Network Security

Søren Debois
February 13, 2017

Lecture 3

SECURITY F2017

Review

Goals & principles

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

Security goals

- **C**onfidentiality
  "Prevent unauthorised access to information."

- **I**ntegrity
  "Prevent unauthorised altering of information."

- **A**vailability
  "Ensure the availability of the system for authorised uses."

- Accountability
  "Actions of a principal may be traced uniquely to that principal."

12 Principles (1-6)

- Simplicity
- Open Design
- Compartmentalisation,
- Minimum exposure
- Least Privilege
- Minimum trust & maximum trustworthiness,

12 Principles (7-12)

- Secure fail-safe defaults
- Complete mediation
- No single point of failure
- Traceability
- Generating secrets
- Usability

Quiz results

- Mostly good.
- C1-08 (Not encrypting because only staff has access to wires) had overlapping answers, sorry.
- 25 (!) people did not get a passing grade.
  You will be contacted by TAs.

Peergrade exercises

- 52 submissions / 106 students ~ 49%, ok.
- 37 / 52 feedback ~ 71%, **not ok**.
- Generally good choice of news articles, good analysis.
- Use flags—very helpful!

Goals & principles

Peergrade top-5

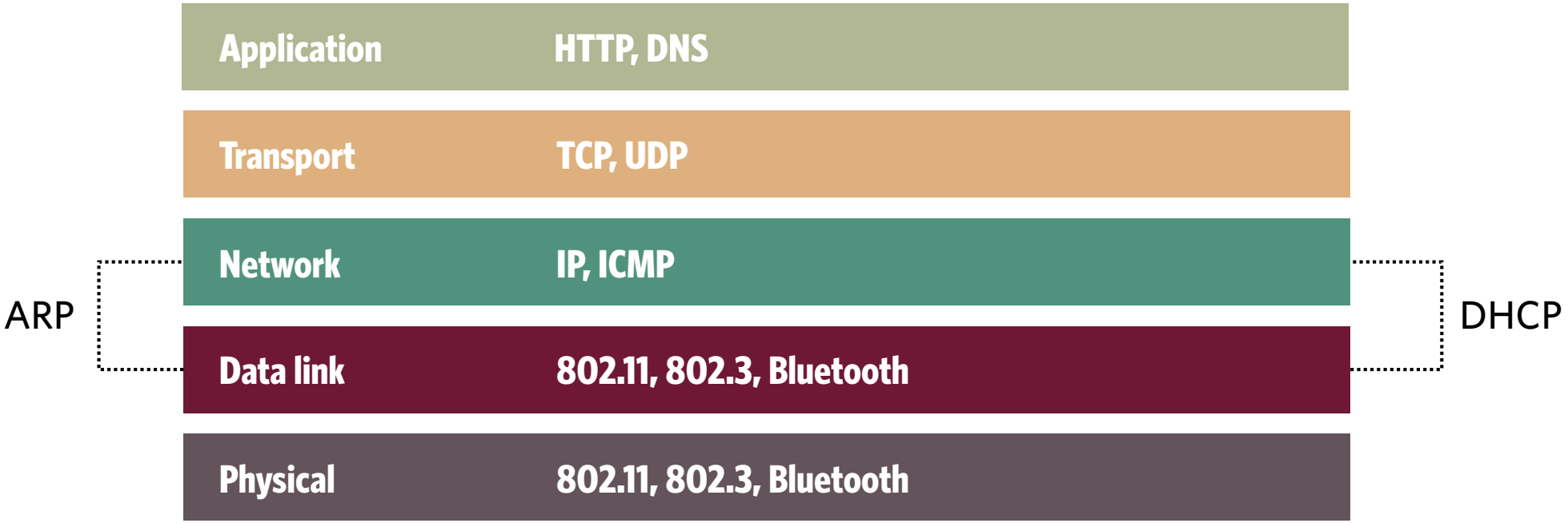
Submissions*

1. Anna Randak
2. Lasse Lange Jakobsen
3. Maurice Mugisha
4. Tom Roberts
5. Malthe Ettrup Kirkbro

Feedback

1. Niels Roesen Abildgaard
2. Alina-Roxana Preda
3. Josephine Sloth Rasmussen
4. Adam Vongrej
5. Lauritz Baess-Lehmann

Computer Networks




Meta

TAs and exercises

- Zero turnout most non-Mondays
- All but monday **cancelled**.
- Contact TAs for one-on-one or single-group questions via Personal Message on ublend.co.
- Use this! A TA is available *every day*.
- (Complain on ublend.co if you dislike this arrangement.)

Introduction

Plan

- Attacks on the network stack
- Port scanning
- Firewalls

The network stack

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

The network

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

Let's break it!

“What's really interesting is that these people will send a tube of live ants to anyone you tell them to.”

