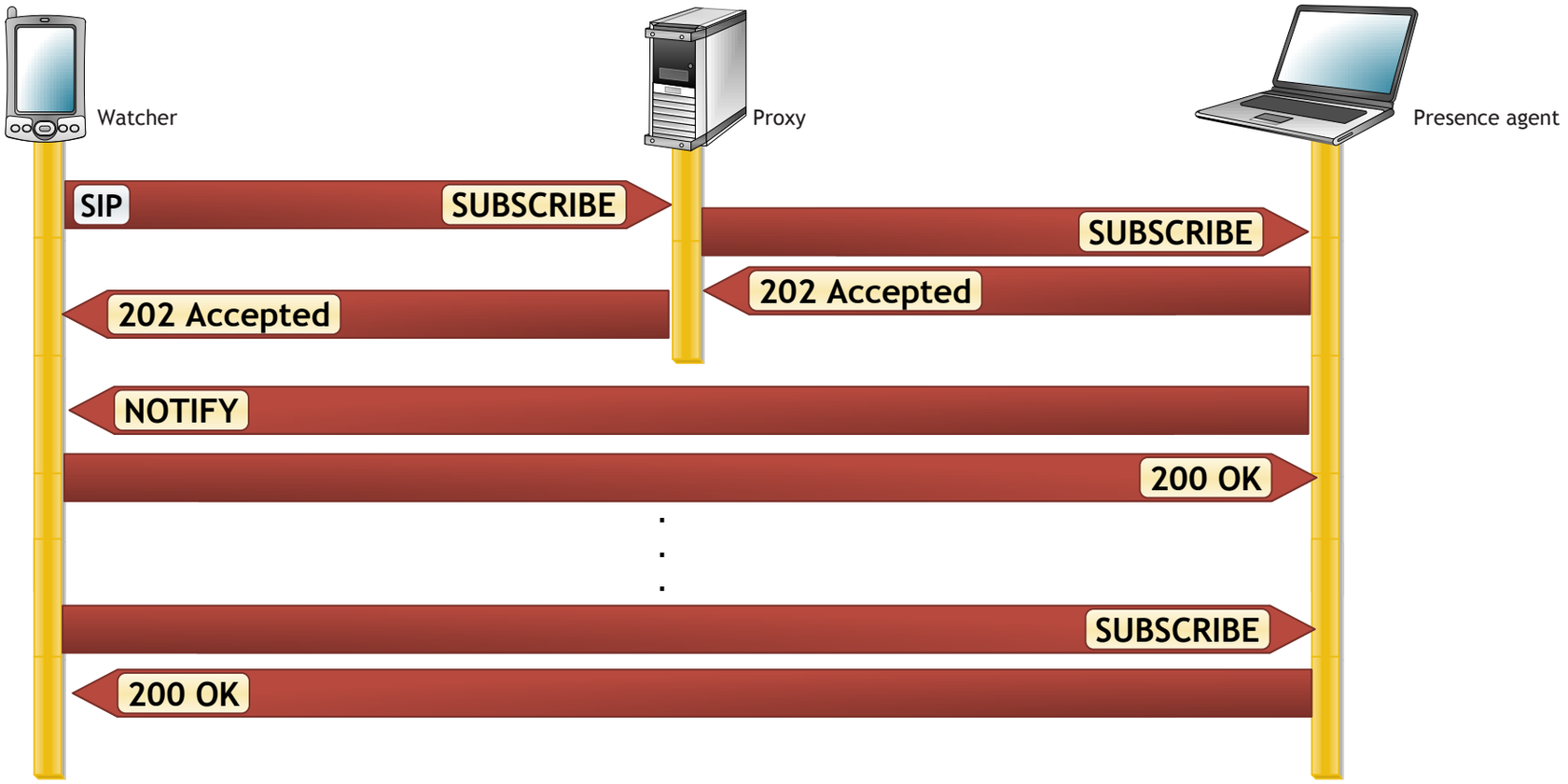
Use of REFER and Replaces to perform attended transfer feature



SUBSCRIBE

- used by a user agent to establish a subscription for the purpose of receiving notifications (via the NOTIFY method) about a particular event
- The subscription request contains an Expires header field, which indicates the desired duration of the existence of the subscription.
- There is no “UNSUBSCRIBE” method
- SUBSCRIBE with Expires:0 requests the termination of a subscription

SUBSCRIBE


Watcher
Proxy
Presence agent
SIP
SUBSCRIBE
SUBSCRIBE
202 Accepted
202 Accepted
NOTIFY
200 OK
SUBSCRIBE
200 OK

SUBSCRIBE

```
SUBSCRIBE sip:ptolemy@rosettastone.org SIP/2.0
Via SIP/2.0/UDP proxy.elasticity.co.uk:5060
;branch=z9hG4bK348471123
Via SIP/2.0/UDP parlour.elasticity.co.uk:5060
;branch=z9hG4bKABDA ;received=192.0.3.4
Max-Forwards: 69
To: <sip:Ptolemy@rosettastone.org>
From: Thomas Young <sip:tyoung@elasticity.co.uk>;tag=1814
Call-ID: 452k59252058dkfj349241k34
CSeq: 3412 SUBSCRIBE
Allow-Events: dialog
Contact: <sip:tyoung@parlour.elasticity.co.uk>
Event: dialog
Content-Length: 0
```

SUBCRIBE

- The type of event subscription is indicated by the required Event header field in the SUBSCRIBE request.
- Each application of the SIP Events framework [RFC 3265] defines a package with a unique event tag. Each package defines:
  - Default subscription expiration interval;
  - Expected SUBSCRIBE message bodies;
  - What events cause a NOTIFY to be sent, and what message body isexpected in the NOTIFY;
  - Whether the NOTIFY contains complete state or increments (deltas);
  - Maximum notification rate.

Event Packages and Template Packages

- Conference - Conference information including participant lists,policy information, and so forth]
- Dialog - Dialog state and identification informat
- Message-summary - Messages notification, used for message waiting indicator (mwi) with voicemail
- Presence - Presence information
- Refer - Refer state implicit subscription created by REFER
- Reg - User registration state
- Winfo - Watcher information template package

NOTIFY

- The NOTIFY method is used by a user agent to convey information about the occurrence of a particular event.
- A NOTIFY request normally receives a 200 OK response to indicate that it has been received
- NOTIFY requests contain an Event header field indicating the package and a Subscription-State header field indicating the current state of subscription

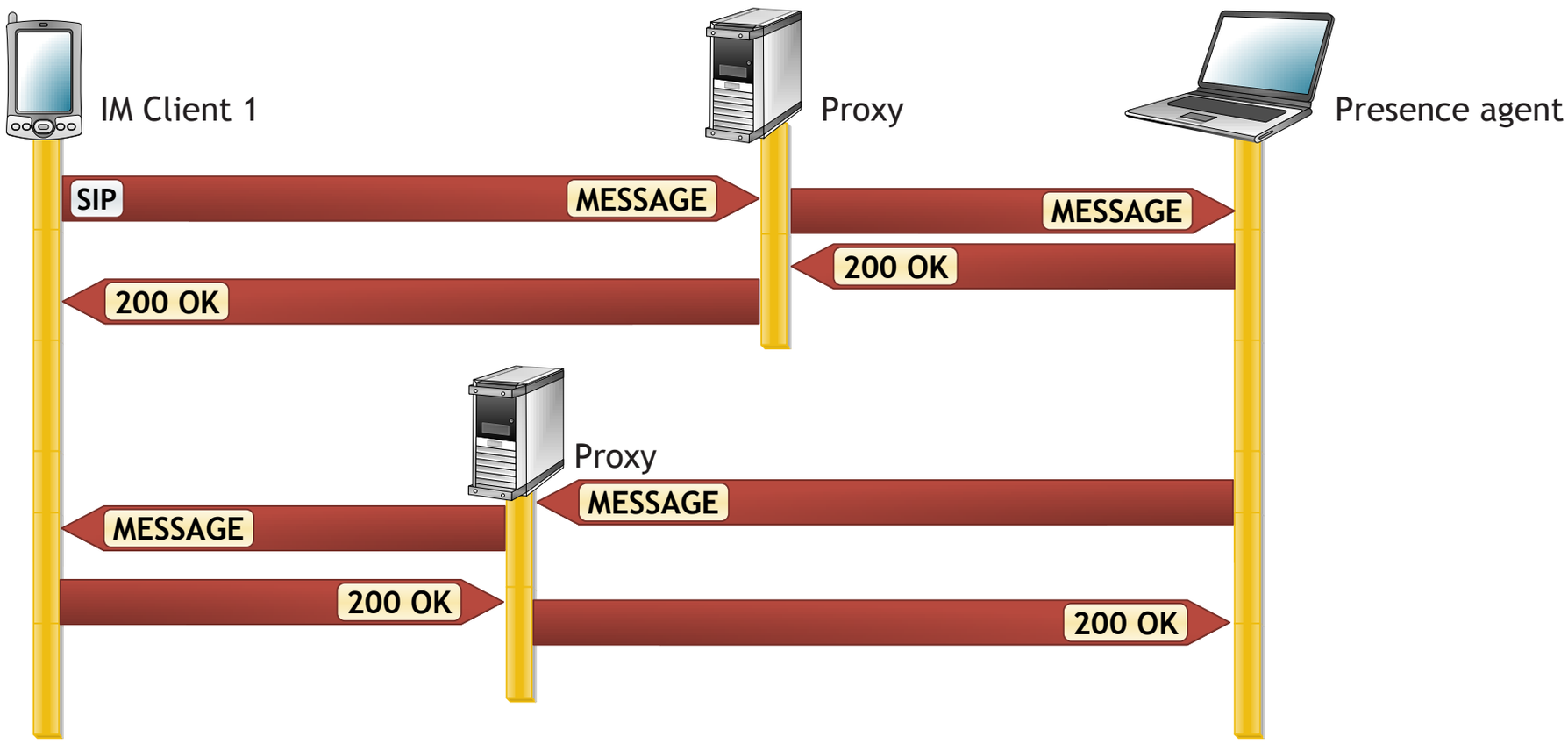
Mandatory Header Fields for a NOTIFY

- To
- From
- Call-ID
- CSeq
- Max-Forwards
- Via
- Contact
- Event
- Subscription-State
- Allow-Events

