Computer Networks

Søren Debois
February 6, 2017

SECURITY F2017

Lecture 2 (MSc only)

Review

Last lecture

- Introduction to the course.
- What is IT Security?
  (Confidentiality, Integrity, Availability, Accountability)
- 12 Security Principles.
- Introduction to the command-line.


Meta

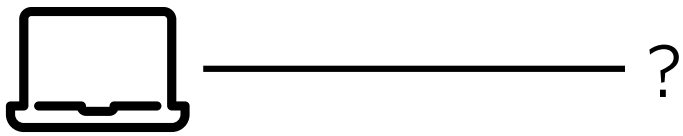
Exercise slots

- We keep some (not all) slots
- We re-arrange activities
- **Take the poll on learnit**

Peergrade

- Excellent submission rates
- Now provide excellent feedback!

Introduction



We have a computer with a network interface.
How do we retrieve the Google homepage?

Plan

- Physical communication
- Point-to-point communication
- Internetworking
- Transmission control
- The domain name system
- Hypertext transport protocol
- The OSI model

Foundations of Networking

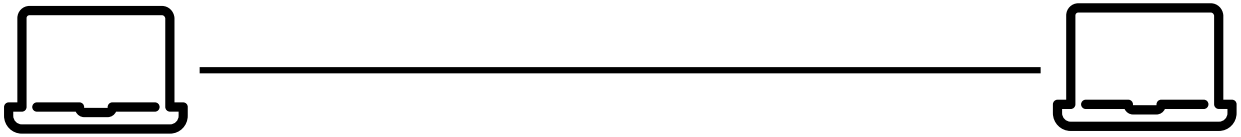
Terminology

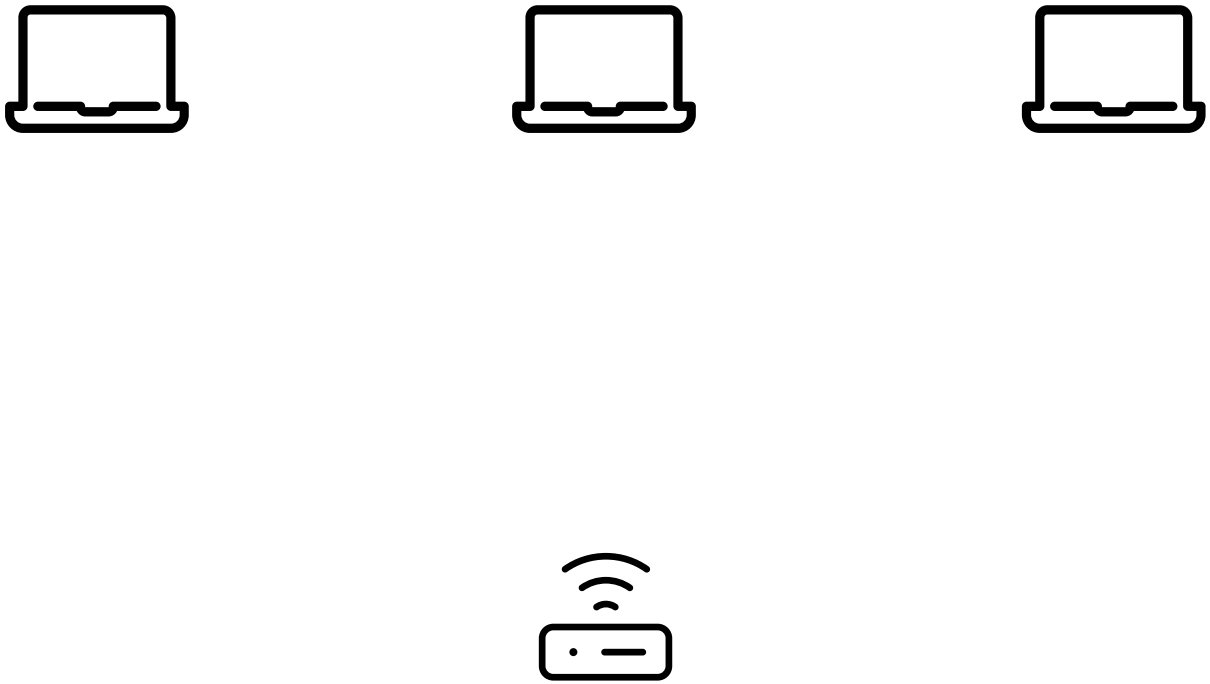
- **Processes** run on **hosts**
- Processes send and receive **messages** on communication **channels**
- Processes adhere to **protocols**:
  agreed-upon sequences of message exchanges and data formats
- Messages have **header** and **payload**

Basic problem

- Channels are subject to **failures**
- Messages may be lost, scrambled, duplicated, and re-ordered

Physical layer





Send messages on physical medium

- The **physical layer** arranges the transmission of short binary messages
- Broadcast only
- Direct link required
- No guarantees of delivery
- No guarantees of correctness

The physical layer

- Responsibility:
  Transmission of binary data across a physical link

- Usually broadcast

- Digital/analog conversion, usually amplitude/frequency modulation

- Usually provides no guarantees:
  Your message may or may not arrive;
  it may or may not be modified along the way

- Put a magnet on the cable, turn on the microwave ...

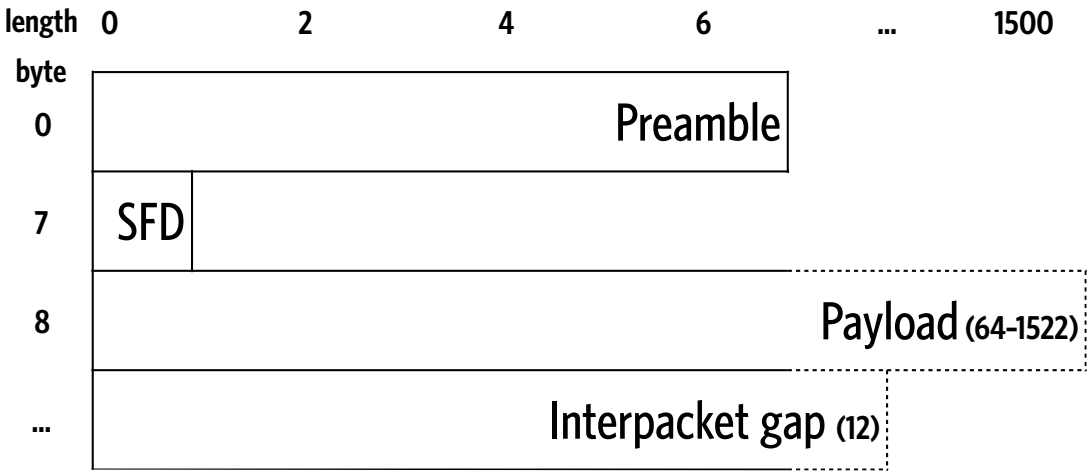
Examples

- IEEE 802.3: Ethernet
  (wired networks, electrical signals, fibre optics)

- IEEE 802.11: Wifi
  Bluetooth SIG: Bluetooth
  (wireless networks, radio)

- Lots and lots of others

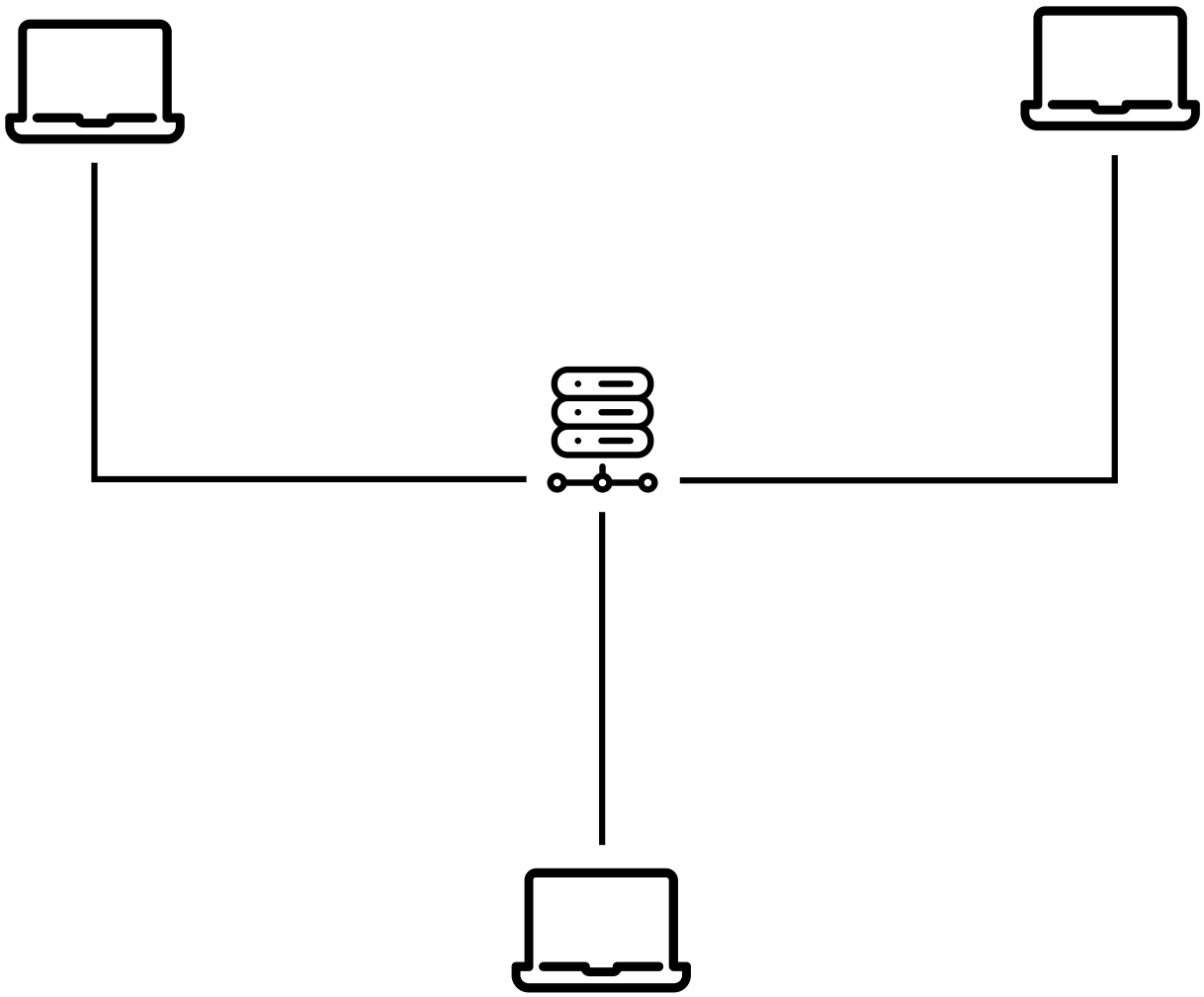
802.3: Ethernet (packet)