–Bruce Schneier

Foundations of Networking

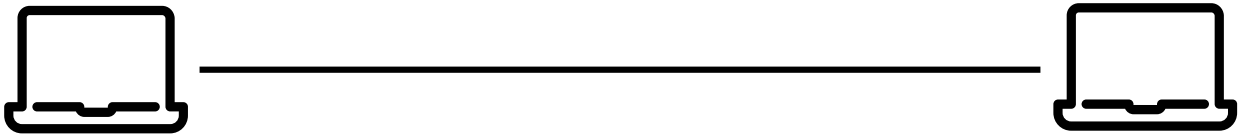
Adversary capabilities

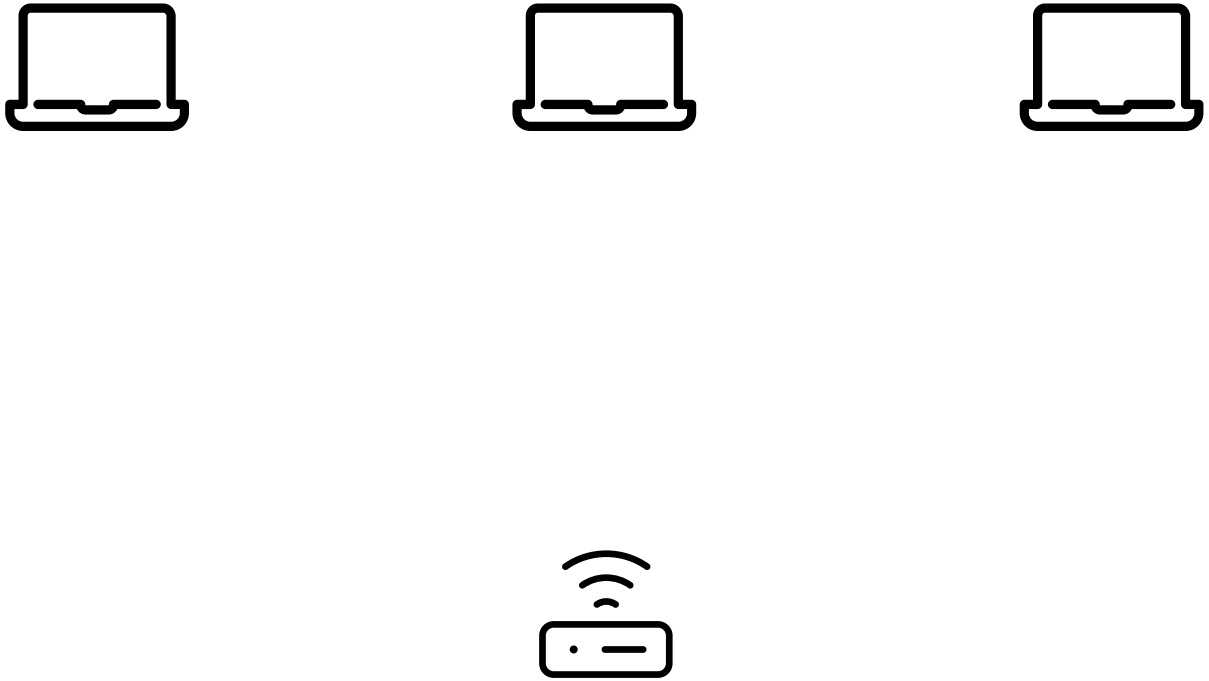
- The adversary has complete control of the network. He may:
  - intercept messages
  - replay messages
  - transform message
  - insert messages
  - delete messages

Adversary incapabilities

- The adversary cannot guess our secrets.
- We'll get back to this in the lecture on Applied Cryptography.
- Today, we're defenseless.

Physical layer





The physical layer

- Responsibility:
  Transmission of binary data across a physical link

- Usually broadcast
  (e.g., IEEE 802.3 Ethernet)

- Usually provides no guarantees.

The physical layer

- Responsibility:
  Transmission of binary data across a physical link

- **Usually broadcast**
  (e.g., IEEE 802.3 Ethernet)

- **Usually provides no guarantees**

Let's break it!

Confidentiality
Integrity
Availability
Accountability

Attacks on the physical layer

- **Eavesdropping** (confidentiality)
  Frames are broadcast;
  I can see them even if they aren't for me.

- **Tampering** (integrity)
  You won't detect my change

- **Denial-of-service** (availability)
  If I put enough noise on the line, you won't send or receive any messages.

- **Message injection**
  I can put arbitrary messages on the wire

Data link layer

The data-link layer

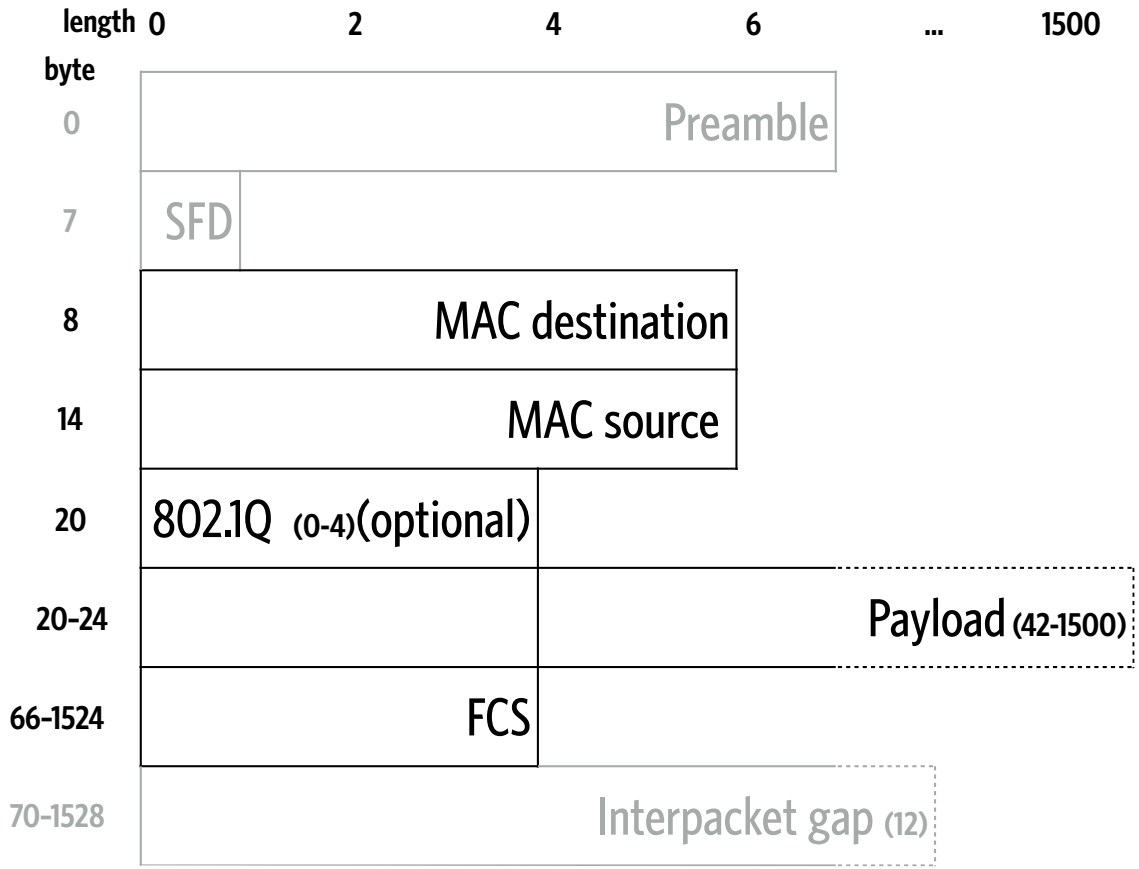
- Responsibility:
  Transmission of packets between hosts connected by a physical link

- Solves addressing:
  Media Access Control (MAC) addresses

- Solves (partly) reliability:
  Checksums (e.g., IEEE 802.3 use of CRC)

- Performance gains by using switches

802.3: Ethernet (frame)