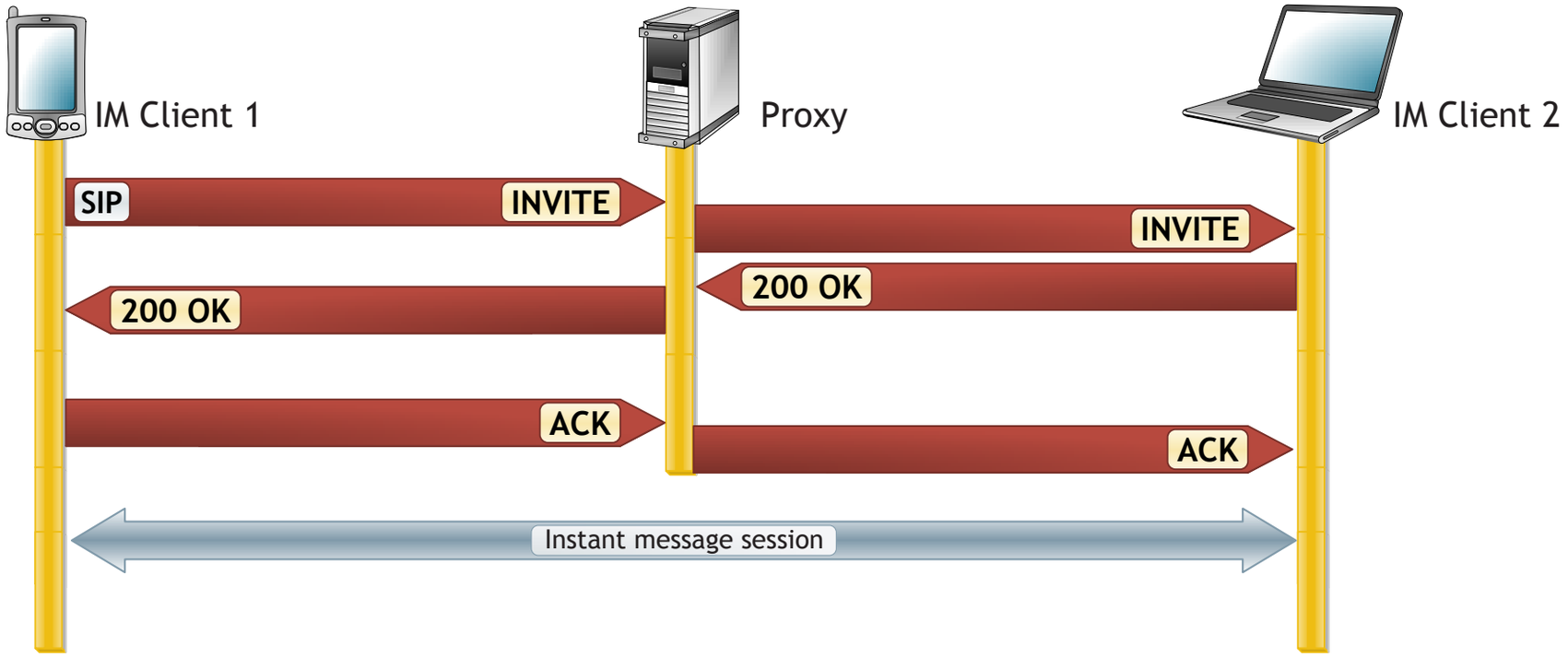
MESSAGE

- The MESSAGE method is used to transport instant messages (IM) using SIP
- All UAs that support the MESSAGE method must support plain/text
- UA may support other formats such as message/cpim or text/html or many others
- A MESSAGE request normally receives a 200 OK response to indicate that the message has been delivered to the final destination

SIP instant message example



Using SIP to establish an instant messaging session.



Message

```
MESSAGE sip:editor@rcs.org SIP/2.0
Via SIP/2.0/UDP
lab.mendeleev.org:5060;branch=z9hG4bK3
Max-Forwards: 70
To: <editor@rcs.org>
From: “D. I. Mendeleev”
<dmitry@mendeleev.org>;tag=1865
Call-ID: 9384719717204934 3
CSeq: 5634 MESSAGE
Subject: First Row
Contact: <sip:dmitry@lab.mendeleev.org>
Content-Type: text/plain
Content-Length: 6
Hello!
```

INFO

- The INFO method is used by a user agent to send call signaling information to another UA with which it has an established media session.
- Unlike re-INVITE, does not change the media characteristics of the call.
- INFO is end-to-end, and is never initiated by proxies.
- A proxy will always forward an INFO request
- It is up to the UAS to check to see if the dialog is valid.
- INFO requests for unknown dialogs receive a 481 Transaction/Dialog Does Not Exist response.

INFO

```
INFO sip:poynting@mason.edu.uk SIP/2.0
Via: SIP/2.0/UDP cavendish.kings.cambridge.edu.uk
;branch=z9hG4bK24555
Max-Forwards: 70
To: John Poynting <sip:nting@mason.edu.uk> ;tag=3432
From: J.C. Maxwell <sip:james.maxwell@kings.cambridge.edu.uk>
;tag=432485820183
Call-ID: 18437@cavendish.kings.cambridge.edu.uk
CSeq: 6 INFO
Content-Type: message/isup
Content-Length: 16
51a6324134527
```

PRACK

- The PRACK method is used to acknowledge receipt of reliably transported provisional responses (1xx).
- The reliability of 2xx, 3xx, 4xx, 5xx, and 6xx responses to INVITEs is achieved using the ACK method.
- The PRACK method applies to all provisional responses except the 100 Trying response, which is never reliably transported

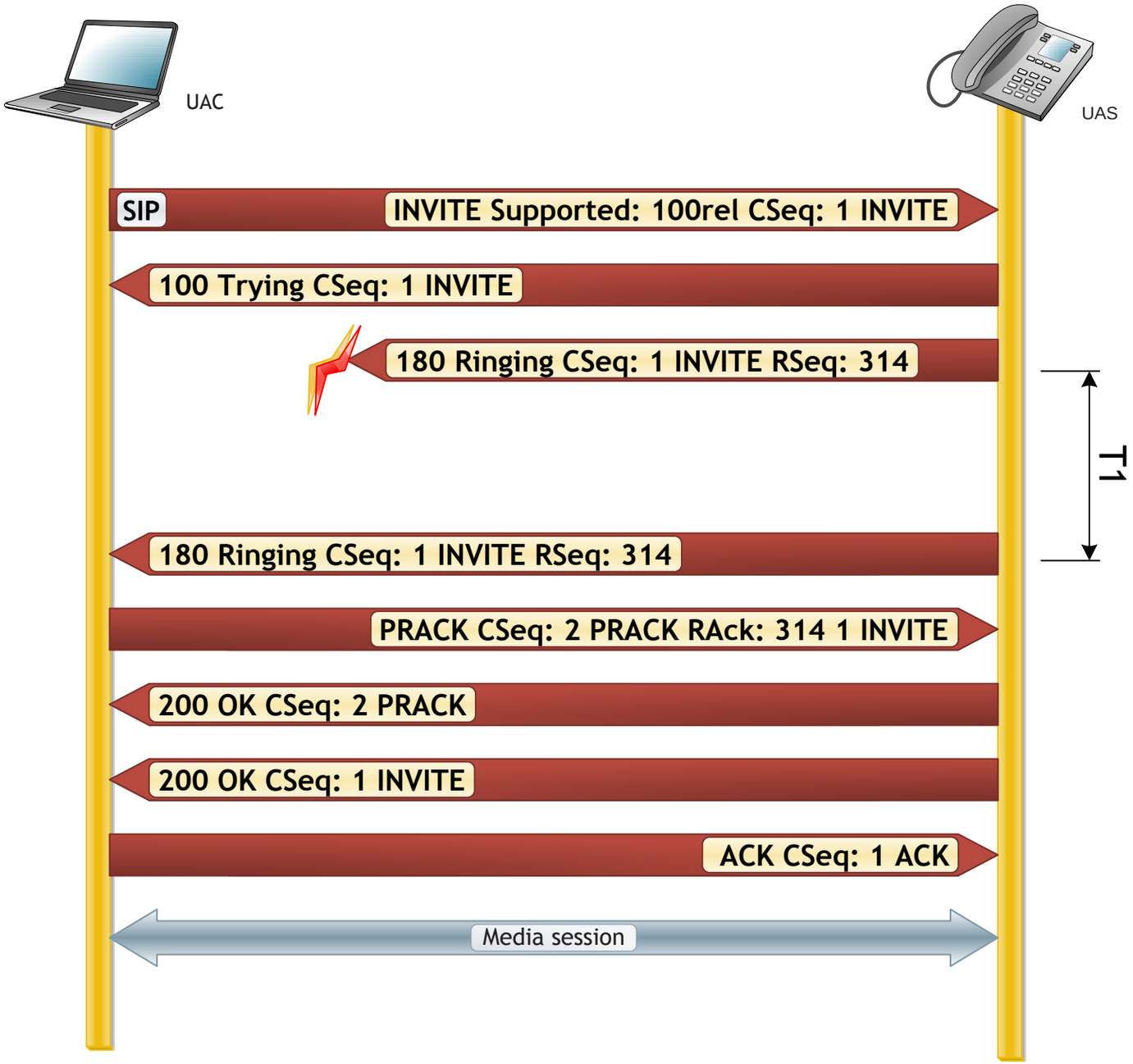
PRACK

- A PRACK is generated by a UAC when a provisional response has been received containing a RSeq reliable sequence number and a Supported: 100rel header
- The PRACK echoes the number in the Rseq and the CSeq of the response in a RAck header.

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP lucasian.trinity.cambridge.edu.uk
;branch=z9hG4bK452352
;received=1.2.3.4
To: Descartes
<sip:rene.descartes@metaphysics.org>;tag=12323
From: Newton
<sip:newton@kings.cambridge.edu.uk>;tag=981
Call-ID: 5@lucasian.trinity.cambridge.edu.uk
RSeq: 314
CSeq: 1 INVITE
Content-Length: 0
```