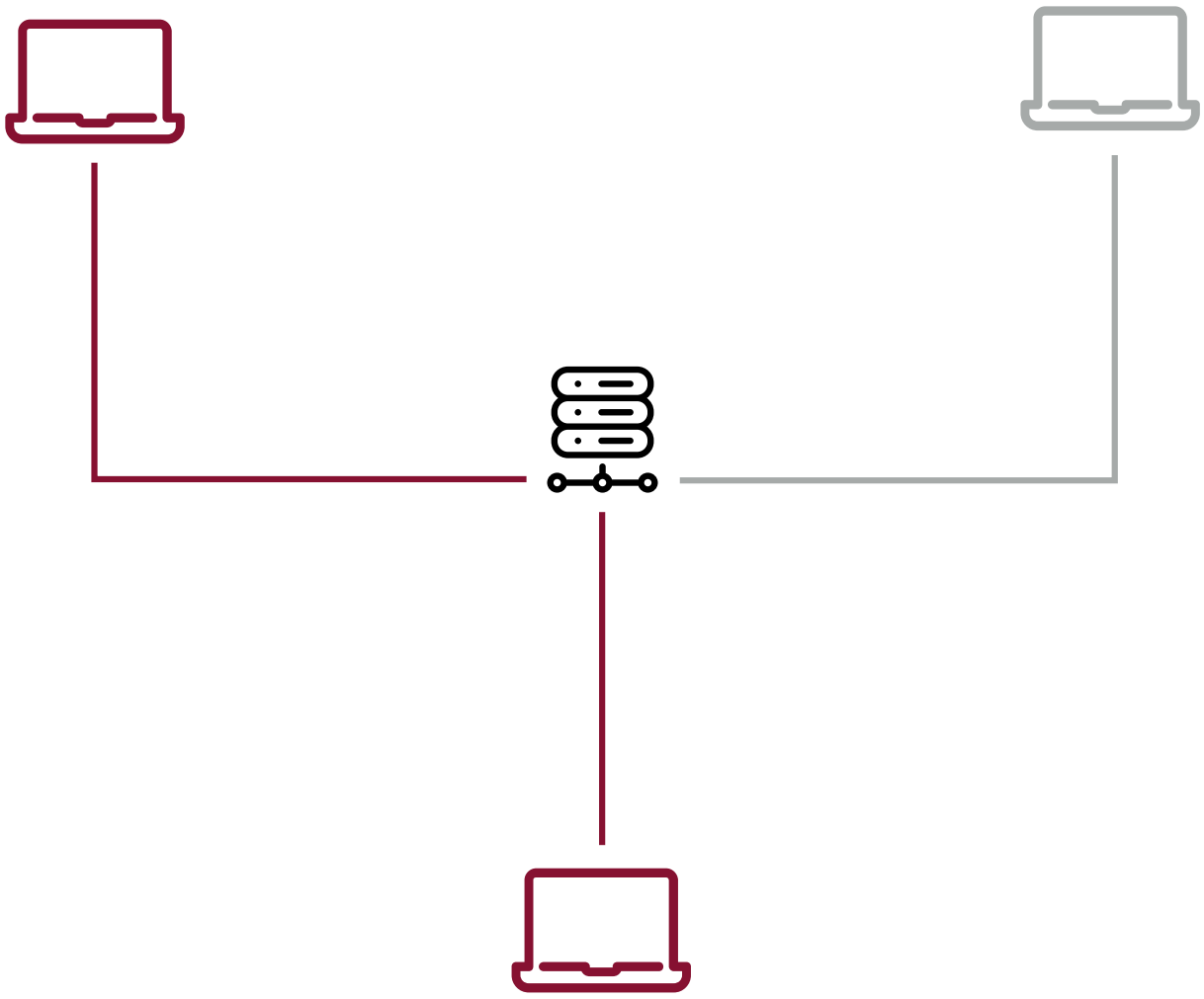
- Preamble:
  10101010 10101010 10101010
  10101010 10101010 10101010
  10101010
- SFD:
  Start-of-Frame Delimiter:
  10101011
- End-of-frame, e.g., loss of carrier.
  Frame gap: No data, just wait.

Hub

- Not every pair of machines inside the ITU has a direct physical link.
- N-way hub: N physical links, broadcasts Ethernet frame on one link to all other links.
- Key limitation:
  Scalability



Data link layer



Point-to-point communication

- How does a host send a message to a particular other host?
- The **data link** arranges the transmission of short binary messages across a physical link
- Point-to-point
- Direct link required
- No guarantees of delivery
- Good probability of correctness

The data-link layer

- Responsibility:
  Transmission of packets between hosts connected by a physical link

- Solves addressing:
  Media Access Control (MAC) addresses

- Solves (partly) reliability:
  Checksums

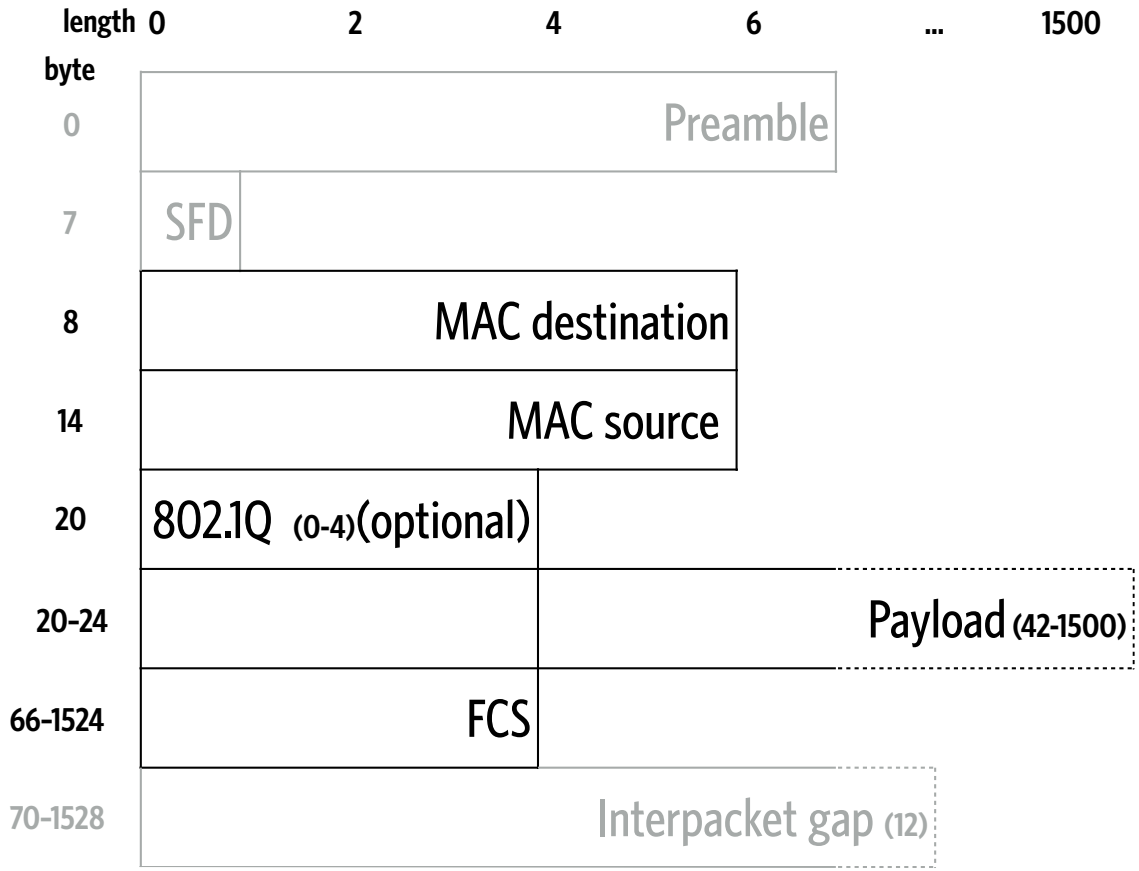
MAC Addresses

- 6-byte identifier (48 bits)

- Usually
  3-byte “Organisationally Unique Identifier”
  3-byte “Network Interface Controller”