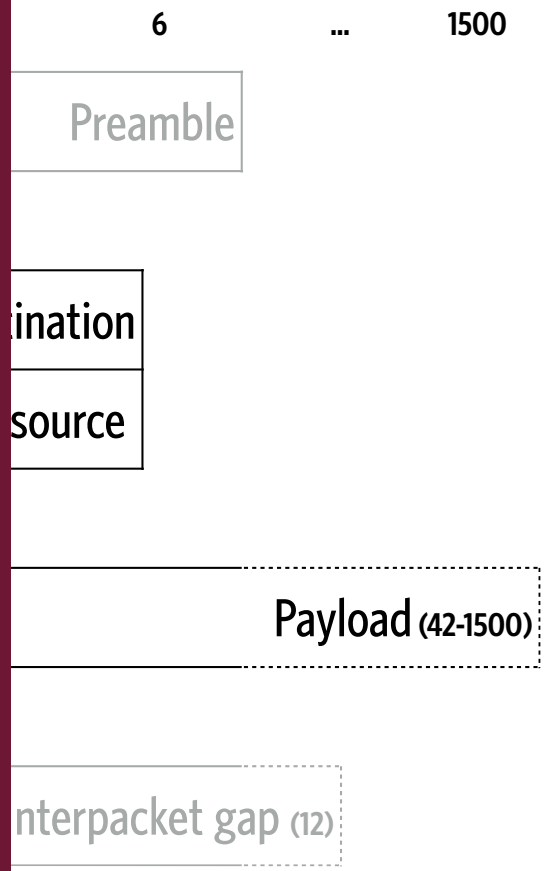


- MAC destination, source: "To", "From"
  (May be broadcast)
- Payload (actual contents) up to 1500 bytes
- Frame Check Sequence:
  32-bit Cyclic Redundancy Check checksum
  Detects error bursts < 32 bit
- Check failed => Frame dropped

Let's break it!

Confidentiality
Integrity
Availability
Accountability

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

Attacks on the data link layer

- **Tampering** (integrity)
  CRC is cryptographically weak;
  you won't detect my change
  (We'll do this in the Crypto-lecture.)

- **Message injection/MAC spoofing**
  I can put arbitrary messages on the wire

- **Eavesdropping** (confidentiality)
  Frames are broadcast;
  I can see them even if they aren't for me.
  Switches: MAC flooding

MAC flooding

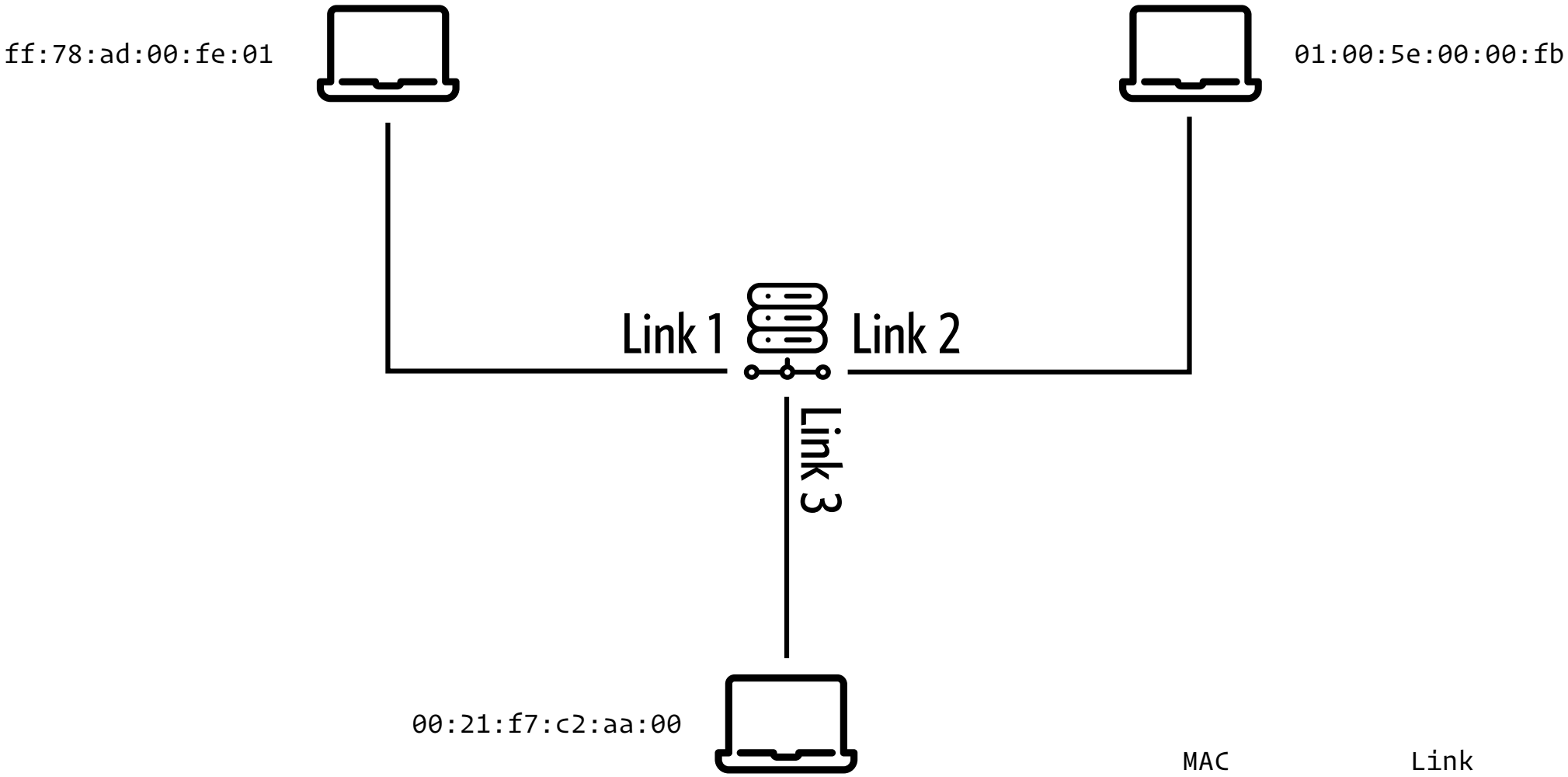
- Transmit enough fake frames with new src addresses that the switches' table contains no actual addresses

- Switch must now broadcast all frames

MAC flooding

- Transmit enough fake frames with new src addresses that the switches' table contains no actual addresses

- Switch must now broadcast all frames



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

Network layer

Network layer

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

IP Operations

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

IP Operations

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

Let's break it!

Confidentiality
Integrity
Availability
Accountability

IP Operations

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

Let's break it!