```
PRACK sip:rene.descartes@metaphysics.org SIP/2.0
Via: SIP/2.0/UDP lucasian.trinity.cambridge.edu.uk
;branch=z9hG4bKdtyw
Max-Forwards: 70
To: Descartes <sip:rene.descartes@metaphysics.org>;tag=12323
From: Newton <sip:newton@kings.cambridge.edu.uk>;tag=981
Call-ID: 5@lucasian.trinity.cambridge.edu.uk
CSeq: 2 PRACK
RAck: 314 1 INVITE
Content-Length: 0
SIP/2.0 200 OK
Via: SIP/2.0/UDP lucasian.trinity.cambridge.edu.uk;branch=z9hG4bKdtyw
;received=1.2.3.4
To: Descartes <sip:rene.descartes@metaphysics.org>;tag=12323
From: Newton <sip:newton@kings.cambridge.edu.uk>;tag=981
Call-ID: 5@lucasian.trinity.cambridge.edu.uk
CSeq: 2 PRACK
Content-Length: 0
```

Use of reliable provisional responses.



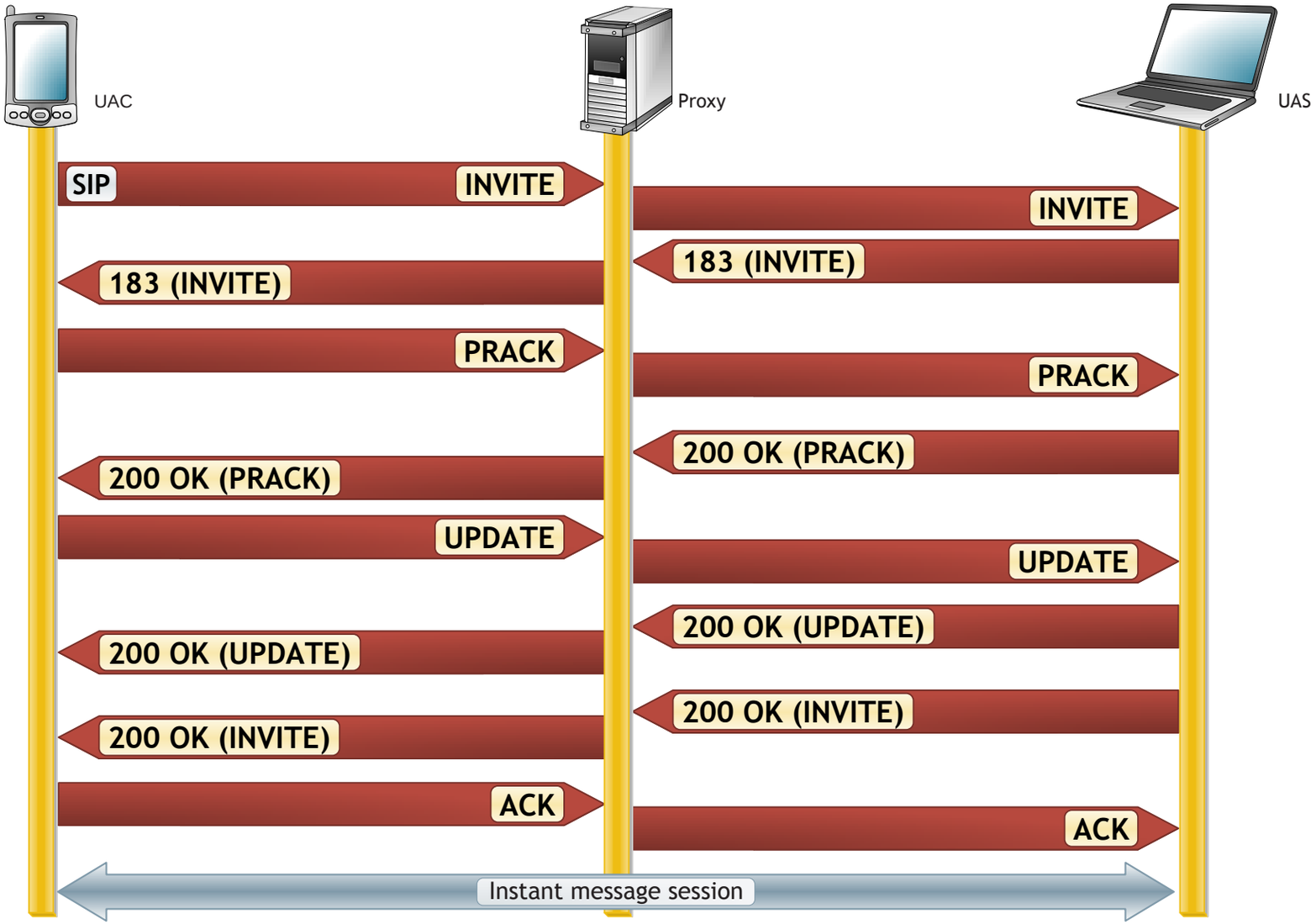
UPDATE

- The UPDATE method is used to modify the state of a session without changing the state of the dialog.

UPDATE