- Broadcast address FF:FF:FF:FF:FF:FF

802.3: Ethernet (frame)



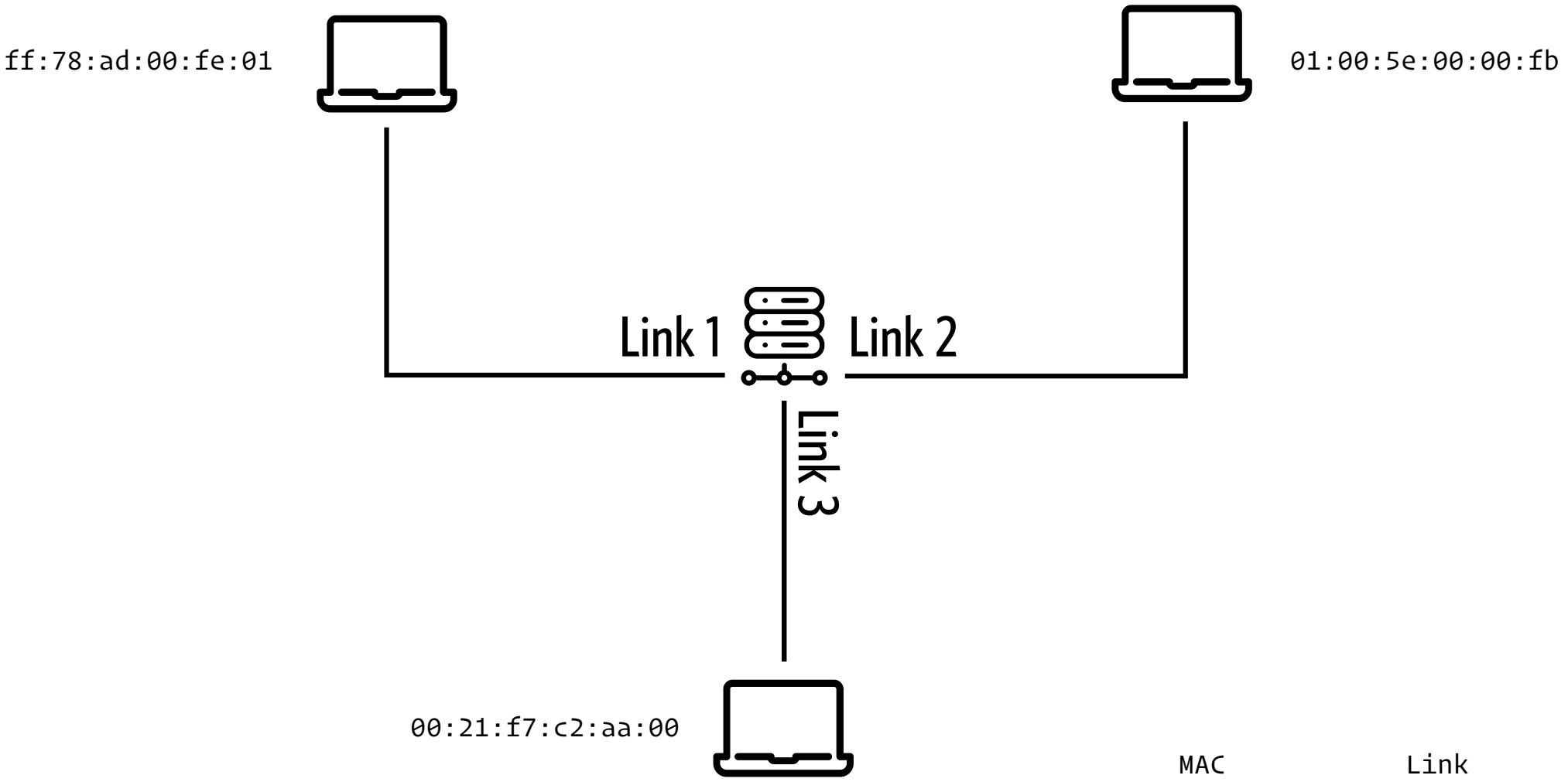
- MAC destination, source: "To", "From"
  (May be broadcast)
- Payload (actual contents) up to 1500 bytes
- Frame Check Sequence:
  32-bit Cyclic Redundancy Check checksum
  Detects error bursts < 32 bit
- Check failed => Frame dropped

Switches

- Improve on hub by using link-layer information (source, destination)
- N-way switch: N physical links; incoming frames forwarded to links where the MAC destination in the frame is
- How does it know what
  link that is?

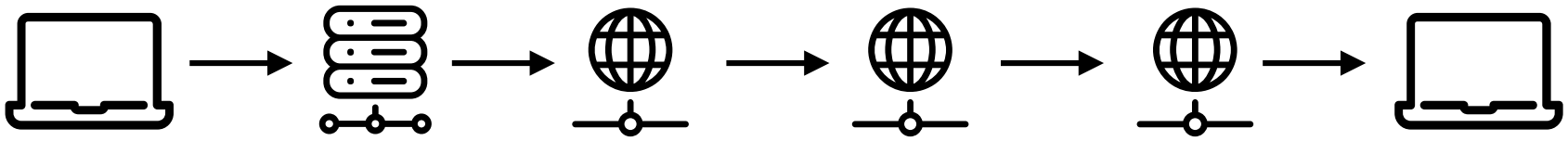
Switches

- A switch has a table of (MAC, Link) pairs
- When receiving a frame from MAC m on Link l, add (m, l) to the table
- If the table is full, discard the least recently used entry
- If a frame has a destination m and (m, l) is in the table, forward the frame on link l.
- If a frame has a destination m not in the table, forward the frame on all links (except the origin).



MAC	Link
ff:78:ad:00:fe:01	1
01:00:5e:00:00:fb	2

Network layer



Inter-network communication

- But Google doesn't have a host inside the ITU network!
- **Network layer**

Network layer

- IP protocol
  (IPv4)
- Hosts identified by IP addresses
- Best-effort (unreliable) delivery
- May introduce packet duplication, out-of-order delivery

IPv4 Addressing

- 32bit number identifies a host, written as 4 8-bit numbers:
  192.168.1.1, 130.226.142.220,

- CIDR: identifying IP ranges by IP-address + number of relevant bits in prefix:

  130.226.132.0/30
  = [130.226.132.0; 130.226.132.3]

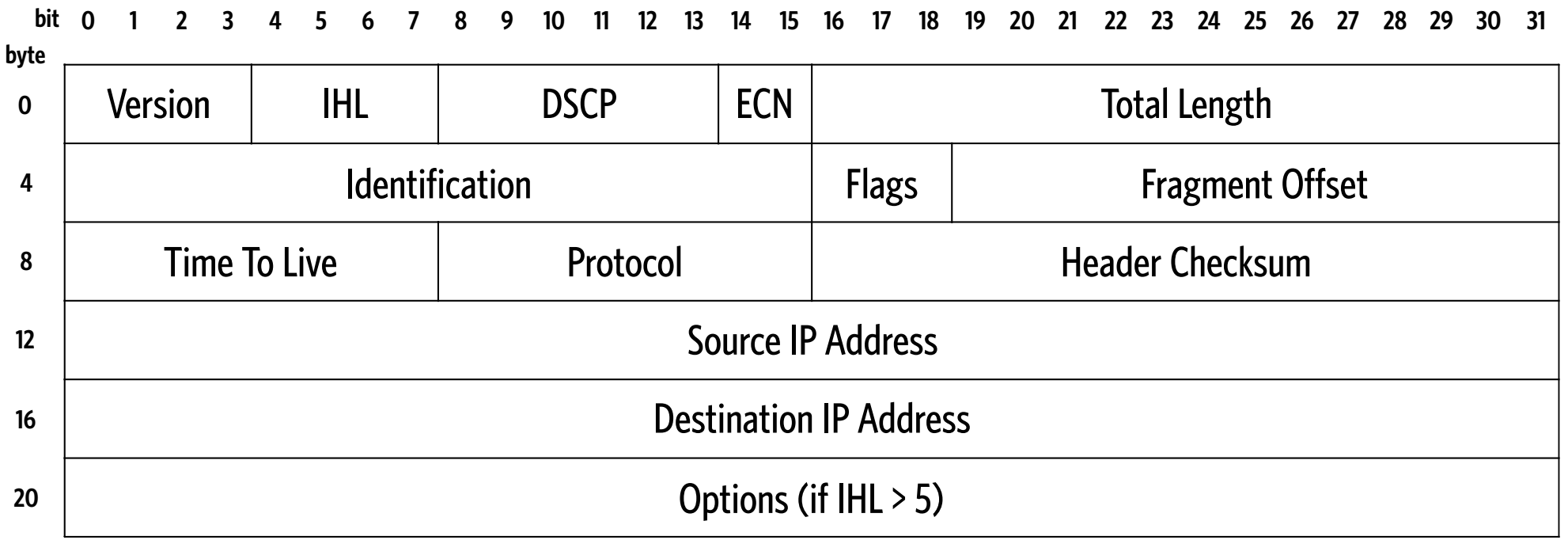
Reserved addresses

Block	Example IP	Usage
0.0.0.0/8	0.0.0.0	This
10.0.0.0/8	10.0.0.1	Private network
127.0.0.0/8	127.0.0.1	Loopback address
192.168.0.0/16	192.168.1.1	Private network
255.255.255.255/32	255.255.255.255	Limited broadcast

