TCP/IP

»Bus Systems«
Karlsruhe University of Applied Sciences

Prof. Dr. Th. Leize
Winter 2024/25

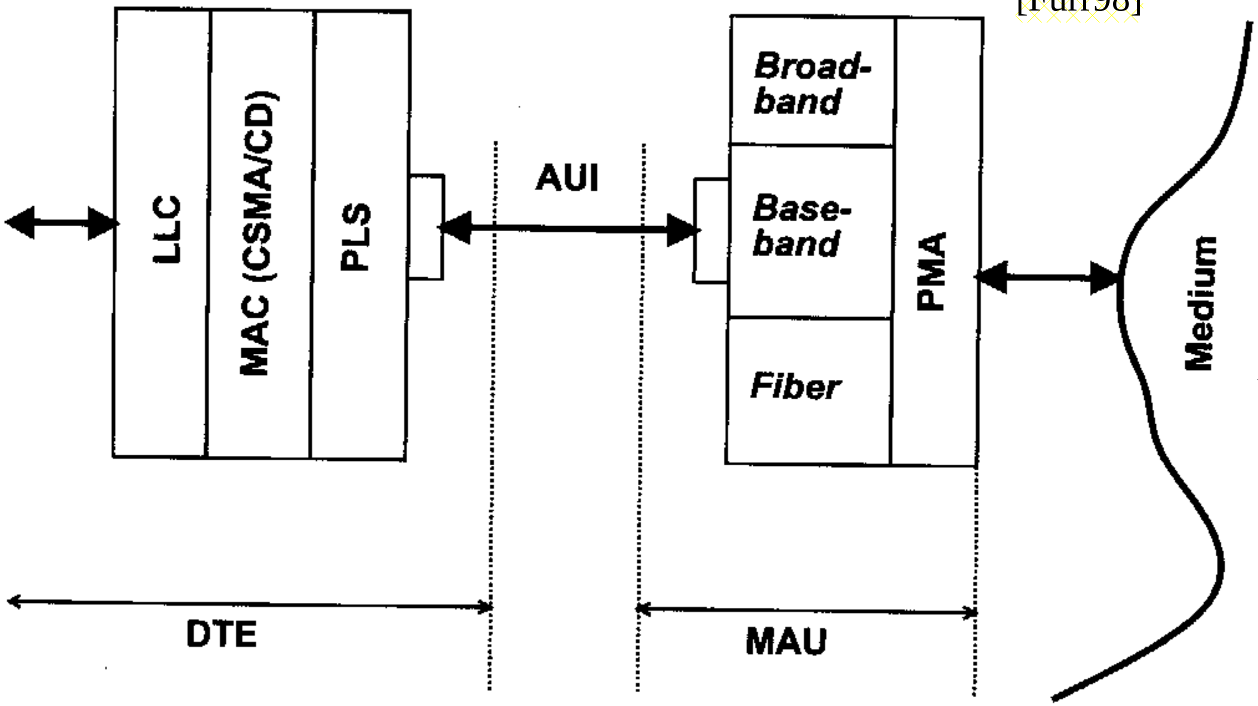
Contents

Ethernet - History

- ca. 1970 increased number of point to point communications -> Lots of wires
  1969: first experimental net: ARPANET (Advanced research projects agency) with 4 nodes. eMail and ftp!
- 1973: Xerox invents Ethernet with 100 nodes. Afterwards enhancements and new media. Start with 3MBit/s, but short time later 10MBit/s
- 1974: Design of TCP/IP
- 80er: Token Ring (IBM) and others, not compatible with Ethernet
- 1983: TCP/IP in ARPANET. (1000 nodes) DNS new. MILNET.
- Level model: Network layer, IEEE 802, currently most frequently used: IEEE 802.3 (1990)
- 1990: WWW, 1993: First internet browser "Mosaic" (CERN)

Ethernet Connection (Scheme)

[Furr98]



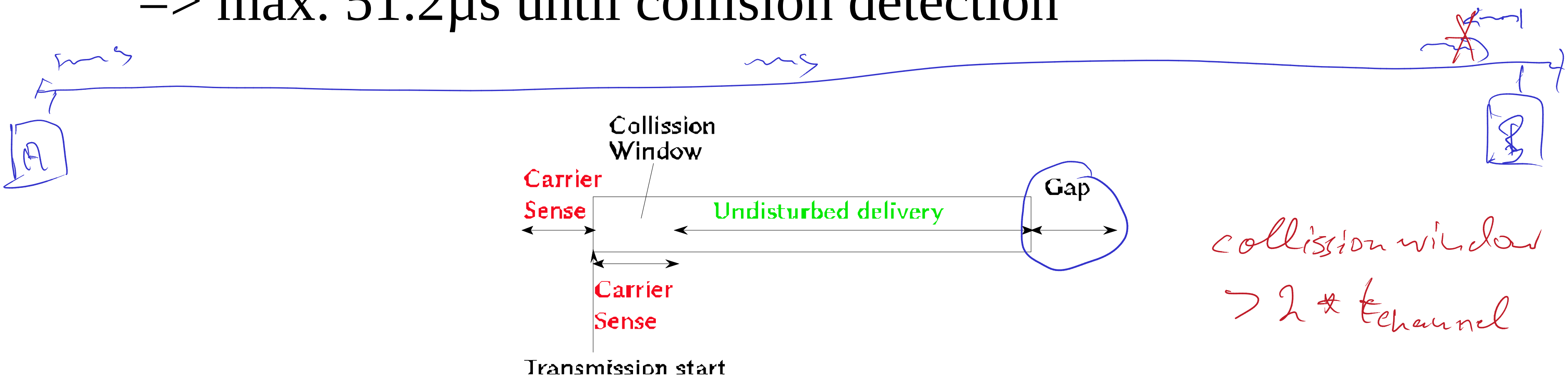
DTE: Data Terminal Equipment
LLC: Logical Link Control
MAC: Medium Access Control
PLS: Physical Signalling
AUI: Attachment Unit Interface
MAU: Medium Access Unit
PMA: Physical Medium Attachment
MDI: Medium Dependent Interface