```
UPDATE sips:beale@bufords.bedford.va.us SIP/2.0
Via SIP/2.0/TLS client.crypto.org:5060;branch=z9hG4bK342
Max-Forwards: 70
To: T. Beale <sips:beale@bufords.bedford.va.us>;tag=71
From: Blaise Vigenere <sips:bvigenere@crypto.org>;tag=19438
Call-ID: 170189761183162948
CSeq: 94 UPDATE
Contact: <sips:client.crypto.org>
Content-Type: application/sdp
Content-Length: ...
(SDP Message body not shown...)
```

Example of UPDATE method



Tags

- A tag is a cryptographically random number with at least 32 bits of randomness,
- added to To and From headers to uniquely identify a dialog.
- To header in the initial INVITE will not contain a tag.
- A caller must include a tag in the from header
- Excluding 100 Trying, all responses will have a tag added to the To header.
- The sending or reception of a response containing a From tag creates an early dialog.
- A tag returned in a 200 OK response is then incorporated as a dialog identifier and used in all future requests for this Call-ID.
- A tag is never copied across calls.

Tags

- Any response generated by a proxy will have a tag added by the proxy.
- An ACK generated by either a user agent or a proxy will always copy the From tag of the response in the ACK request.
- If a UAC receives responses containing different tags, this means that the responses are from different UASs, and hence the INVITE has been forked.
- It is up to the UAC as to how to deal with this situation. For example, the UAC could establish separate sessions with each of the responding UAS. The dialogs would contain the same From, Call-ID, and CSeq, but would have different tags in the To header.
- The UAC also could BYE certain legs and establish only one session.
- Tags are not part of the To or From URI but are part of theheader and always placed outside any “<>”.

Message Bodies

- Message bodies in SIP may contain various types of information.
- The optional Content-Disposition header is used to indicate the intended use of the message body.
- If not present, the function is assumed to be session, which means that the body describes a media session.

Message Bodies

- The encoding scheme of the message body is indicated in the Content-Encoding header
- If not specified, the encoding is assumed to be text/plain
- The Content-Length header contains the number of octets in the message body. If there is no message body, the Content-Length header should still be included but has a value of 0
- Message bodies in SIP, should be small enough so that they do not exceed the UDP MTU

SIP response messages

Table of Responses (SIP)

- 1xx – Provisional: request received, continuing to process the request

- 2xx – Success: the action was successfully received, understood and accepted

- 3xx – Redirection: further action needs to be taken in order to complete the request
- 4xx - Client Error: the request contains bad syntax or cannot be fulfilled at this server
- 5xx - Server Error: the server failed to fulfill an apparently valid request
- 6xx - Global Failure: the request cannot be fulfilled at any server

non-2xx final responses

final responses

SIP responses

Informational (1xx)

- The informational class of responses 1xx are used to indicate call progress.
- Informational responses are end-to-end responses and may contain message bodies.
- 100 Trying response, is only ahop-by-hop response and may not contain a message body.
- Any number of informational responses can be sent by a UAS prior to a final response (2xx,3xx, 4xx, 5xx, or 6xx class response) being sent.
- The first informational response received by the UAC confirms receipt of the INVITE, and stops retransmission of the INVITE.

Informational (1xx)

- Servers returning 100 Trying responses minimizes INVITE retransmissions in the network.
- Further informational responses have no effect on INVITE retransmissions.
- A stateful proxy receiving a retransmission of an INVITE will resend the last provisional response sent to date.
- Informational responses are optional
- UAS can send a final response without first sending an informational response.
- Final responses to an INVITE receive an ACK to confirm receipt, provisional responses are not acknowledged, except using the PRACKmethod.

- 100 Trying - special case response is only a hopby-hop request.
  - It is never forwarded
  - may not contain a message body.
  - A forking proxy must send a 100 Trying response, since the extended search being performed may take a significant amount of time.
  - This response can be generated by either a proxy server or a UA.
  - It only indicates that some kind of action is being taken to process the call
  - It does not indicate, that the user has been located.
  - A 100 Trying response typically does not contain a To tag.

- 180 Ringing - This response is used to indicate that the INVITE has been received by the user agent and that alerting is taken place
  - This response is important in interworking with telephony protocols
  - Typically mapped to messages such as an ISDN Progress or ISUP Address Complete Message (ACM).
  - When the user agent answers immediately, a 200 OK is sent without a 180 Ringing; the scenario is called the “fast answer” case in telephony.
  - A message body in this response could be used to carry QoS or security information, or to convey ring tone or animations from the UAS to the UAC.
  - A UA normally generates its own ringback tone or remote ringing indication, unless a Alert-Info header field is present.

- 181 Call Is Being Forwarded - used to indicate that the call has been handed off to another end-point.
  - Is sent when this information may be of use to the caller.
  - Also, because a forwarding operation may take longer for the call to be answered, this response gives a status for the caller.
- 182 Call Queued - used to indicate that the INVITE has been received, and will be processed in a queue.
  - The reason phrase can be used to indicate the estimated wait time or the number of callers in line.
  - Body in this response can be used to carry music on hold or other media.

- 183 Session Progress - indicates that information about the progress of the session (call state) may be present in a message body or media stream
  - end-to-end response
  - does establish a dialog
  - A typical use of this response is to allow a UACto hear ring tone, busy tone, or a recorded announcement in calls through a gateway into the PSTN.

Success (2xx)

- 200 OK - When used to accept a session invitation, it will contain a message body containing the media properties of the UAS (called party).
  - When used in response to other requests, it indicates successful completion or receipt of the request.
  - The response stops further retransmissions of the request.
  - In response to an OPTIONS, the message body may contain the capabilities of the server.
  - A message body may also be present in a response to a REGISTER request.
  - For 200 OK responses to CANCEL, INFO, MESSAGE, SUBSCRIBE, NOTIFY, and PRACK, a message body is not permitted.
- 202 Accepted - indicates that the UAS has received and understood the request, but that the request may not have been authorized or processed by the server. It is commonly used in responses to SUBSCRIBE and REFER , and sometimes MESSAGE methods.

Redirection

- Redirection class responses are generally sent by a SIP server acting as a redirect server in response to an INVITE
- There is no requirement that a UAC receiving a redirection response must retry the request to the specified address.

- 300 Multiple Choices
  - contains multiple Contact header fields, which indicate that the location service has returned multiple possible locations for the sip or sips URI in the Request-URI.
  - The order of the Contact header fields is assumed to be significant. That is, they should be tried in the order in which they were listed in the response.
- 301 Moved Permanently
  - Contact header field with the new permanent URI of the called party. The address can be saved and used in future INVITE requests.

- 302 Moved Temporarily -
  - Contains a URI that is currently valid but that is not permanent. As a result, the Contact header field should not be cached across calls unless an Expires header field is present, in which case the location is valid for the duration of the time specified.
- 305 Use Proxy contains a URI that points to a proxy server who has authoritative information about the calling party.
  - The caller should resend the request to the proxy for forwarding. This response could be sent by a UAS that is using a proxy for incoming call screening.