IP Operations

- Next-hop routing
- BGP
- MTU (v4 only), Fragmentation
- ICMP

IPv4 Header


bit 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
byte
0
Version
IHL
DSCP
ECN
Total Length
4
Identification
Flags
Fragment Offset
8
Time To Live
Protocol
Header Checksum
12
Source IP Address
16
Destination IP Address
20
Options (if IHL > 5)

Routers

- Table (IP space, Link)

```
> netstat -rnl
Routing tables

Internet:
Destination        Gateway            Flags    Refs      Use    Mtu  Netif Expire
default            10.28.0.1          UGSc      647        0   1500    en0
10.28/23           link#4             UCS         3        0   1500    en0
10.28.0.1/32       link#4             UCS         2        0   1500    en0
10.28.0.1          0:21:f7:c2:aa:0    UHLWIir   648       52   1500    en0    390
10.28.0.77         link#4             UHLWIi      1       14   1500    en0
10.28.0.146/32     link#4             UCS         1        0   1500    en0
10.28.1.255        link#4             UHLWbI      1      354   1500    en0
127                127.0.0.1          UCS         2        7  16384    lo0
127.0.0.1          127.0.0.1          UH         11    68489  16384    lo0
127.0.0.11         127.0.0.1          UHWIi       1        2  16384    lo0
169.254            link#4             UCS         1        0   1500    en0
224.0.0            link#4             UmCS        2        0   1500    en0
224.0.0.251        1:0:5e:0:0:fb      UHmLWI      1        0   1500    en0
255.255.255.255/32 link#4             UCS         2        0   1500    en0
255.255.255.255    link#4             UHLWbI      1      320   1500    en0
```

Internet Control Message Protocol

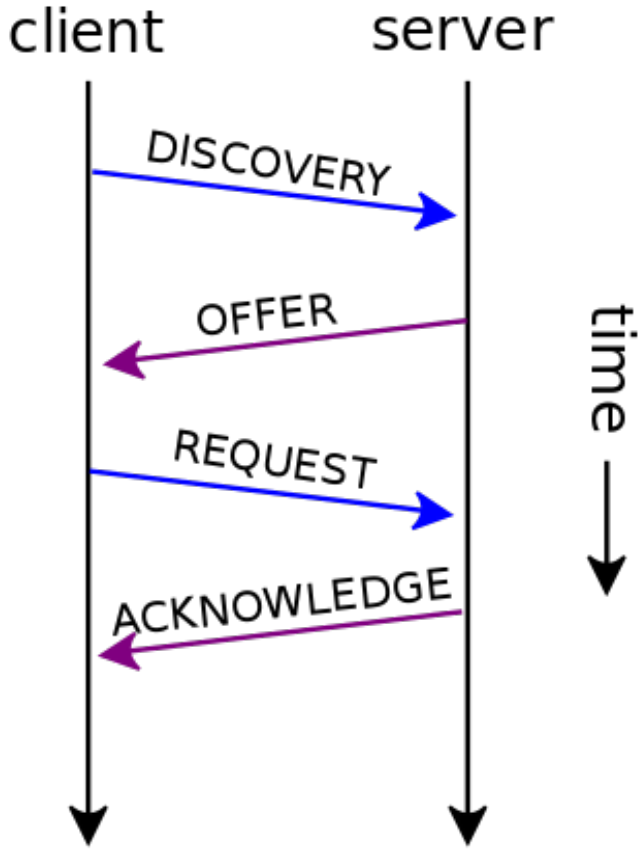
- Ping
- Traceroute

Getting an IP address

- Set it statically
- Dynamic Host Configuration Protocol (DHCP)

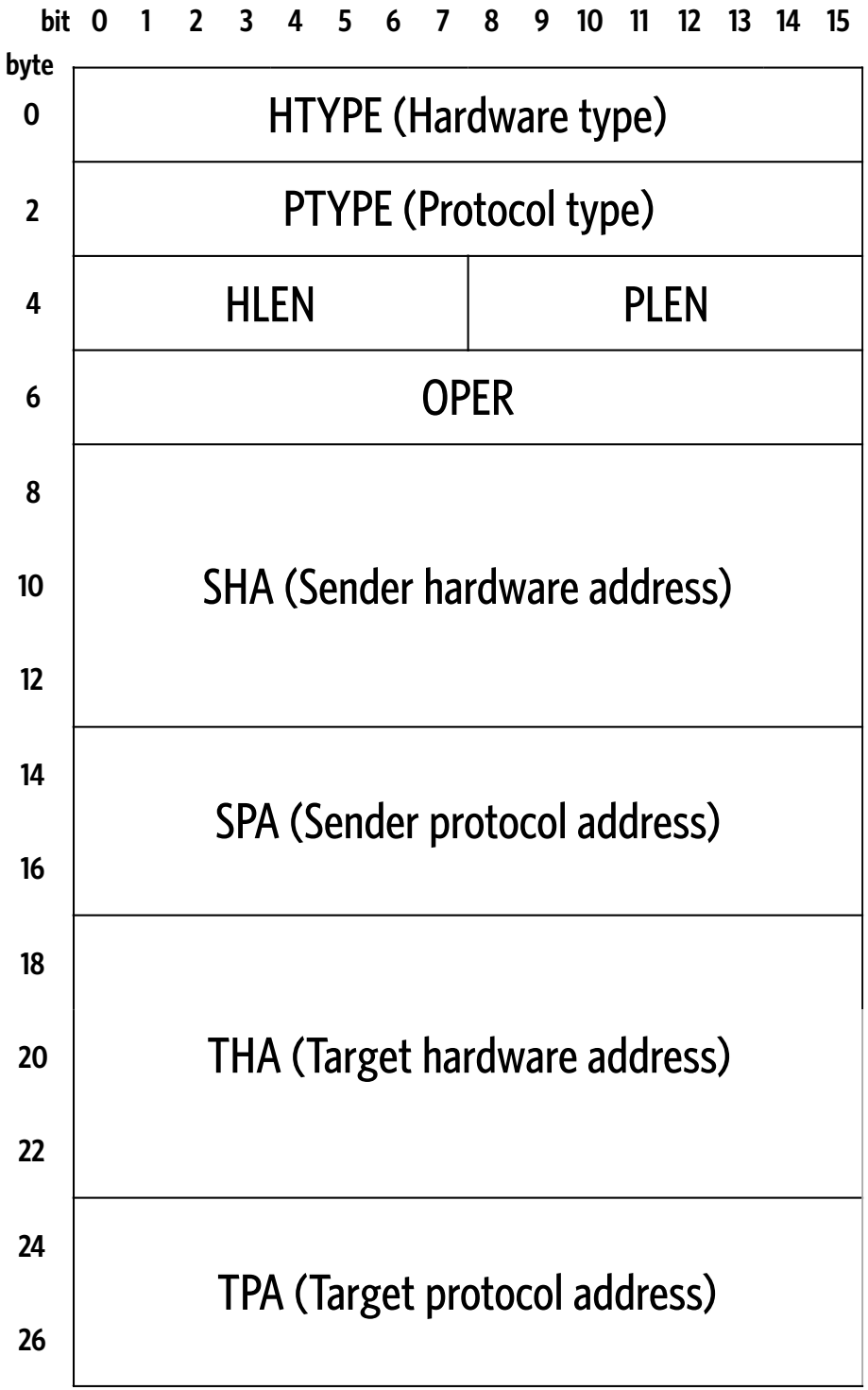
DHCP

- DHCPDISCOVER: UDP (!) packet broadcast on local net
- Server offers IP, DNS, domain name, ...
- Client requests particular IP
- Server acknowledges



Getting someone else's IP address

- Given an IP of a host on the local network, how does a host know the corresponding MAC?
- Address Resolution Protocol (ARP)
- Basic operation: broadcast request for MAC of given IP.