Confidentiality
Integrity
Availability
Accountability

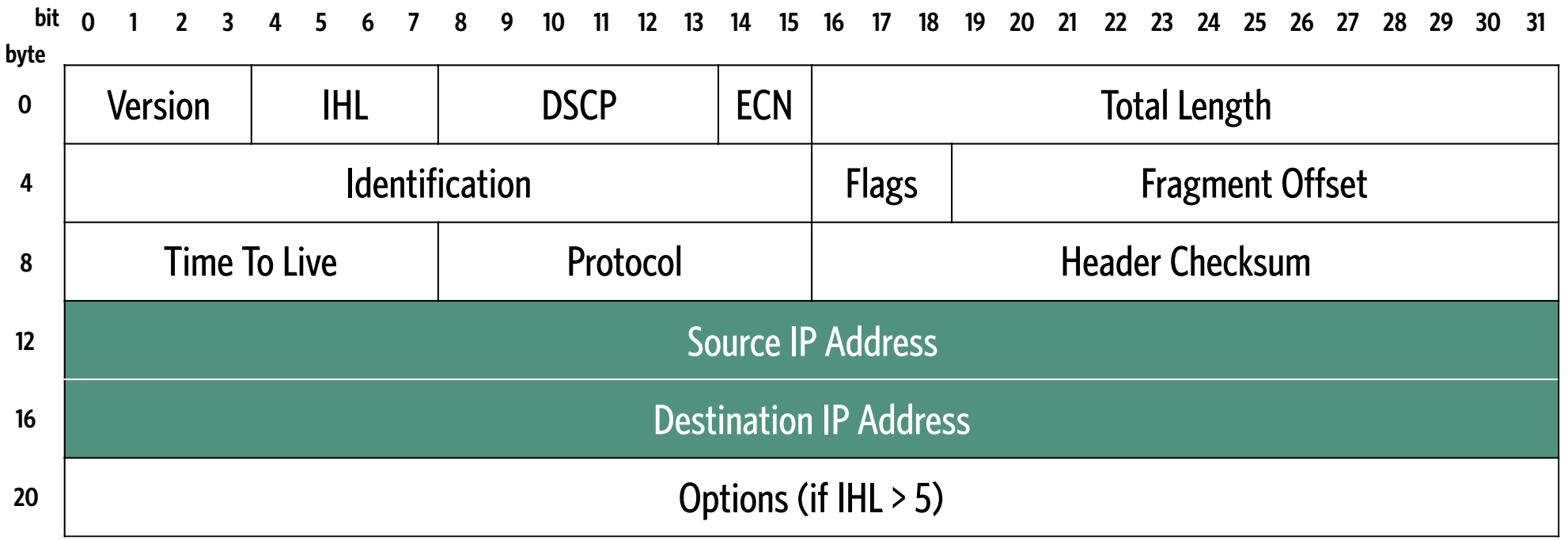
Spoofing

- **ARP Cache Poisoning** (?)
  (aka ARP spoofing, ARP Poison Routing)
  Spoof ARP "is" packets redirecting traffic for other IP to my machine.

- **IP Spoofing**
  "*What's really interesting is that these people will send a tube of live ants to anyone you tell them to*"

- **DHCP Spoofing**

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)

Local denial-of-service attacks

- ARP Cache poisoning
- DHCP Starvation

Remote denial-of-service attacks

- Live: http://map.norsecorp.com/#/
- Ping flooding
- IP fragmentation attack
- Distribute the attack from many attacking machines for maximum effect

IPv4 Fragmentation Attacks

- Overlapping fragments
  E.g., teardrop.

- Fragmentation buffer filling/overrun

- Too many fragments
  E.g., Rose attack: send first and last bytes of large volumes of 65k packet

Transport layer

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

TCP

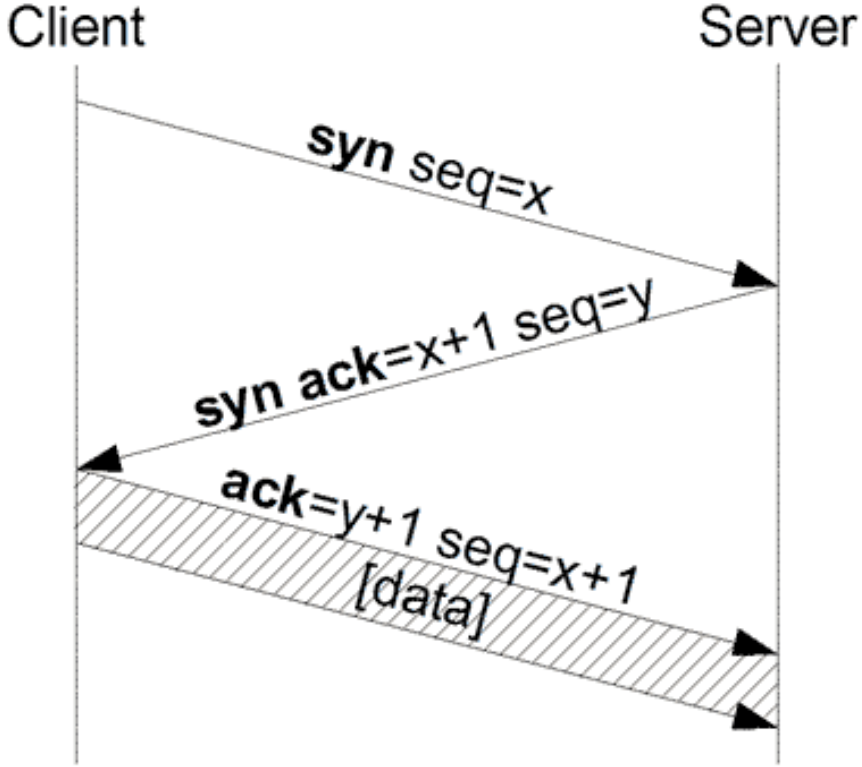
- Connection-oriented, reliable, streaming protocol.
- Achieved by message/acknowledgment sequence numbers, timeouts.
- Protocol specified as a fairly complex state machine