- 380 Alternative Service - returns a URI that indicates the type of service that the called party would like. An example might be a redirect to a voicemail server.

4xx Client error

- 400 Bad Request - indicates that the request was not understood by the server.
  - missing required header fields such as To,From, Call-ID, or CSeq.
  - also used if a UAS receives multiple INVITE requests (not retransmissions) for the same Call-ID.
- 401 Unauthorized - indicates that the request requires the user to perform authentication. generally sent by a user agent, since the 407 Proxy Authentication Required is sent by a proxy
  - The exception is a registrar server, which sends a 401 Unauthorized response to a REGISTER message that does not contain the proper credentials.

- 402 Payment Required - placeholder for future definition in the SIP protocol. It could be used to negotiate call completion charges.
- 403 Forbidden - used to deny a request without giving the caller any recourse.
  - sent when the server has understood the request, found the request to be correctly formulated, but will not service the request.
  - not used when authorization is required.
- 404 Not Found - indicates that the user identified by the sip or sips URI in the Request-URI cannot be located by the server, or that the user is not currentlysigned on with the user agent.
- 405 Method Not Allowed - indicates that the server or user agent has received and understood a request but is not willing to fulfill the request.
  - An example might be a REGISTER request sent to a user agent.
  - An Allow header field must be present to inform the UAC what methods are acceptable.
  - Different from the case of an unknown method, in which a 501 Not Implemented response is returned.

- 406 Not Acceptable - Indicates that the request cannot be processed due to a requirement in the request message.
  - The Accept header field in the request did not contain any options supported by the UAS.
- 407 Proxy Authentication Required sent by a proxy indicates that the UAC must first authenticate itself.
  - The response should contain information about the type of credentials required by the proxy in a Proxy Authenticate header field.
  - 407 may not be used by a proxy to authenticate another proxy.

- 408 Request Timeout sent when an Expires header field is present in an INVITE request, and the specified time period has passed. Alternatively could be sent if transaction timer times out, even with no Expires header in the request.
- 409 Conflict - removed from RFC 3261 but is defined in RFC 2543.
- 410 Gone - similar to the 404 Not Found response but contains the hint that the requested user will not be available at this location in the future. (eg. cancelled service)

- 413 Request Entity Too Large - can be used by a proxy to reject a request that has a message body that is too large.
  - A proxy suffering congestion could temporarily generate this response to save processing long requests.
- 414 Request-URI Too Long - Request-URI in the request was too long and cannot be processed correctly. (There is no maximum length defined for a Request-URI in the SIPstandard document. :-D)

- 415 Unsupported Media Type - Indicates that the media type contained in the INVITE request is not supported.
  - For example, a request for a video conference to a PSTN gateway that only handles telephone calls will result in this response.
  - The response should/can contain header fields to help the UAC reformulate the request.
  - Is generated by UA exlusively
- 416 Unsupported URI Scheme - new to RFC 3261
  - used when a UAC uses a URI scheme in a Request-URI that the UAS does not understand.
  - If a request URI contains a secure SIP (sips) scheme that a proxy does not understand, it would return a 416.
  - Since all SIP elements must understand the sip scheme, the request should be retried using a sip uri in the request-uri.

- 420 Bad Extension- indicates that the extension specified in the Require header field is not supported by the proxy or user agent.
  - The response should contain a Supported header field listing the extensions that are supported. The UAC could resubmit the same request without the extension in the Require header field or submit the request to another proxy or user agent
- 421 Extension Required- indicates that a server requires an extension to process the request that was not present in a Supported header field in the request.
  - The required extension should be listed in a Required header field in the response.

- 422 Session Timer Interval Too Small - used to reject a request containing a Session-Expires header field with too short an interval. The ability to reject short durations
- 423 Interval Too Brief returned by a registrar that is rejecting a registration request because the requested expiration time on one or more Contacts is too brief.
  - The response must contain a Min-Expires header
- 428 Use Authentication Token - used by UAS that is requiring the use of an Authentication Information Body (AIB)
  - The AIB is a S/MIME body that is an encrypted message/sip or message/sipfrag body.

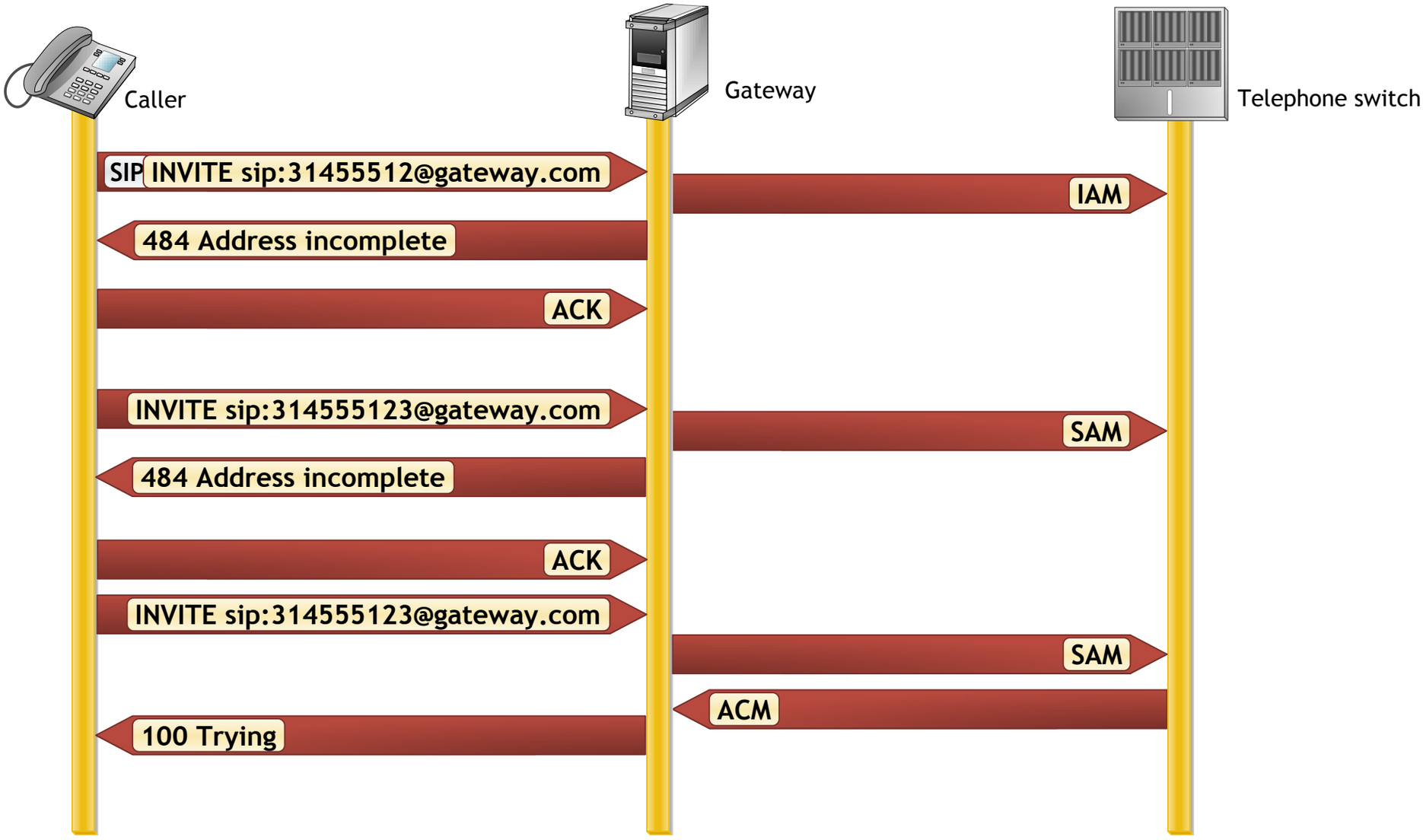
- The 429 Provide Referror Identity - used to request that a Referred-By header field be resent with a valid Referred-By security token. The security token is carried as an S/MIME message body.
  - recipient of this error message (the UA that received and accepted the REFER) should relay this request back to the originator of the REFER by including it in a NOTIFY.
  - The sender of the REFER can then generate the Referred-By security token and include it in the REFER, which would then be copied into the triggered request.

- 480 Temporarily Unavailable - indicates that the request has reached the correct destination, but the called party is not available for some reason.
  - The reason phrase should be modified for this response to give the caller a better understanding of the situation.
  - The response should contain a Retry-After header indicating when the request may be able to be fulfilled.
  - could be sent when a telephone has its ringer turned off, or a “do not disturb” button has been pressed.
  - This response can also be sent by a redirect server.

- 481 Dialog/Transaction Does Not Exist - response referencing an existing call or transaction has been received for which the server has no records or state information.
- 482 Loop Detected - the request has been looped and has been routed back to a proxy that previously forwarded the request.
  - Each server that forwards a request adds a Via header with its address to the top of the request.
  - A branch parameter is added to the Via header, which is a hash function of the Request-URI, and the To, From, Call-ID, and CSeq number.
  - A second part is added to the branch parameter if the request is being forked.
  - Branch parameter must be checked is to allow a request to be routed back to a proxy, provided that the Request-URI has changed. This could happen with a call forwarding feature. In this case, the Via headers would differ by having different branch parameters.

- 483 Too Many Hops indicates that the request has been forwarded the maximum number of times as set by the Max-Forwards header.
- 484 Address Incomplete - indicates that the Request-URI address is not complete.
  - This could be used in an overlap dialing scenario in PSTN interworking where digits are collected and sent until the complete telephone number is assembled by a gateway and routed
  - Note that the follow-up INVITE requests may use the same Call-ID as the original request.

Overlap dialing



- 485 Ambiguous - indicates that the Request-URI was ambiguous and must be clarified in order to be processed.
  - This occurs if the username matches a number of registrations.
  - If the possible matching choices are returned in Contact header fields, then this response is similar to the 300 Multiple Choices response.
- 486 Busy Here - This response is used to indicate that the user agent cannot accept the call at this location.
  - Different, however, from the 600 Busy Everywhere response, which indicates that the request should not be tried elsewhere.
  - Equivalent to the busy tone in the PSTN.

- 487 Request Terminated - can be sent by a user agent that has received a CANCEL request for a pending INVITE request.
  - A 200 OK is sent to acknowledge the CANCEL
  - 487 is sent in response to the INVITE.
- 488 Not Acceptable Here indicates that some aspect of the proposed session is not acceptable.
  - May contain a Warning header field indicating the exact reason.
  - Has a similar meaning to 606 Not Acceptable, but only applies to one location and may not be true globally as the 606 response.