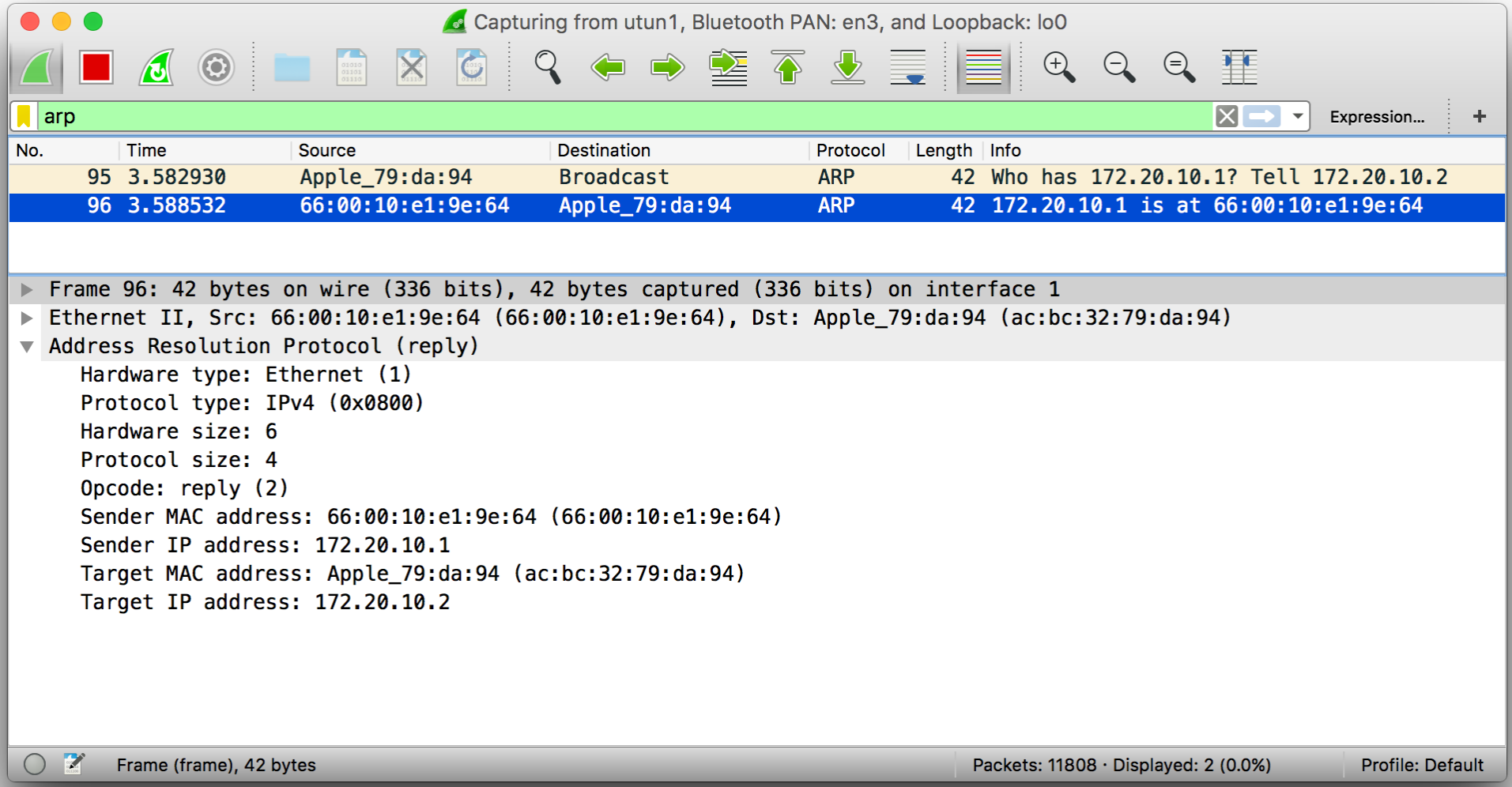
Address Resolution Protocol (ARP)

- Example on the left is for IPv4/Ethernet

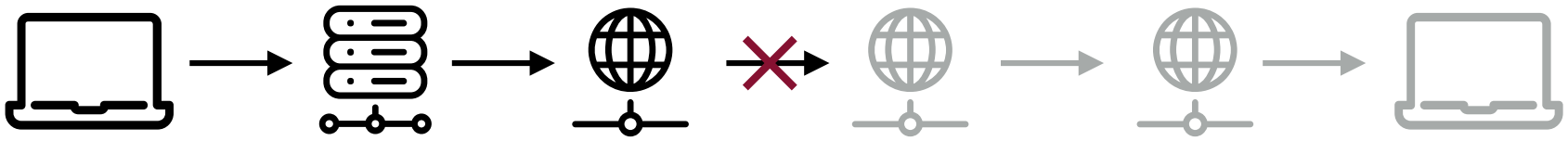
- OPER indicates request (1) or reply (2)

- **Request**
  "who-has TPA tell SHA"
  THA not significant

- **Response**:
  "TPA is THA'"


Capturing from utun1, Bluetooth PAN: en3, and Loopback: lo0
arp
Expression...
No. Time Source Destination Protocol Length Info
95 3.582930 Apple_79:da:94 Broadcast ARP 42 Who has 172.20.10.1? Tell 172.20.10.2
96 3.588532 66:00:10:e1:9e:64 Apple_79:da:94 ARP 42 172.20.10.1 is at 66:00:10:e1:9e:64
Frame 96: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 1
Ethernet II, Src: 66:00:10:e1:9e:64 (66:00:10:e1:9e:64), Dst: Apple_79:da:94 (ac:bc:32:79:da:94)
Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: 66:00:10:e1:9e:64 (66:00:10:e1:9e:64)
Sender IP address: 172.20.10.1
Target MAC address: Apple_79:da:94 (ac:bc:32:79:da:94)
Target IP address: 172.20.10.2
Frame (frame), 42 bytes
Packets: 11808 · Displayed: 2 (0.0%)
Profile: Default

Transport layer



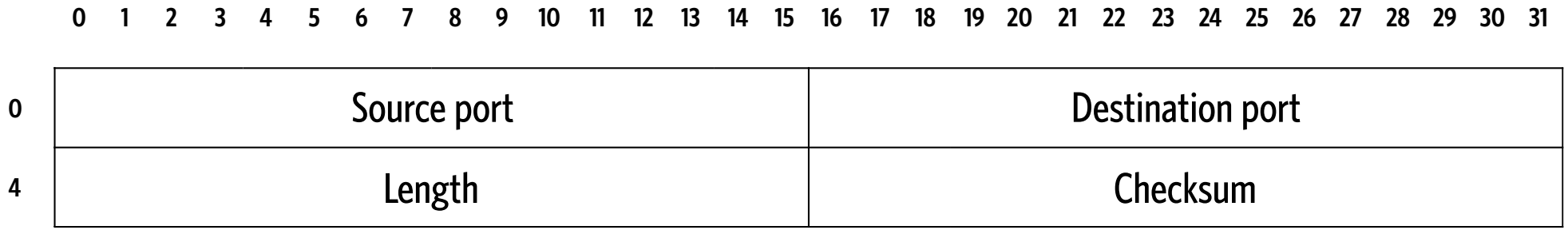
Problem 3

- Have: Host-to-host communication
  Need: Process-to-process communication

- Have: Unreliable messaging
  Need: Reliable messaging

- **Transport layer**

UDP



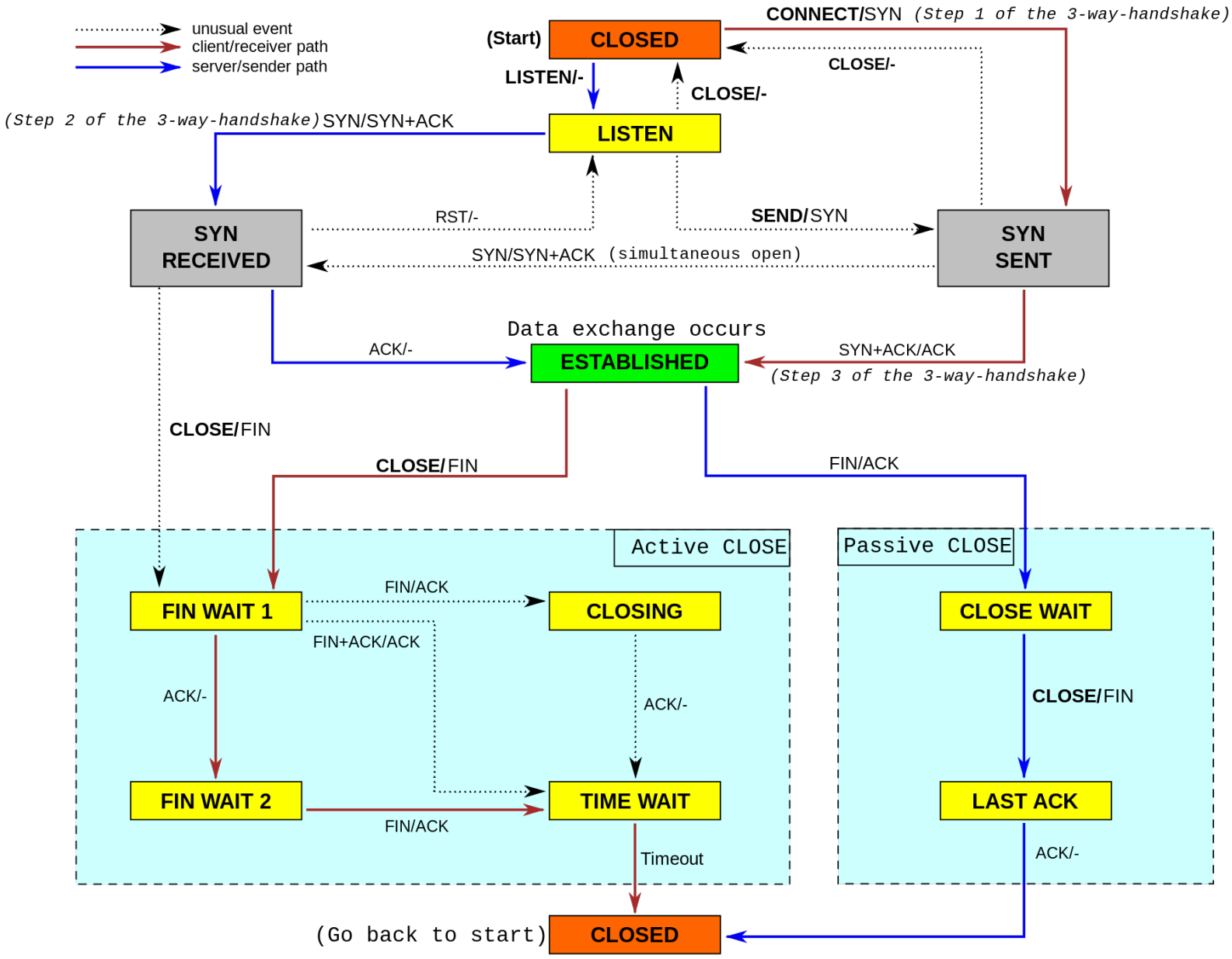
- Add source/destination **port** to IP.
- Useful when dropped frames are acceptable, e.g., streaming