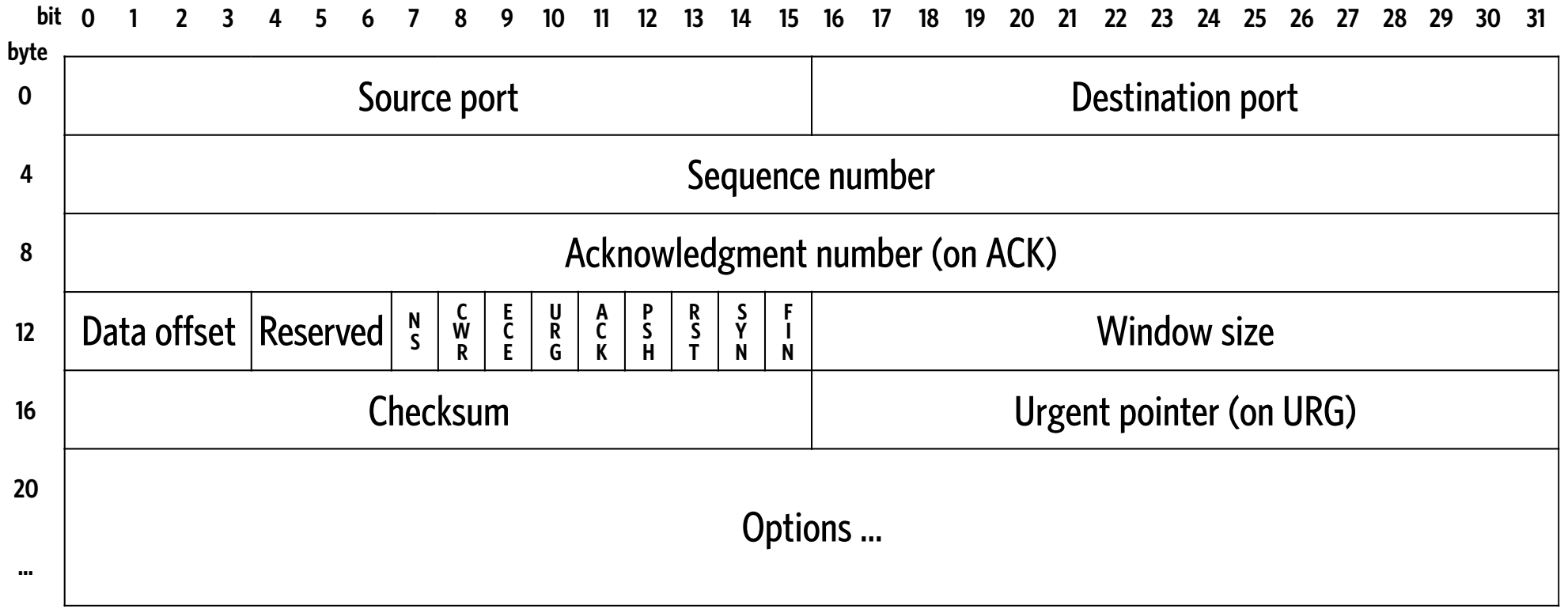
Let's break it!

Confidentiality
Integrity
Availability
Accountability

Connection setup

The 3-way handshake





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

TCP sequence prediction attack

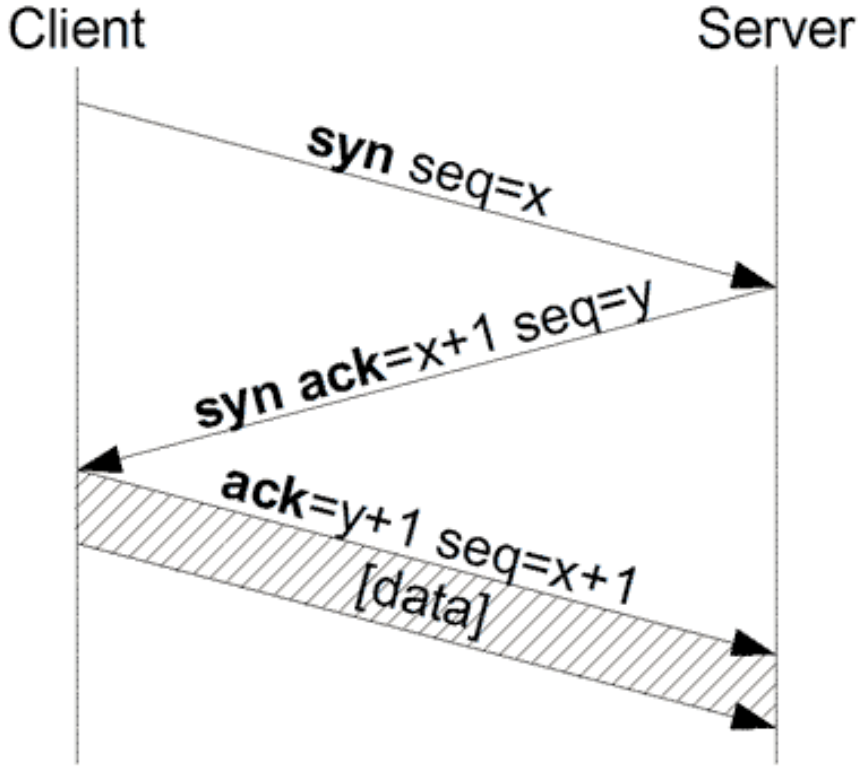
- Suppose we want to hi-jack a connection from host A to B.
- TCP Sequence numbers are sent in cleartext (eavesdropping)
- Listen to traffic from B. Kill B's end of the connection (e.g., next slide)
- Spoof TCP packets to A.

TCP RESET attack

- Spoof TCP packet with RST set to 1.
- Remote system should drop connection
- Bypassing IDS/Firewall may require sequence prediction.

TCP SYN flood

The 3-way handshake



- Send large volumes of initial SYN message
- Very cheap
- Ties up buffers at receiving end

Application layer

Domain-name System

Domain names

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

Domain names

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

Let's break it!

Confidentiality
Integrity
Availability
Accountability

Domain names

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

Let's break it!

Confidentiality
Integrity
Availability
Accountability

DNS attacks

- **DNS reply spoofing/hi-jacking**
  You're not talking to your bank, you're talking to me.

- **DNS reflection/amplification attack**
  I spoof DNS queries from you.
  You'll receive large numbers of large responses

- **DNS tunneling**
  I use DNS packets to shuffle data through your firewall

DNS attacks

- **DNS reply spoofing/hi-jacking**
  You're not talking to your bank, you're talking to me.

- **DNS reflection/amplification attack**
  I spoof DNS queries from you.
  You'll receive large numbers of large responses

- **DNS tunneling**
  I use DNS packets to shuffle data through your firewall

Bonus round

- Telnet (remote access), FTP (file transfer protocol)
- Neither use encryption

Let’s break it!