- 489 Bad Event - used to reject a subscription request or notification containing an Event package that is unknown or not supported by the UAS.
  - also used to reject a subscription request that does not specify an Event package, assuming that the server does not support the PINT protocol.

- 491 Request Pending - used to resolve accidental simultaneous re-INVITEs by both parties in a dialog.
  - Since both INVITEs seek to change the state of the session, they cannot be processed at the same time.
  - While a user agent is awaiting a final response to a re-INVITE, any re-INVITE request received must be replied to with this response code.
  - This is analogous to the “glare” condition in telephony in which both ends seize a trunk at the same time. The reconsideration algorithm in SIP is for the user agent to generate a delay (randomly selected within a range determined by if the user agent send the initial INVITE or not) then retry the re-INVITE, assuming that another re-INVITE has not been received in the meantime.
  - One side or the other will “win” the race condition and have the re-INVITEprocessed.

- 493 Request Undecipherable - used when an S/MIME message body can not be decrypted because the public key is unavailable.
  - If the UAS does not support S/MIME, no message body will be present in the response. If the UAS does support S/MIME, the response will contain a message body containing a public key suitable for the UAC to use for S/MIME encryption.

5xx Server error

- Indicate that the request cannot be processed because of an error with the server.
- The response may contain a Retry After header field if the server anticipates being available within a specific time period.
- The request can be tried at other locations because there are no errors indicated in the request.

- 500 Server Internal Error - server has experienced some kind of error that is preventing it from processing the request.
  - The reason phrase can be used to identify the type of failure.
  - The client can retry the request again at this server after several seconds.
- 501 Not Implemented - indicates that the server is unable to process the request because it is not supported.
  - Can be used to decline a request containing an unknown method.
  - A proxy, however, will forward a request containing an unknown request method. Thus, a proxy will forward an unknown SELF-DESTRUCT request, assuming that the UAS will generate this response if the method is not known.
- 502 Bad Gateway - sent by a proxy that is acting as a gateway to another network, and indicates that some problem in the other network is preventing the request from being processed.
- 503 Service Unavailable This response indicates that the requested service is temporarily unavailable.
  - The request can be retried after a few seconds, or after the expiration of the Retry-After header field. Instead of generating this response, a loaded server may just refuse the connection.
  - This response code is important in that its receipt triggers a new DNS lookup to locate a backup server to obtain the desired service.

6xx Global Error

- Indicates that the server knows that the request will fail wherever it is tried.
- As a result, the request should not be sent to other locations.
- Only a server that has definitive knowledge of the user identified by the Request-URI in every possible instance should send a global error class response.
- Otherwise, a client error class response should be sent. A Retry-Afterheader field can be used to indicate when the request might be successful.

- 603 Decline same effect as the 600 Busy Everywhere, but does not give away any information about the call state of the server.
  - Could indicate the called party is busy, or simply does not want to accept the call.
- 604 Does Not Exist Anywhere - Similar to the 404 Not Found response but indicates that the user in the Request-URI cannot be found anywhere.
  - This response should only be sent by a server that has access to all informationabout the user.
- 606 Not Acceptable - can be used to implement some session negotiation capability in SIP.
  - This response indicates that some aspect of the desired session is not accept able to the UAS, and as a result, the session cannot be established.
  - The response may contain a Warning header field with a numerical code describing exactly what was not acceptable.
  - The request can be retried with different media session information.

SIP entities

- User Agents:

- Dedicated VoIP phone
  - below the surface these dedicated VoIP-phones are nothing more but regular computers with a simple operating system and a SIP-softphone

- Mobile Stations
  - Mobile stations with an integrated SIP-client will be the typical user devices of tomorrow's 3GPP-networks.
  - 3GPP bases its entire IMS session control on SIP.

- Softphones

  - The best known and most widespread SIP user agent is certainly the softphone.
  - These softphones are nowadays available for literally every operating system and every platform.
  - This includes softphones running on regular Pcs, laptops, PDAs or smartphones

- Set Top Boxes

  - Set top boxes are no telephones but they represent the other end of SIP user agents, dedicated for gaming and VoD or audio on demand.

Stateless SIP-Proxy Server

- Unlike stateful proxies, stateless proxies do not maintain or observe the state of a SIP-transaction which is routed through them.
  - No UAC or UAS functions into the stateless proxy.
  - Stateless proxies will NOT retransmit SIP-messages.
  - Inspect the content of SIP-messages and may add header fields autonomously.
  - Not allowed to autonomously generate SIP-Requests (just like statefull).
  - In contrast to stateful SIP-proxies, the stateless SIP-proxy cannot generate CANCEL-Requests.
  - Cannot redirect a Request: INVITE-message to a new direction if receive a redirection response (<--> Response: 3XX) from a redirect server.
  - Cannot be used for forking.

Stateful SIP-Proxy Server

- SIP-proxy is a device which is addressable by a SIP-User Agent or by another SIP-proxy server through a SIP-URI.
  - SIP-proxies will relay SIP-messages somewhat closer to their final destination. However, with one exception a SIP-proxy server is not allowed to generate SIP-requests autonomously.
  - The exception are Request: CANCEL-messages which need to be generated by the proxy server e.g. after a called SIP-device has been ringing for some time and now the call shall be forked to the next possible device.
  - Stateful SIP-proxy servers maintain and observe the state of every transaction which is routed through them.
  - Note that they do not necessarily maintain dialog or call state, this is the domain of B2BUA's.
  - Only stateful proxies can be used as redirect server or as registrar.
  - Only stateful proxies can be used for forking.

SBC (Session Border Controller), B2BUA (Back-to-Back User Agent)

- "Session Border Controller" or "SBC" have no representation in IETF.
- In practice, SBC's represent the combination of
  - B2BUA's (which actually have been defined in RFC 3261)
  - and media gateway like equipment that allows:
    - media stream observation
    - and even modification (e.g. change of codec type).
- Most importantly, B2BUA's represent SIP servers that act like user agents.
- B2BUA's can autonomously generate SIP-Requests
- can autonomously terminate a session (Request: BYE) which is something that a SIP-proxy cannot do.
- When B2BUA's are also used for media transversal then they become SBC's.

Proxy

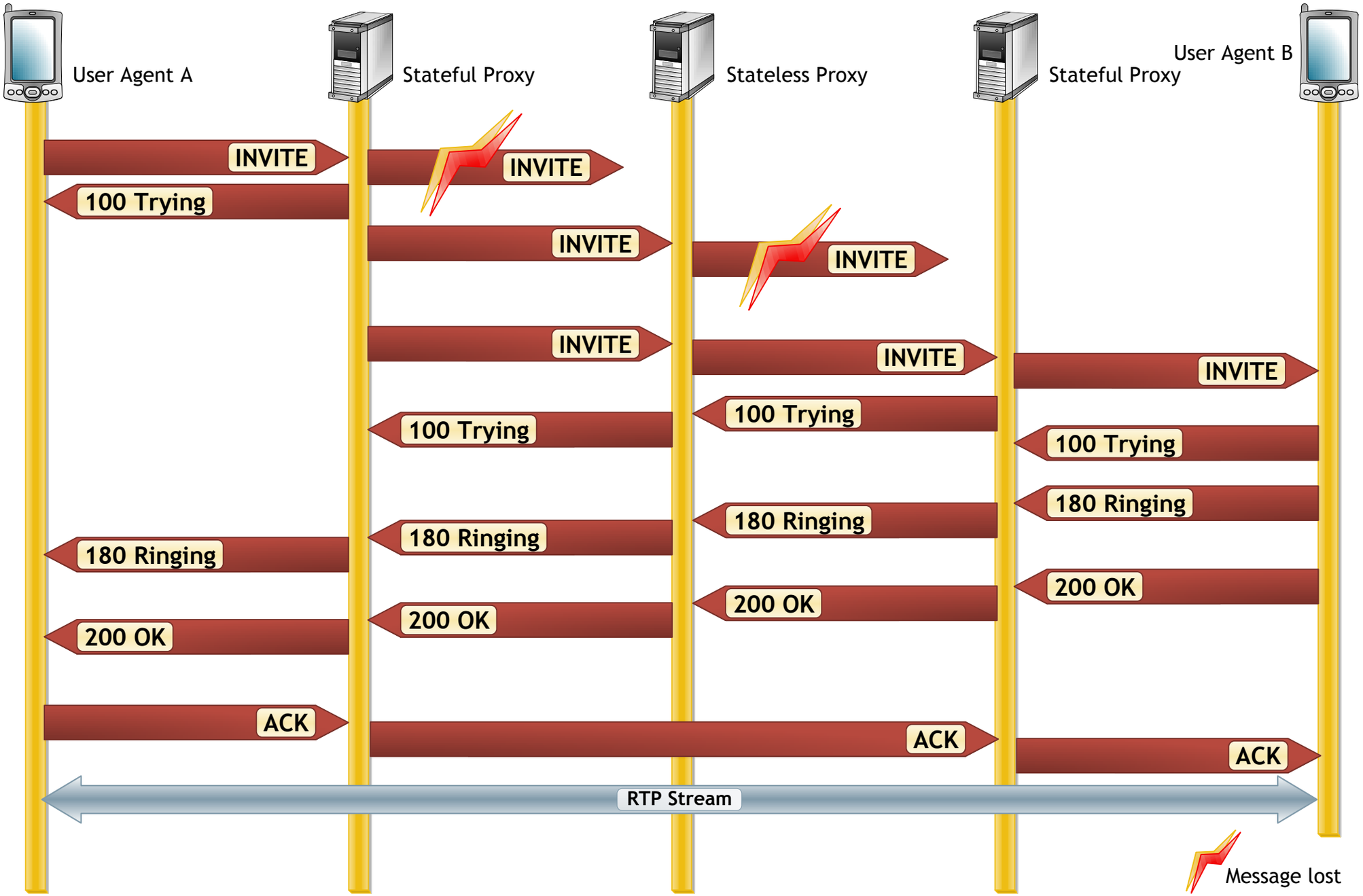
- Stateful Proxy

  A logical entity that maintains the client and server transaction state machines defined by this specification during the processing of a request, also known as a transaction stateful proxy.

- Stateless Proxy

  A logical entity that does not maintain the client or server transaction state machines defined in this specification when it processes requests. A stateless proxy forwards every request it receives downstream and every response it receives upstream.

Stateful vs. Stateless Proxy



Proxy

- Call Stateful Proxy

  A proxy is call stateful if it retains state for a dialog from the initiating INVITE to the terminating BYE request. A call stateful proxy is always transaction stateful, but the converse is not necessarily true.

- Transaction Stateful Proxy

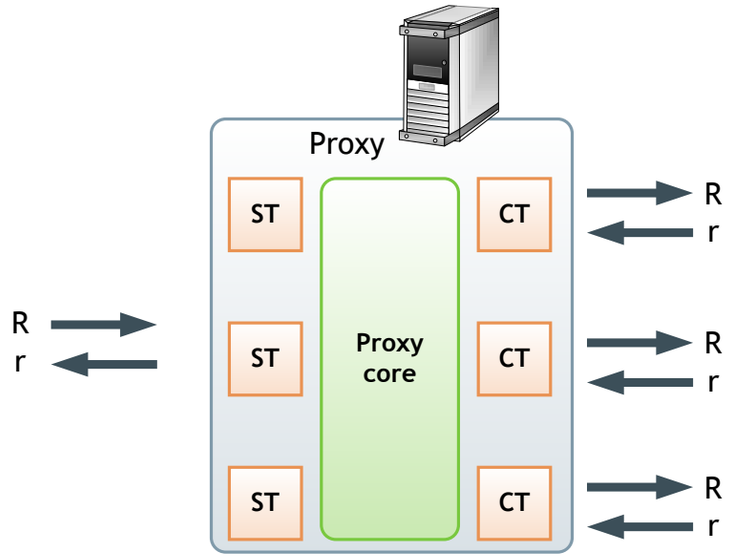
  The same as the stateful proxy

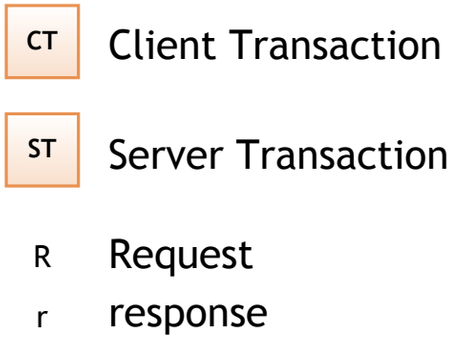
Proxy

An intermediary entity that acts as both a server and a client for the purpose of making requests on behalf of other clients.