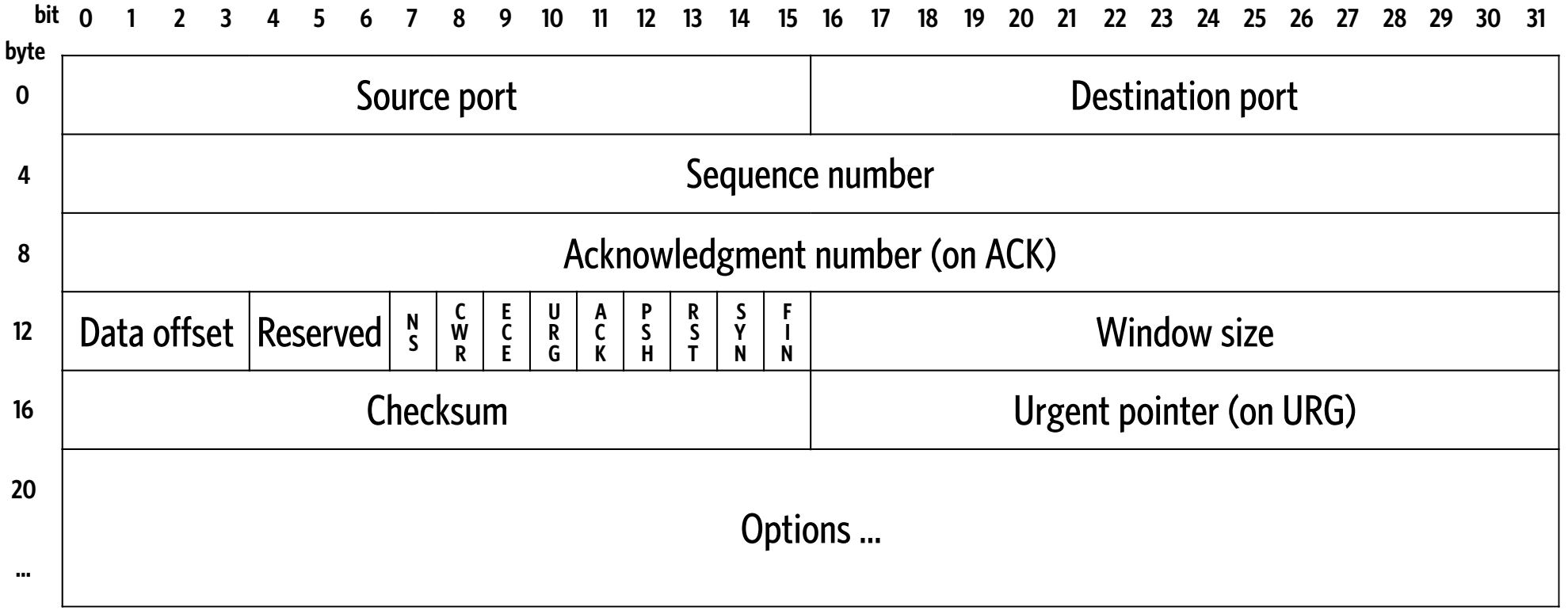
Reliability

- **Validity**: Any message sent is eventually delivered.
- **Integrity**: The message received is identical to the one sent, and no message is delivered twice.
- **Order:** if message A is sent before message B, A is delivered before B

TCP

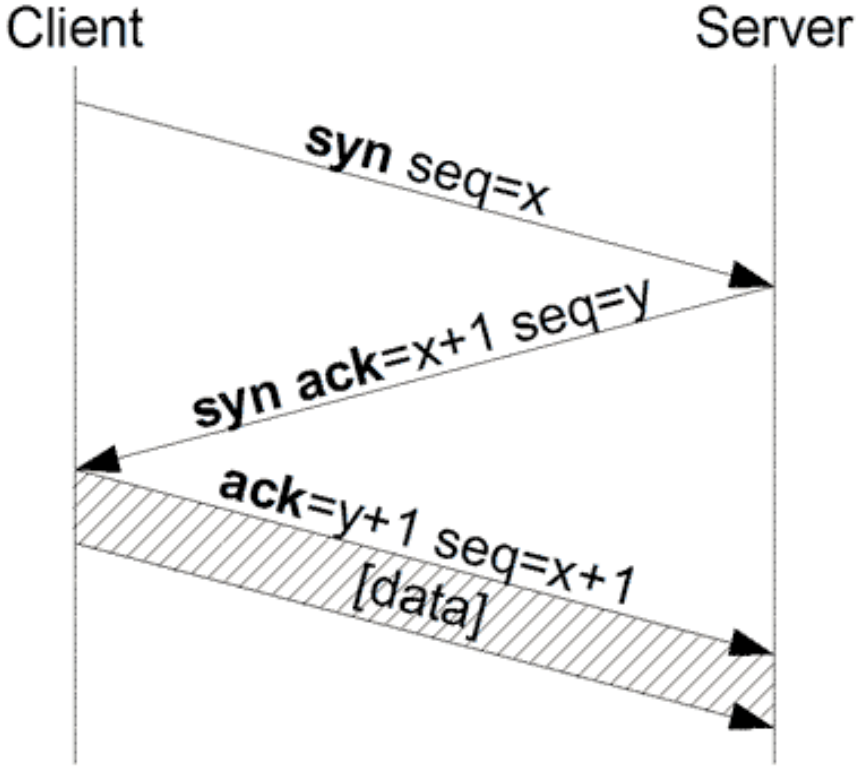
- Connection-oriented, reliable, streaming protocol.
- Achieved by message/acknowledgment sequence numbers, timeouts.
- Protocol specified as a fairly complex state machine
- Also: Flow control, congestion control


unusual event
client/receiver path
server/sender path
(Start)
CLOSED
CONNECT/SYN (Step 1 of the 3-way-handshake)
CLOSE/-
LISTEN/-
CLOSE/-
(Step 2 of the 3-way-handshake) SYN/SYN+ACK
LISTEN
SYN RECEIVED
RST/-
SEND/SYN
SYN SENT
SYN/SYN+ACK (simultaneous open)
Data exchange occurs
ACK/-
ESTABLISHED
SYN+ACK/ACK
(Step 3 of the 3-way-handshake)
CLOSE/FIN
CLOSE/FIN
FIN/ACK
Active CLOSE
Passive CLOSE
FIN WAIT 1
FIN/ACK
CLOSING
FIN+ACK/ACK
CLOSE WAIT
ACK/-
ACK/-
CLOSE/FIN
FIN WAIT 2
TIME WAIT
LAST ACK
FIN/ACK
Timeout
ACK/-
(Go back to start)
CLOSED



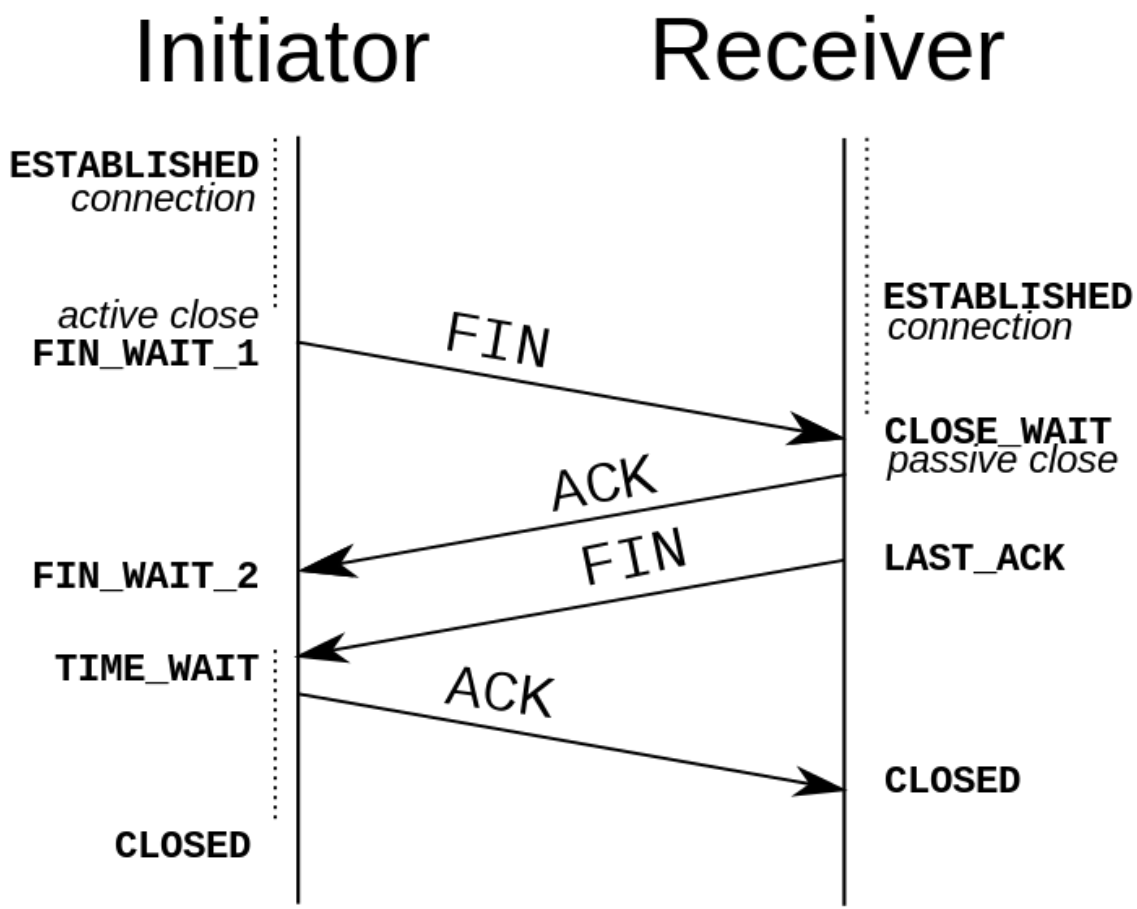
URG	out-of-band receive	SYN	synchronise sequence number
ACK	acknowledgment significant	RST	drop connection
PSH	do not buffer	FIN	last packet

Connection setup



The 3-way handshake

Connection teardown



Application layer

Domain-name System

Domain names

- How do I find the IP address for www.itu.dk?
- Using a query to the Domain-name system
- Premise: You must know *some* nameserver

DNS Resolver*

- Ask a nameserver for the IP of www.google.com
- If it knows, it tells you.
  In this case, the answer is either **authoritative** or **non-authoritative** (cached, TTL).
- If it doesn't, it tells you who to ask.
- Repeat.