Attacks on FTP, Telnet

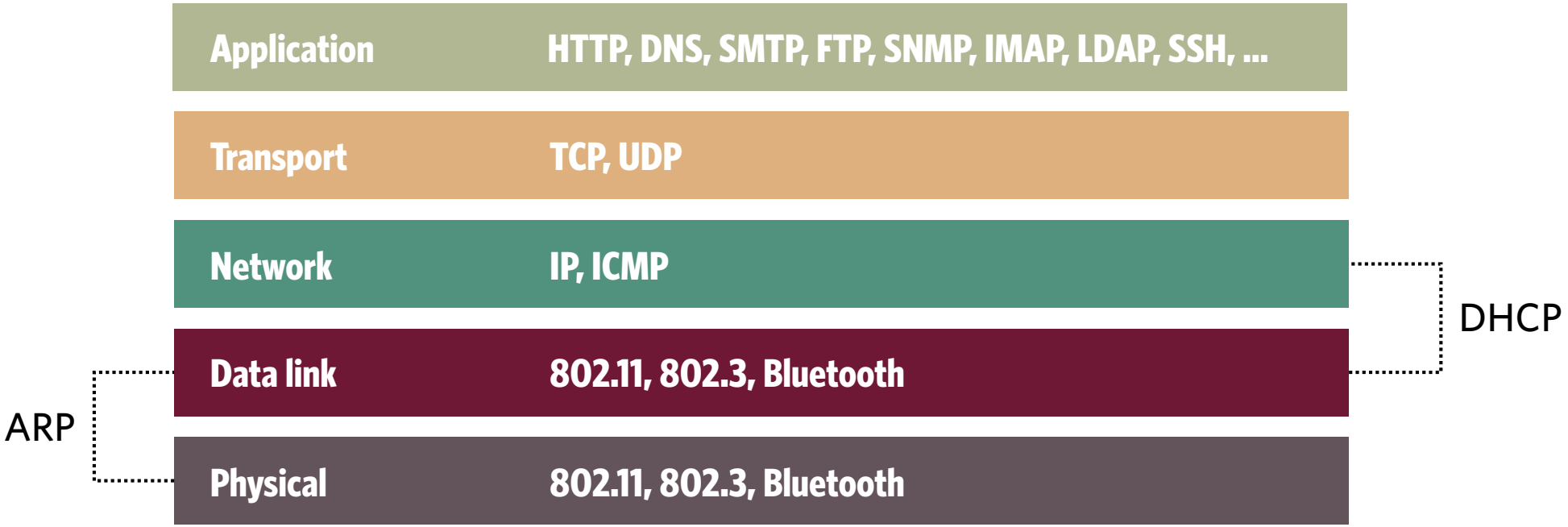
- Login credentials sent in cleartext.
- Adversary may obtain username/password merely by eavesdropping.
- Adversary may obtain session traffic (telnet commands, file contents) also merely by eavesdropping.

Hypertext Transport

We'll get back to this

Protocol layers

All broken


Application
HTTP, DNS, SMTP, FTP, SNMP, IMAP, LDAP, SSH, ...
Transport
TCP, UDP
Network
IP, ICMP
Data link
802.11, 802.3, Bluetooth
Physical
802.11, 802.3, Bluetooth
DHCP
ARP

Port scanning

Port scanning

Port scanning

- How much can the adversary learn about your system using only network traffic?

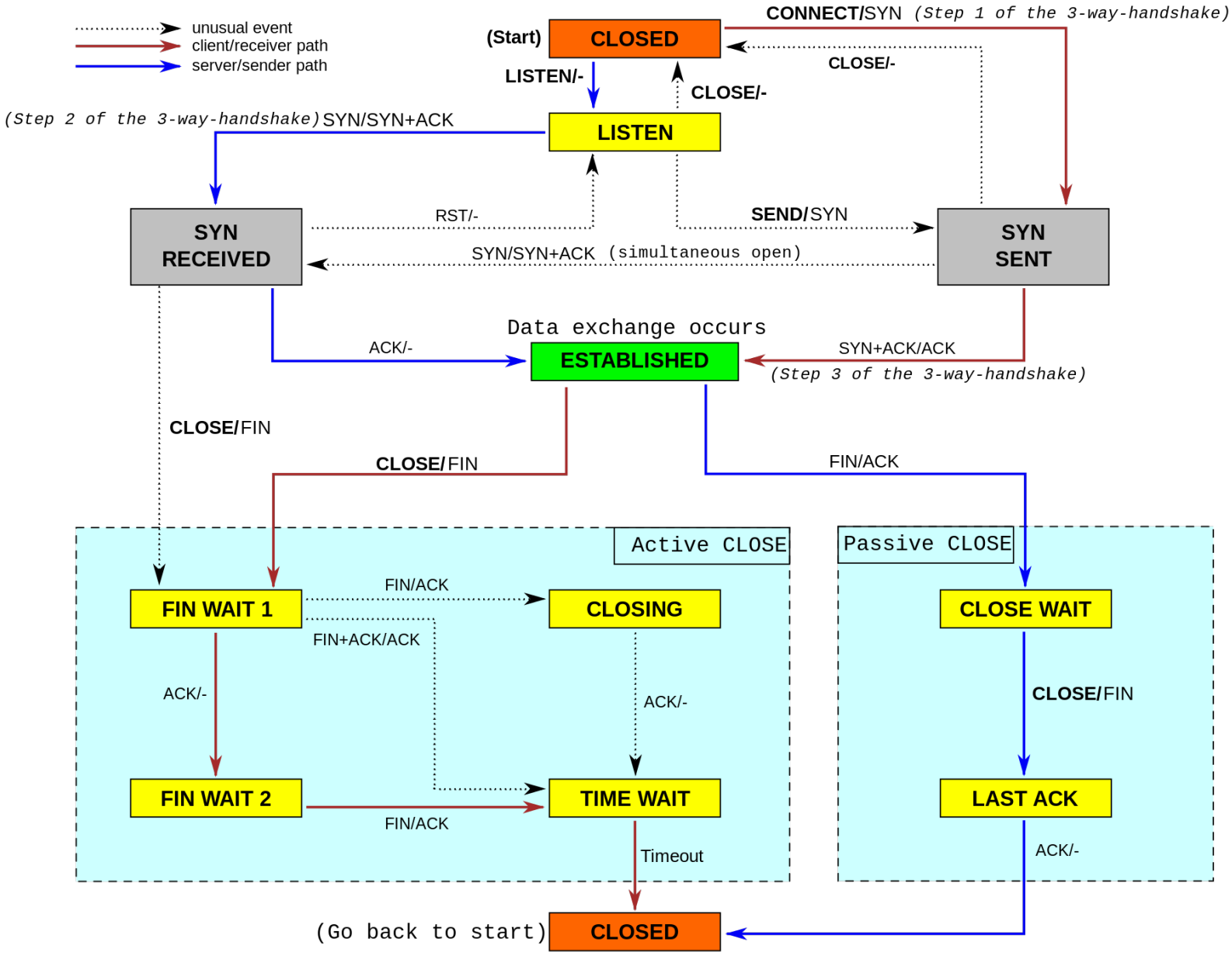
```
> nmap nmap -p 1-65535 -T4 -A -v www.itu.dk

Starting Nmap 7.40 ( https://nmap.org ) at 2017-02-03 16:31 CET
...
Scanning www.itu.dk (130.226.142.6) [4 ports]
Completed Ping Scan at 16:31, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 16:31
Completed Parallel DNS resolution of 1 host. at 16:31, 0.00s elapsed
Initiating SYN Stealth Scan at 16:31
Scanning www.itu.dk (130.226.142.6) [65535 ports]
Discovered open port 443/tcp on 130.226.142.6
Discovered open port 80/tcp on 130.226.142.6
Discovered open port 22/tcp on 130.226.142.6
Increasing send delay for 130.226.142.6 from 0 to 5 due to 37 out of
92 dropped probes since last increase.
Completed SYN Stealth Scan at 16:52, 1254.04s elapsed (65535 total
ports)
Initiating Service scan at 16:52
Scanning 3 services on www.itu.dk (130.226.142.6)
Completed Service scan at 16:52, 12.46s elapsed (3 services on 1
host)
Initiating OS detection (try #1) against www.itu.dk (130.226.142.6)
Initiating Traceroute at 16:53
Completed Traceroute at 16:53, 3.02s elapsed
```

```
Nmap scan report for www.itu.dk (130.226.142.6)
Host is up (0.0064s latency).
rDNS record for 130.226.142.6: asterix.itu.dk
Not shown: 65532 filtered ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 5.3 (protocol 2.0)
| ssh-hostkey:
|   1024 09:6c:46:3a:19:47:1c:2d:b7:8b:75:1a:72:96:af:89 (DSA)
|_  2048 2f:ad:c4:86:59:33:45:12:fd:10:bd:78:f1:8e:ce:79 (RSA)
80/tcp  open  http     Apache httpd 2.4.18 ((Red Hat) OpenSSL/1.0.1e-
fips)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.18 (Red Hat) OpenSSL/1.0.1e-fips
|_http-title: Did not follow redirect to https://www.itu.dk/
443/tcp open  ssl/http Microsoft IIS httpd 7.5
...
|_http-server-header: Microsoft-IIS/7.5
...
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.10, Linux 2.6.32 - 3.13, Linux 3.10,
Linux 3.4 - 3.10
Uptime guess: 20.756 days (since Fri Jan 13 22:45:07 2017)
Network Distance: 4 hops
TCP Sequence Prediction: Difficulty=257 (Good luck!)
IP ID Sequence Generation: All zeros
```