- Plays role of routing.
- Interprets, and, if necessary, rewrites specific parts of a request message before forwarding it.
- Stateful vs. Stateless

(Note: stateless does not contain CT or ST)





Basic Steps for requests

- The proxy starts with a copy of the received request.
- The copy MUST contain all of the header fields from the received request (MUST NOT reorder field values with a common field name).
- The proxy MUST NOT add to, modify, or remove the message body.

For all new requests, including any with unknown methods, an element intending to proxy the request MUST:

1. Validate the request
2. Preprocess routing information
3. Determine target(s) for the request
4. Forward the request to each target
5. Process all responses

Requests Validation (1)

A valid message must pass the following checks:

a) Reasonable syntax - unknown method must not be refused
b) URI scheme - when not understood -> 416 (Unsupported URI Scheme)
c) Max-Forwards - if value of zero -> 483 (Too many hops)
d) (Optional) Loop Detection - branch in Via header checked
e) Proxy-Require
f) Proxy-Authorization

If any of these checks fail, the element MUST behave as a user agent server and respond with an error code.

Loop

A request that arrives at a proxy, is forwarded, and later arrives back at the same proxy. When it arrives the second time, its Request-URI is identical to the first time, and other header fields that affect proxy operation are unchanged, so that the proxy would make the same processing decision on the request it made the first time. Looped requests are errors, and the procedures for detecting them and handling them are described by the protocol.

Spiral

A spiral is a SIP request that is routed to a proxy, forwarded onwards, and arrives once again at that proxy, but this time differs in a way that will result in a different processing decision than the original request. Typically, this means that the request's Request-URI differs from its previous arrival. A spiral is not an error condition, unlike a loop.

A typical cause for this is call forwarding. A user calls joe@example.com. The example.com proxy forwards it to Joe's PC, which in turn, forwards it to bob@example.com. This request is proxied back to the example.com proxy. However, this is not a loop. Since the request is targeted at a different user, it is considered a spiral, and is a valid condition.

Detection Loops

Proxy uses a globally unique branch parameter for detection loops. It is performed by verifying that, when a request returns to a proxy, those fields having an impact on the processing of the request have not changed.

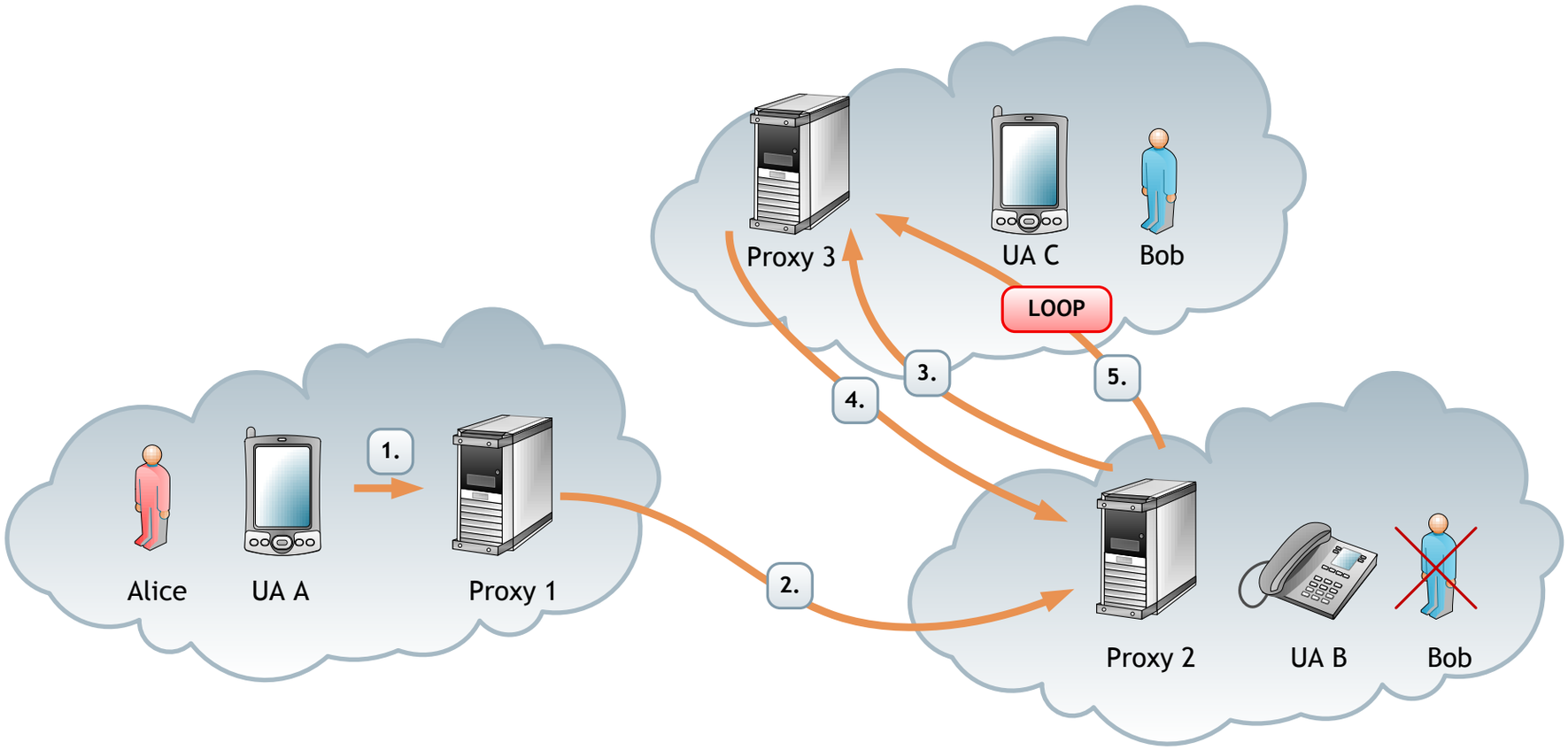
The common way to create the branch value is to compute a cryptographic hash of:

To and From tag, Call-ID header field, Request-URI, the topmost Via header, sequence number from the CSeq header field, in addition to any Proxy-Require and Proxy-Authorization header fields.

Note:

- The request method MUST NOT be included in the calculation of branch parameter.
- Loop detection can be substituted by Max-Forwards header.
- Loop detection is optional.

Detection Loops - A Call is Redirected



When a loop is detected 482 Loop Detected response is sent.

Route Information Preprocessing (2)

The proxy must inspect the Request-URI of the request.

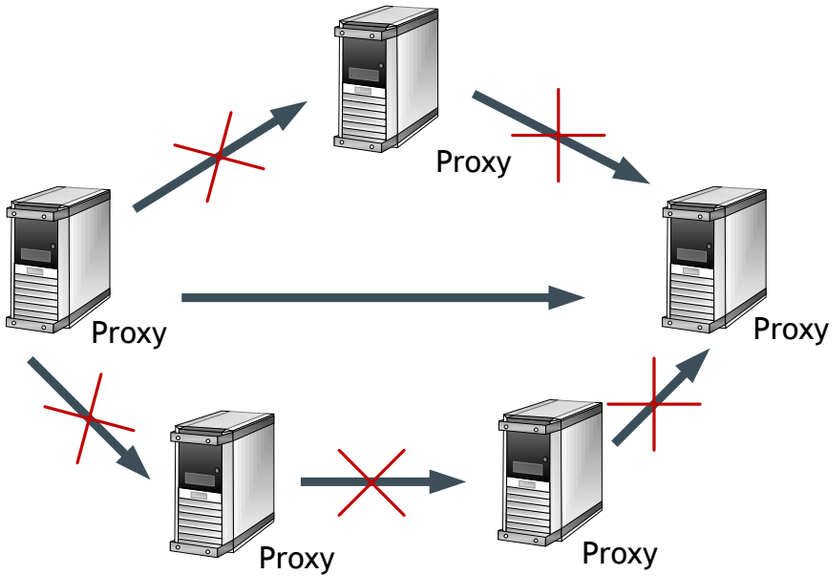
If the Request-URI of the request contains a value this proxy previously placed into a Record-Route header field, the proxy must replace the Request-URI in the request with the last value from the Route header field.

If the first value in the Route header field indicates this proxy, the proxy MUST remove that value from the request.

Note: Proxy MUST check if has been used strict or loose routing.

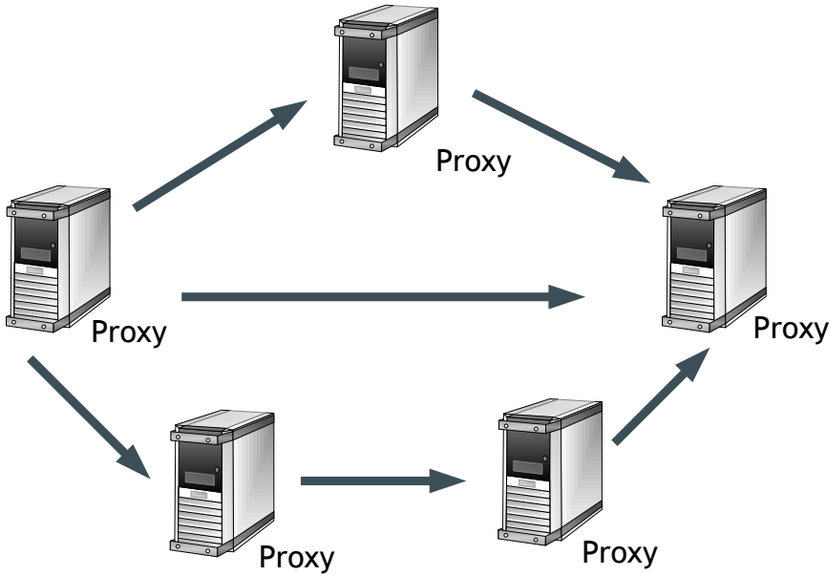
Strict Routing

A proxy is said to be strict routing if it follows the Route processing rules of RFC 2543 and many prior work in progress versions of RFC 3261. That rule caused proxies to destroy the contents of the Request-URI when a Route header field was present. Strict routing behavior is not used in RFC 3261, in favor of a loose routing behavior. Proxies that perform strict routing are also known as strict routers.



Loose Routing

A proxy is said to be loose routing if it follows the procedures which separate the destination of the request (present in the Request-URI) from the set of proxies that need to be visited along the way (present in the Route header field). A proxy compliant to these mechanisms is also known as a loose router.



Determining Request Targets (3)

The set of targets will either be predetermined by the contents of the request or will be obtained from an abstract location service. Each target in the set is represented as a URI.

On the construction of the target set has influence many conditions:

- Contents or the presence of header fields and bodies
- The time of day of the request's arrival
- Failure of previous requests, etc.

Request Forwarding (4)

As soon as the target set is non-empty, a proxy MAY begin forwarding the request.

Processing the set by stateful proxy:

- Serial – allowing each client transaction to complete before starting the next
- Parallel
- Dividing the set into groups – processing the groups serially and processing the targets in each group in parallel

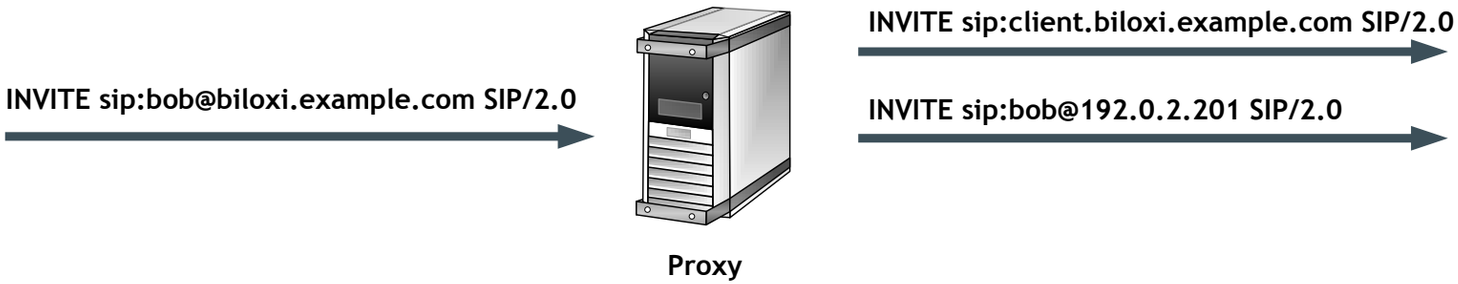
A common ordering mechanism is to use the qvalue parameter of targets obtained from Contact header.

Steps of Proxy Request Forwarding #1

a) Update the Request-URI
b) Update the Max-Forwards header field
c) Optionally add a Record-Route header field value
d) Optionally add additional header fields
e) Postprocess routing information
f) Determine the next-hop address, port, and transport
g) Add a Via header field value
h) Add a Content-Length header field if necessary
i) Forward the new request
j) Set timer C

Steps of Proxy Request Forwarding #2

Request-URI: MUST be replaced with the URI for the target, if known.



Max-Forwards: the proxy MUST decrement its value by one. If the copy does not contain Max-Forwards header field, the proxy MUST add one with a field value, which SHOULD be 70.
Note: Max-Forwards header is used only in requests.

Steps of Proxy Request Forwarding #3

Record-Route: If the proxy wishes to remain on the path of future requests in a dialog created by the request (assuming the request creates dialog), it MUST insert a Record-Route header field value into the copy before any existing Record-Route values.

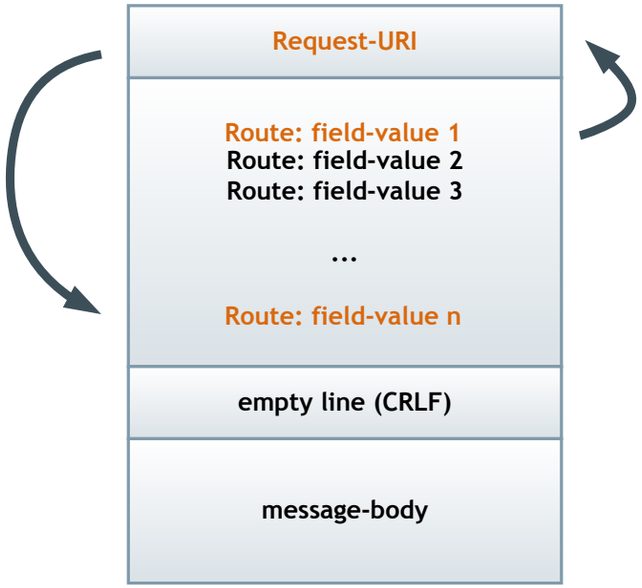
- Request establishing a dialog may contain a preloaded Route header field (e.g. from registration process).
- The Record-Route process is designed to work for any SIP request that initiates a dialog but a proxy MAY insert a
  Record-Route header filed into any request. If the request
  does not initiate a dialog, the endpoints will ignore the value.
- The URI in the Record-Route MUST contain an lr (loose routing) parameter.

Add Additional Header Fields: the proxy MAY add any appropriate.

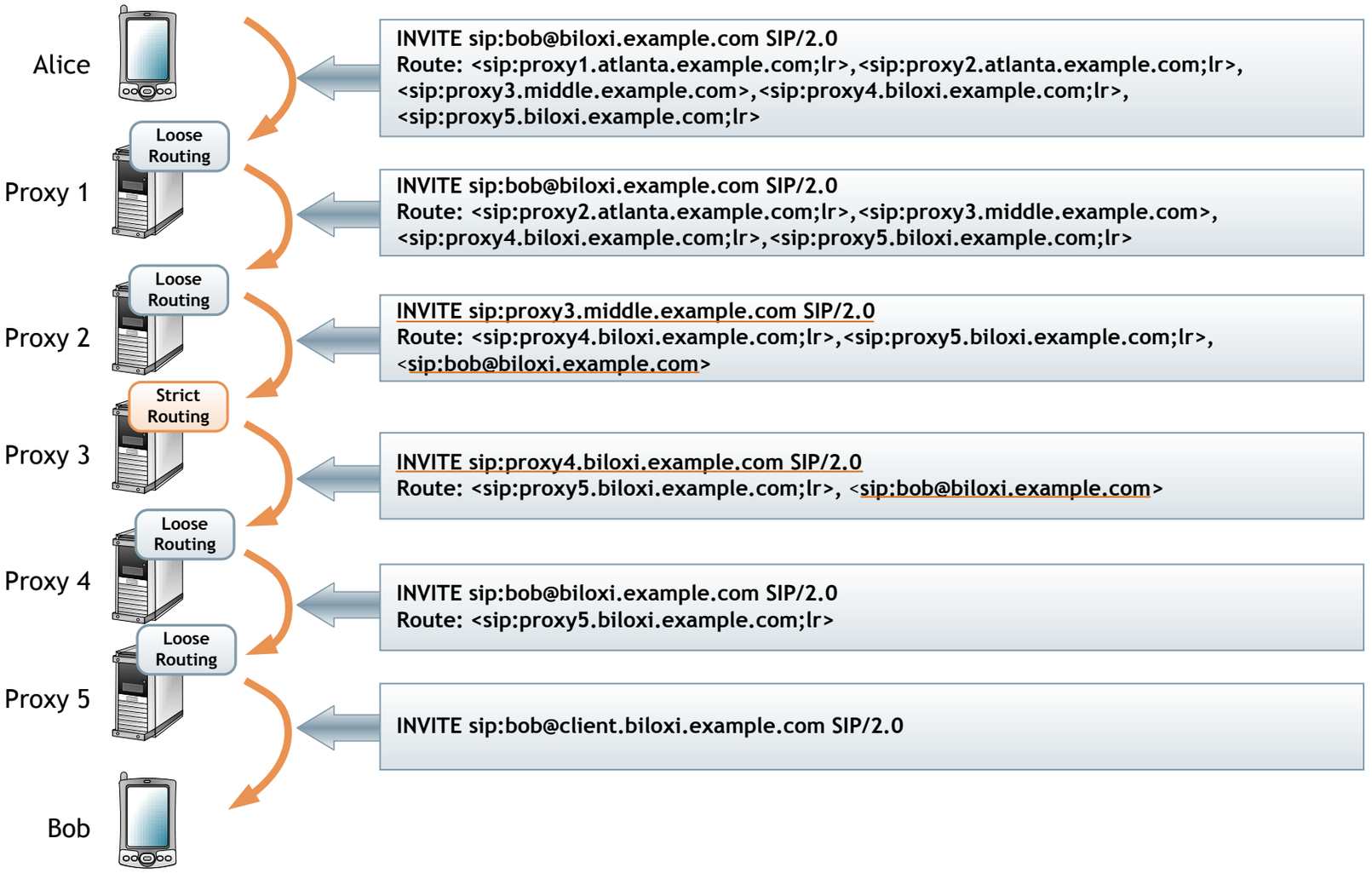
Steps of Proxy Request Forwarding #4

Postprocess routing information: if the copy contains a Route header field, the proxy MUST inspect the URI in its first value. If that URI does not contain lr parameter, the proxy MUST modify the copy as follows:

- The proxy MUST place the Request-URI into the Route header field as the last value.
- The proxy MUST then place the first Route header filed value into the Request-URI and remove that value from the Route header field.



Traversing a Strict-Routing Proxy



Steps of Proxy Request Forwarding #5

Determine Next-Hop Address, Port, and Transport: the proxy MAY have a local policy to send the request to a specific IP address, port, and transport, independent of the values of the Route and Request-URI. Such a policy MUST NOT be used if the proxy is not certain that the IP address, port, and transport correspond to a server that is a loose router.

Add a Via header field value: the proxy MUST insert a Via header field value into the copy before existing Via headers. This implies that the proxy will compute its own branch parameter (globally unique for that branch) used for detection loops.

Note: *Server Policy* is the collective set of rules that govern proxy routing decision making.

Steps of Proxy Request Forwarding #6

Add a Content-Length header field if necessary: if the request will be sent to the next hop by using a stream-based transport and the copy contains no Content-Length header value, the proxy MUST insert one.