* **Iterative**. Server may do it for you (**recursive**).

Hierarchy

- Root servers
- Zones (i.e., .com, .uk, .dk, ...)
- Delegation (.com -> google.com)
- Every nameserver knows a root server
- Every zone must have two authoritative servers

```
> dig @ns1.google.com www.google.com A

; <<>> DiG 9.8.3-P1 <<>> @ns1.google.com www.google.com A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39449
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.google.com.         IN A

;; ANSWER SECTION:
www.google.com.      300IN A  216.58.209.132

;; Query time: 21 msec
;; SERVER: 216.239.32.10#53(216.239.32.10)
;; WHEN: Wed Feb  1 16:23:27 2017
;; MSG SIZE  rcvd: 48
```

```
> dig www.google.com A

; <<>> DiG 9.8.3-P1 <<>> www.google.com A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11226
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 4

;; QUESTION SECTION:
;www.google.com.         IN A

;; ANSWER SECTION:
www.google.com.      147 IN A   216.58.201.164

;; AUTHORITY SECTION:
google.com.    162023 IN NS  ns1.google.com.
...

;; ADDITIONAL SECTION:
ns1.google.com.      164019 IN A   216.239.32.10
...

;; Query time: 2 msec
;; SERVER: 130.226.142.2#53(130.226.142.2)
;; WHEN: Wed Feb  1 16:15:14 2017
;; MSG SIZE  rcvd: 184
```

```
> dig www.google.com A +trace

; <<>> DiG 9.8.3-P1 <<>> www.google.com A +trace
;; global options: +cmd
.       334020 IN NS c.root-servers.net.
.       334020 IN NS h.root-servers.net.
...
;; Received 496 bytes from 130.226.142.2#53(130.226.142.2) in 38 ms

com.        172800 IN NS a.gtld-servers.net.
com.        172800 IN NS b.gtld-servers.net.
...
;; Received 492 bytes from 192.203.230.10#53(192.203.230.10) in 65 ms

google.com.    172800 IN NS ns2.google.com.
google.com.    172800 IN NS ns1.google.com.
...
;; Received 168 bytes from 192.42.93.30#53(192.42.93.30) in 23 ms

www.google.com.      300 IN A  216.58.209.132
;; Received 48 bytes from 216.239.36.10#53(216.239.36.10) in 79 ms
```

Hypertext Transport

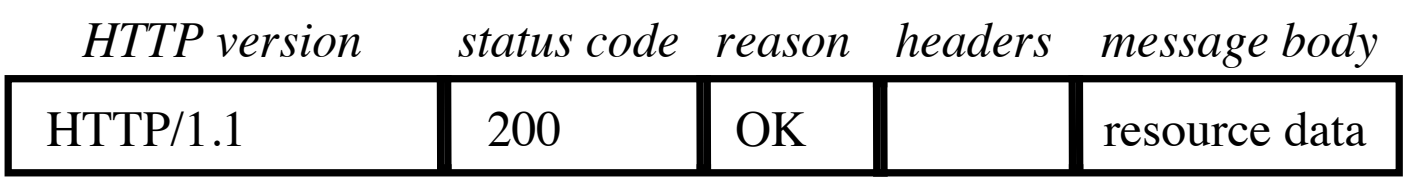
Getting the Google homepage

- Hypertext Transport Protocol (HTTP)
- Request-reply
- Request specifies which **resource** is requested, what encodings will be understood, etc*.
- Reply provides resource, caching information, redirection, ...
- Server may leave state with client in Cookie-header
- Client request may involve POST'ing information to the server