Scan outcome

- List of (Transport layer) ports and their states:
  - **Open:** Target accepts connections
  - **Closed**: Accessible, but not accepting connections. (Probably no application is listening.)
  - **Filtered**: Not accessible.
  - **Unfiltered**: Open or closed.


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

Scan types

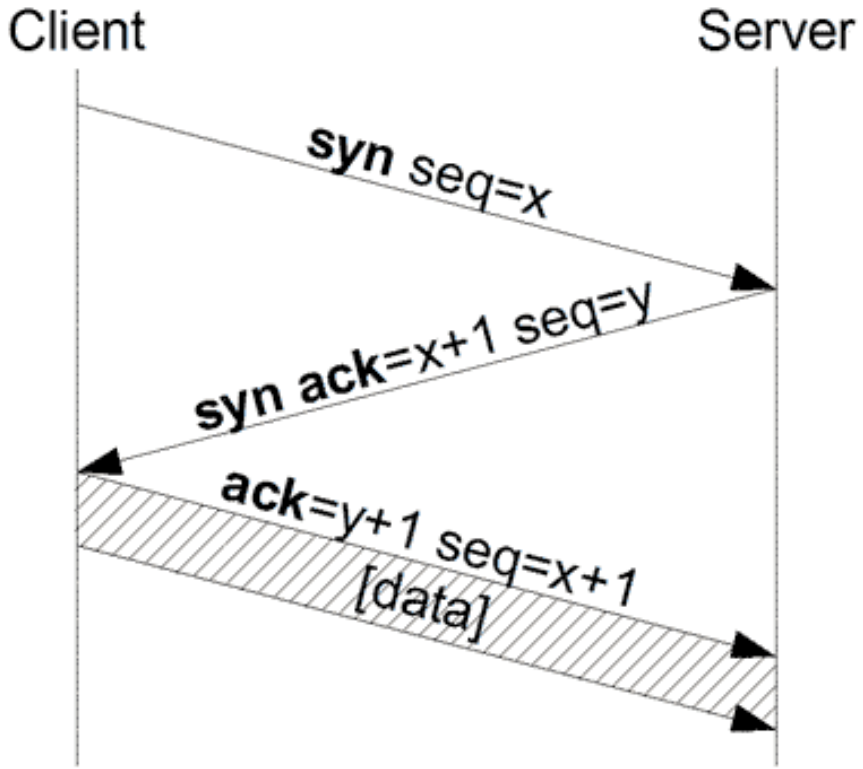
- Ping scan.
  Fast'ish. Noticeable but innocuous.

- TCP connect scan.
  Establish connection
  Slow. Noticeable.

- TCP SYN scan.
  Send SYN, wait for SYN+ACK.
  Faster.
  Noticeable to OS, not to application.