Forward Request: a stateful proxy MUST create a new client transaction for the request and instructs the transaction to send the request using the address, port and transport determined in step 7.

Set timer C: in order to handle the case where an INVITE request never generates a final response, the TU uses a timer which is called timer C. Timer C MUST be set for each client transaction when an INVITE request is proxied. The timer MUST be larger than 3 minutes.

Response Processing (5)

When a response is received by an element, it first tries to locate a client transaction matching the response. If none is found, the element MUST process the response as a stateless proxy. If a match is found, the response is handed to the client transaction.

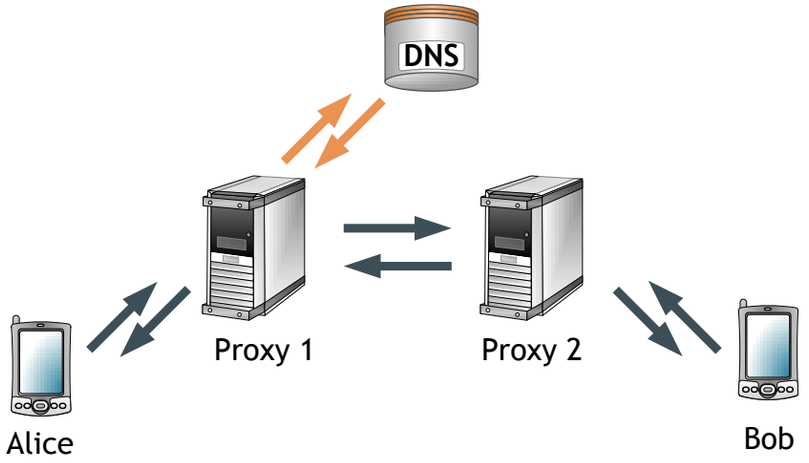
The stateful proxy MUST choose the “best” final response among those received and stored in the response context.

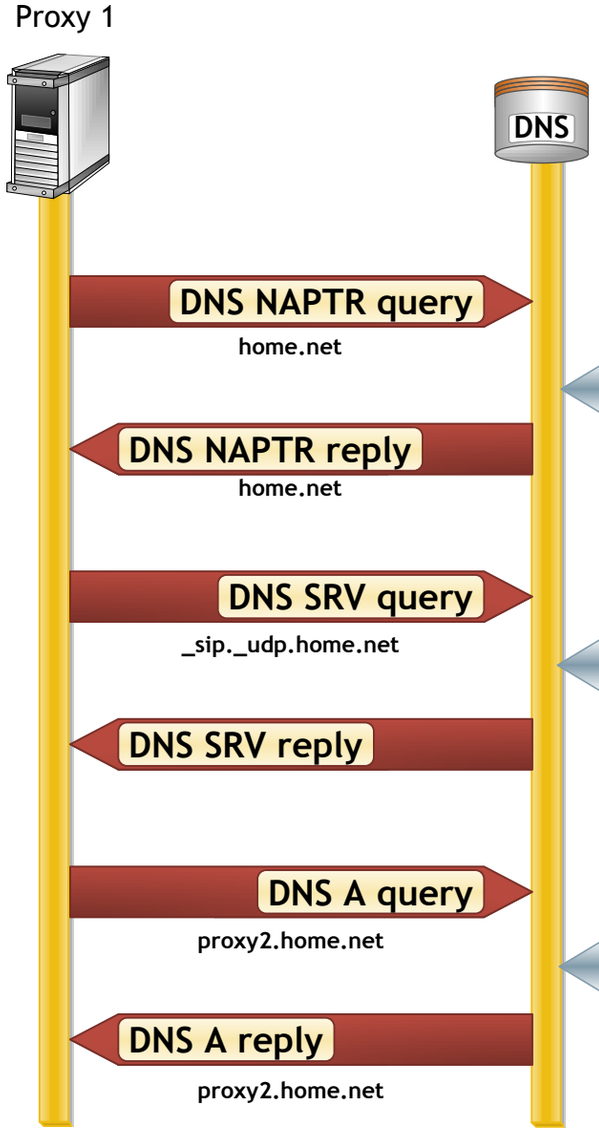
If there are no final responses in the context, the proxy MUST send a 408 (Request Timeout) response to the server transaction.

Steps of Response Processing

a) Find the appropriate response context
b) Update timer C for provisional responses
c) Remove the topmost Via
d) Add the response to the response context
e) Check to see if this response should be forwarded immediately
f) When necessary, choose the best final response from the response context
g) Aggregate authorization header field values if necessary
h) Optionally rewrite Record-Route header fields values
i) Forward the response
j) Generate any necessary CANCEL requests

DNS procedures to locate a SIP server





Domain	TTL	Class	Type	Order	Preference	Flag	Service	Regexp	Replacement
home.net	7200	IN	NAPTR	50	50	"S"	"SIP+D2T"	""	_sip._tcp.home.net
home.net	7200	IN	NAPTR	50	50	"S"	"SIP+D2T"	""	_sip._tcp.home.net

Service, Proto, Name	TTL	Class	Type	Priority	Weight	Port	Target
_sip._udp.home.net	7200	IN	SRV	0	1	5060	proxy2.home.net
_sip._udp.home.net	7200	IN	SRV	0	3	5060	proxy3.home.net

Name	TTL	Class	Type	Address
proxy2.home.net	7200	IN	A	192.0.2.222

DNS procedures

DNS procedures allow a client to resolve a SIP URI into the IP address, port, and transport protocol of the next hop.

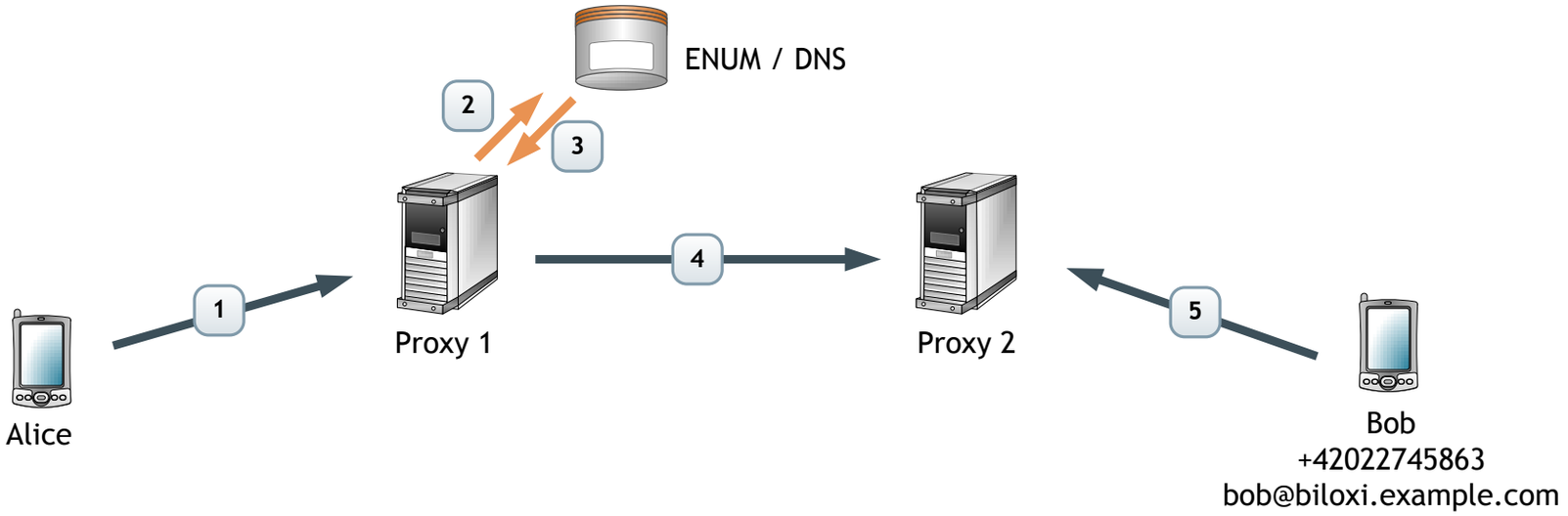
Basic types of DNS query:

- NAPTR – provides a mechanism for the called domain to specify which protocols it prefers a SIP request to use
- SRV – discovers servers and gives information about ports
- DNS A/AAAA – used for translating URI to IP address (A – IPv4, AAAA – IPv6)

ENUM

- ENUM = E.164 Number Mapping
- ENUM defines a method to convert a regular telephone number into a format that can be used in Internet within DNS to look up Internet addressing information (e.g. URIs)
- More than one contact information can be stored in the DNS record that is belonging to a specific ENUM number -> one identifier for more services
- There is allocated a specific zone, namely "e164.arpa" for use with E.164 numbers. Any phone number, such as +420222745863 can be transformed into a hostname by reversing the numbers, separating them with dots and adding the e164.arpa suffix thus: 3.6.8.5.4.7.2.2.2.0.2.4.e164.arpa.

ENUM



1 INVITE tel:+420222745863 SIP/2.0

2 INVITE sip:bob@biloxi.example.com SIP/2.0

3 ENUM/DNS query: 3.6.8.5.4.7.2.2.2.0.2.4.e164.arpa

4 5 ENUM/DNS response: sip:bob@biloxi.example.com

Notes

Two alternatives of carrying telephone number in INVITE:

- INVITE tel:+420222745863 SIP/2.0
- INVITE sip:+420222745863;user=phone SIP/2.0

- URL – Uniform Resource Locator (RFC 2806); defines three URL schemes: tel, fax, modem
- URI – Uniform Resource Identifier (RFC 3966); defines one URI scheme tel for all phone numbers (telephones, telefaxes or modems) and extension ext

Note: RFC 2806 is obsolete, but IANA still registers schemes fax and modem.

SIP addressing

- SIP addressesuniquely identify:
  - Users,
  - Services
  - Equipment.
- SIP applications are using the same addressing scheme to name users and services as email, web and other Internet applications
  - Unified Resource Identifiers, URIs.
- User Bob:
  - mailto:bob@company.org
  - sip:bob@company.org
- SIP addresses are generally used to uniquely identify a sender or receiver of a message:
  - User,
  - an IP phone,
  - a service,
  - or a telephone number.
- While not frequently used, a URI may even be unrelated to any kind of telephony:
  - For example a SIP server may redirect a caller to a web URI for a textual announcement to be rendered on his screen.

Public SIP user address (AOR)

- Formally called address of record (AOR).
  - sip:bob@company.org
- This address is printed on business-cards, stored in phonebooks, and displayed on web-pages for use with click-to-dial clients.
- A client attempting to send to this address will use:
  - DNS extensions NAPTR and SRV to find out:
    - IP address,
    - Transport protocol (TCP/UDP/UDP)
    - port number to communicate with the machine ”company.org”.
- A request sent to this addresses reaches a SIP server.
- The server then resolves it using the database of registrations
- The database is filled and actualized by Registrar server.

Address parameters

- SIP address can be accompanied with a list of parameters
- For example the parameters can dictate use of a specific transport protocol
  - sip:bob@company.org;transport=tcp
- The most frequently used parameter is ”;lr” which is used to indicate RFC3261-compliant proxy server in Record-Route header-fields.

Address variants

- sip:10.0.0.1:16001;transport=udp - Specifies IP address port and transport protocol. Can be used to associate UA address with AOR.
- sips:bob@company.org - implies hop-by-hop TLS encryption all the way down from client to the final recipient.
  - Hops that cannot guarantee secure forwarding are supposed to decline such SIP request.
  - There is no way to detect a server in the SIP that cheats and forwards the SIP traffic in plain-text!!

Address variants

- tel:+420-123-456-789 - sender of SIP message shall find a way to break out to the PSTN. Unlike SIP URIs, it leaves the routing choice open in that there is no server name.
  - Usually an outbound SIP server resolves a tel URI into SIP URI using some Least-Cost-Routing
  - number database like ENUM could help
  - tel URIs should include a full international E.164 number
  - numbers following a private dialing plan are allowed as well.

Address variants

- sip:anouncements@company.org – basically the same as users URI but provides rather the address of some machine than a living user's address
- http://company.org/bob/away_message.http - Non-telephony URI
  - webpage content may be shown on callers display
  - allows meshed internet applications to be created

SIP address/URI general scheme

protocol:[user@]host[:port][;parameter=value]

- Protocol discriminator - identifies the protocol to be used to communicate with the recipient
  - sip, sips, tel, http,...
- host – DNS resolvable name of host or its direct IP
- port – port may be specifyed
- parameter – other parameters may be included (transport, uri,)

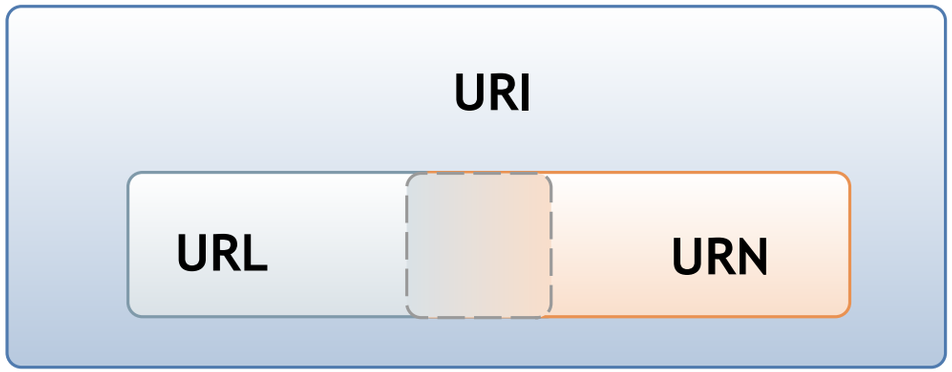
URI / URL / URN / AOR

- URI - RFC 2396 - Uniform Resource Identifiers (URI) Generic Syntax.
- URL - Uniform Resource Locator (URL) is a Uniform Resource Identifier (URI) that specifies where an identified resource is available and the mechanism for retrieving it.

URI / URL / URN / AOR

- URN - A Uniform Resource Name (URN) is a Uniform Resource Identifier (URI) that uses the urn scheme, and does not imply availability of the identified resource. Both URNs (names) and URLs (locators) are URIs, and a particular URI may be a name and a locator at the same time.

URI / URL / URN / AOR



URI / URL / URN / AOR

- AOR - An address-of-record (AOR) is a SIP or SIPS URI that points to a domain with a location service that can map the URI to another URI where the user might be available.
- Typically, the location service is populated through registrations.
- An AOR is frequently thought of as the "public address" of the user.

E.164

- ITU-T recommendation which defines the international public telecommunication numbering plan used in the PSTN.
- Also defines the format of telephone numbers.
- E.164 numbers can have a maximum of fifteen digits and are usually written with a + prefix.
- To dial such numbers from a normal fixed line phone, the appropriate international call prefix must be used.
- example: +420 221 234 567 (some Czech number in Prague)

Structure of MSISDN

- MSISDN - Mobile Subscriber ISDN Number

CC Country Code, e.g. CZ – 420
NDC National Destination Code, e.g. O_2 – 602
SN Subscriber Number, e.g. 150000