HTTP

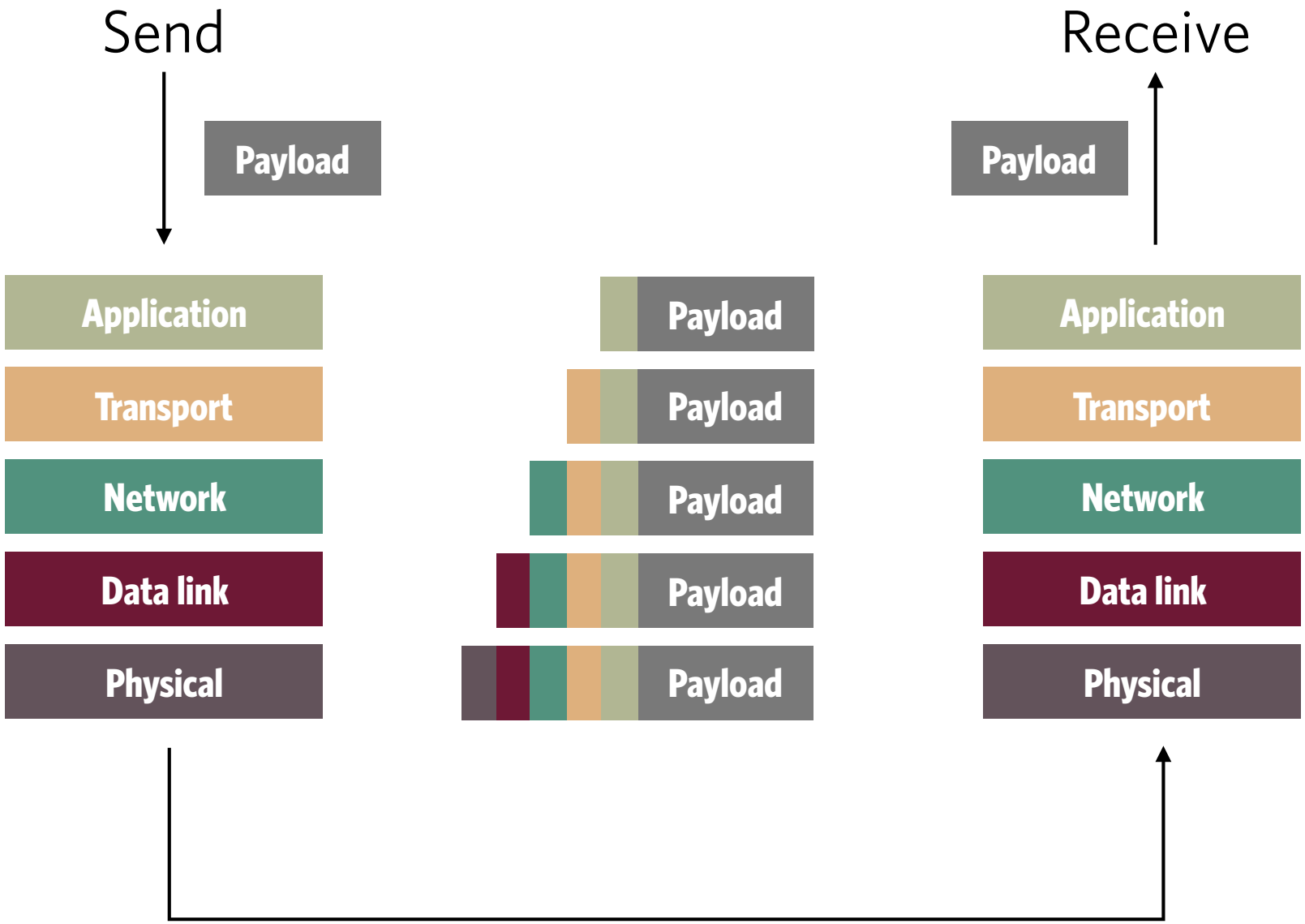
request

reply

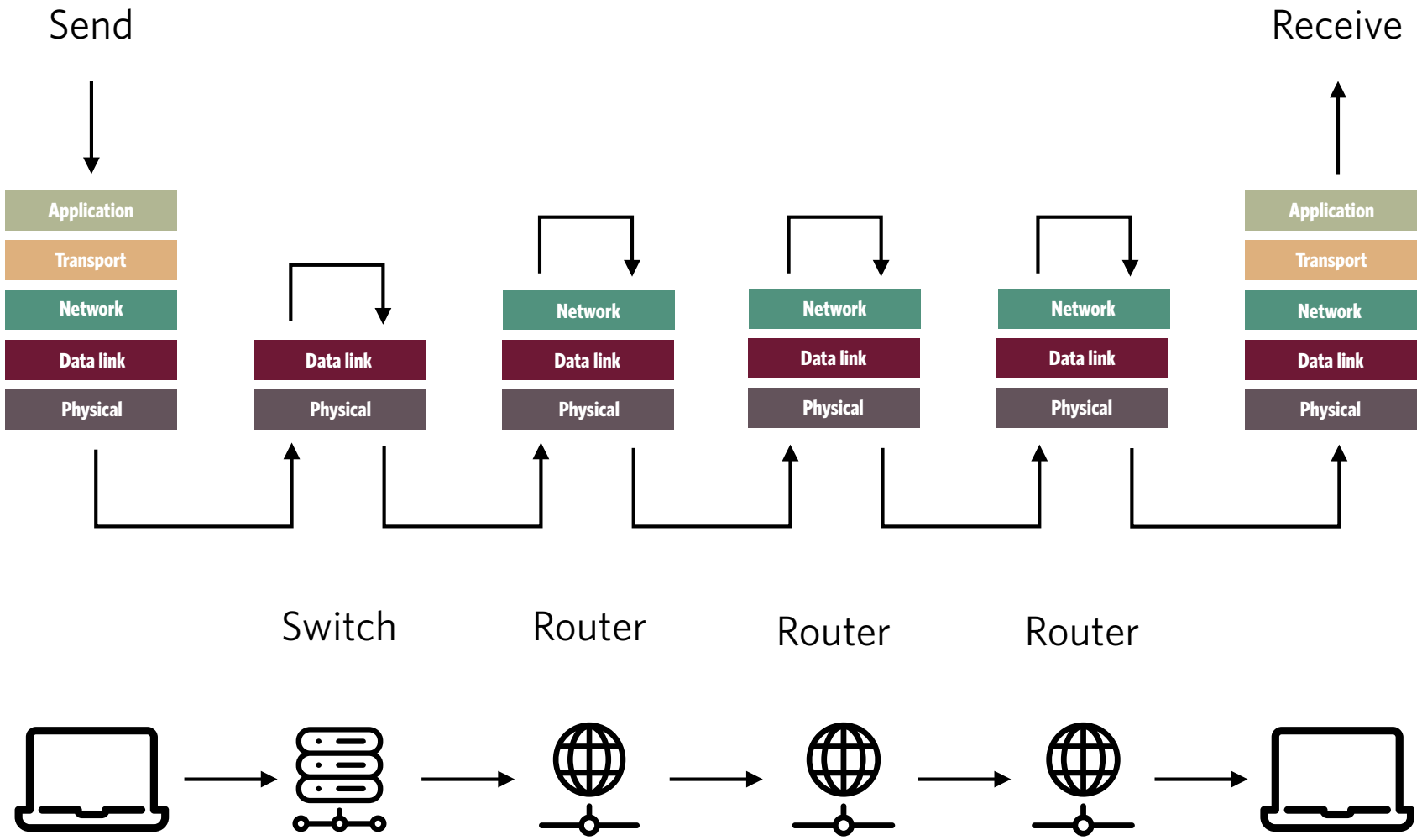


```
GET / HTTP/1.1
Host: www.itu.dk

Date: Thu, 02 Feb 2017 13:35:01 GMT
Server: Microsoft-IIS/7.5
Cache-Control: no-cache, no-store
Pragma: no-cache
Content-Type: text/html; charset=utf-8
Expires: -1
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Content-Length: 106032
Set-Cookie: ASP.NET_SessionId=o2l30gp4a4dfzafmplydtrla; path=/;
HttpOnly
Set-Cookie: cookieConsent=maybe; domain=.itu.dk; expires=Wed, 02-
Aug-2017 12:35:01 GMT; path=/

<!DOCTYPE html>
...
```

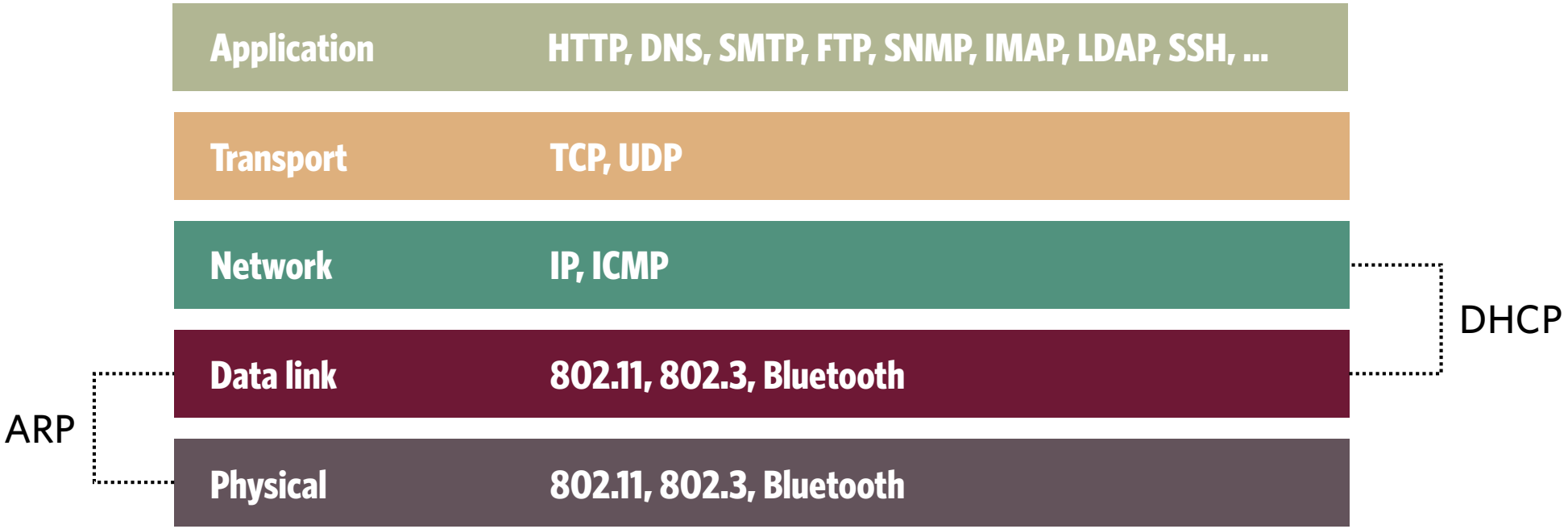
Protocol layers


Send
Payload
Application
Transport
Network
Data link
Physical
Payload
Payload
Payload
Payload
Payload
Receive
Payload
Application
Transport
Network
Data link
Physical

End-to-end


Send
Receive
Application
Transport
Network
Data link
Physical
Data link
Physical
Network
Data link
Physical
Network
Data link
Physical
Network
Data link
Physical
Application
Transport
Network
Data link
Physical
Switch
Router
Router
Router

TCP/IP Protocol layers

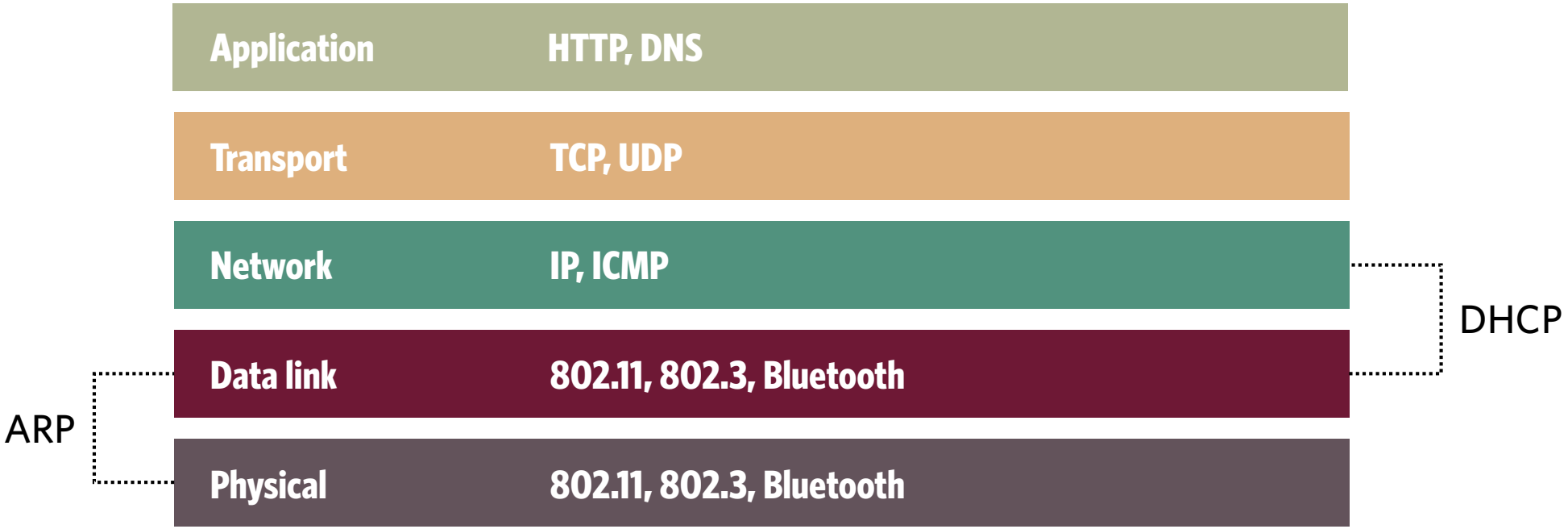


Full OSI model

Application	HTTP, DNS, DHCP, SMTP, FTP, IMAP, LDAP, SSH, ...
Presentation	MIME
Session	NetBIOS, PPTP, RTP, SOCKS, SPDY
Transport	TCP, UDP
Network	IP, ICMP
Data link	802.11, 802.3, Bluetooth, ARP
Physical	802.11, 802.3, Bluetooth

Summary

TCP/IP Protocol layers



Thank you!

- See learn-it for exercises etc.
- Questions?

Credits

Icons designed by Gregor Cesnar, FlatIcon

TCP Message Sequence Diagrams Wikipedia