- TCP ACK scan. Send ACK, wait for RST. Successful scan results in "unfiltered"

- Stealth scan, idle scan



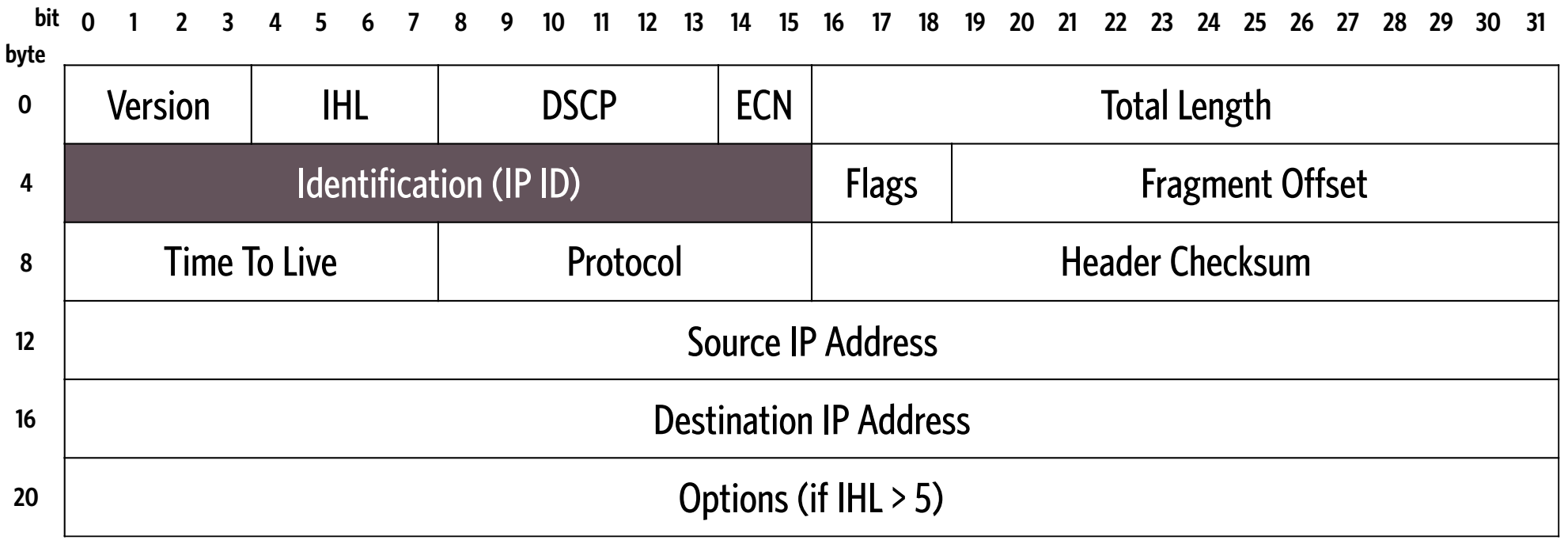
Stealth Scan

- Exploits RFC 793 details to distinguish OPEN/CLOSED ports:
- *"... if the [destination] port state is CLOSED .... an incoming segment not containing a RST causes a RST to be sent in response."*
- *"... if you [receive on an OPEN connection a packet without SYN, RST, or ACK], drop the segment, and return."*
- I.e., a packet with SYN, RST, and ACK set (the christmas tree) will:
  - Result in RST on CLOSED port
  - Result in no response on OPEN port
- Stealth: Confounds some packet filtering firewalls.

Idle scan (1)

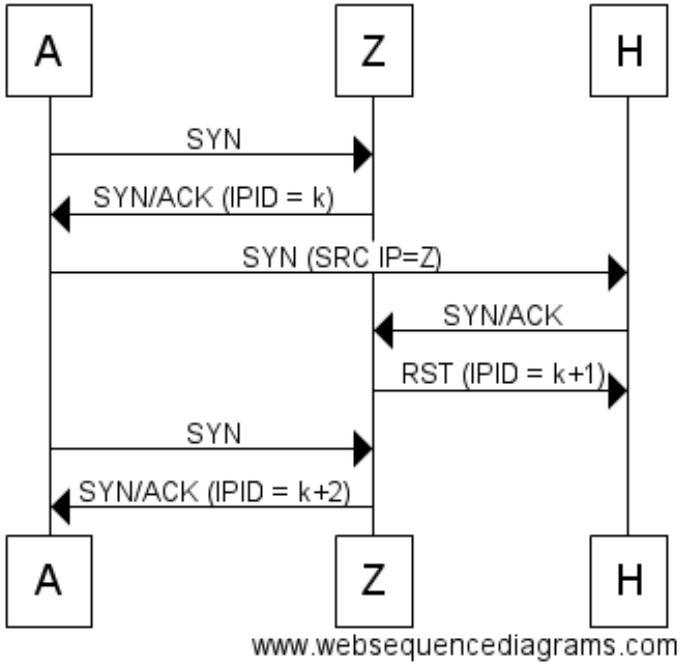
- Scan of host H with no traffic between you and H.
- Requires host Z (zombie) with
  - no or very little traffic
    (E.g., network-connected printer outside office hours.)
  - incrementing IP fragment identifier on each IP packet
    (Many network stacks do this.)

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 (IP ID)
Flags
Fragment Offset
8
Time To Live
Protocol
Header Checksum
12
Source IP Address
16
Destination IP Address
20
Options (if IHL > 5)

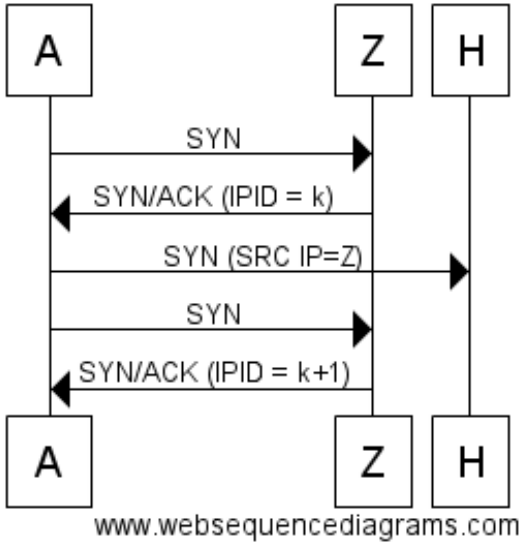
Idle scan

Port open



Note IPID=k+2

Port closed



Note IPID=k+1

Idle scan

- Probe IP ID of Z with SYN scan.
- Forge SYN packet from Z to H.
- Probe IP ID of Z again.

Legality

- No-one has been convicted of port scanning in Denmark.

- It is rather rude, though—think peeking in through people's windows.

Counter-measures

Firewalls

- Packet filters
  Reject/accept packets based on
  src/dest IP/port/MAC

- Stateful filters
  Reject/accept packets based on connection state (e.g., TCP state)

- Application layer
  Reject/accept packets based on protocol (e.g., HTTP)

This helps,
but is not enough

Summary

Plan

- Attacks on the network stack
- Port scanning
- Firewalls

Thank you!

- See learn-it for exercises etc.
- NB! Peergrade exercise
- NB! Mandatory quiz
- Questions?

Credits

TCP diagram:
Wikipedia

Icons designed by
Gregor Cesnar,
FlatIcon