In previous times: Bus with MAU, then max. 50 m connection wire to the DTE. Today all these parts are integrated in 1 chip. Compare to the things we discussed concerning signals on wires.

CSMA/CD

Carrier Sense Multiple Access with Collision Detection

- Max. signal time between two nodes: 25.6µs
  => max. 51.2µs until collision detection

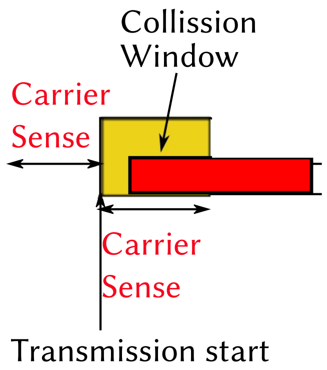


collission window
$> 2 * t_{channel}$

Algorithm:
- Listen, if someone else sends (carrier sense)
- no: Start sending
- Continue with carrier sense.
- At the end keep quiet for at least 9.6µs

CSMA/CD: Collisions

- Would be nice if there were no collisions. But due to statistics there will be collisions in any case.
- Does a node detect a collision it immediately send 4 to 6 bytes with ones ("jamming burst"). Everybody stops sending. -> quiteness.



Random waiting time:

n^th^ try (n =1 … 9):

Random number r: 0 … 2^{n}-1

$t_w = \underbrace{9.6\mu s}_{\text{gap}} + r * \underline{51.2\mu s}$

10^{th} to 15^{th} try: Repeat with r: 0 … 1023

If even the 15^{th} start fails the data package is „dropped“.

Example: 2 devices, $n = 1$

~~How~~ What is the probability of another collission?

A	B	
0	0	↯
0	1	✓
1	0	✓
1	1	↯

$\frac{1}{2}$!

2 devices, $n = 2$. probability?

$\frac{1}{4}$

A	B	
0	0	↯
0	1	✓
0	2	✓
0	3	✓

A	B	
1	0	✓
1	1	↯
1	2	✓
1	3	✓

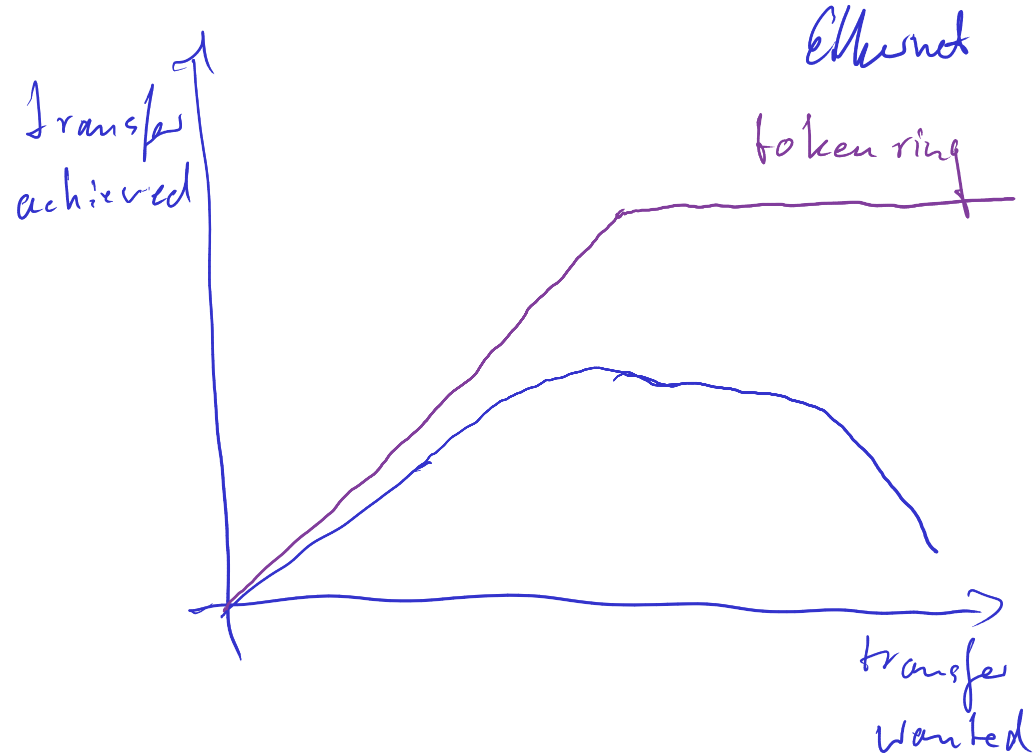
A	B	
2	0	✓
2	1	✓
2	2	↯
2	3	✓

A	B	
3	0	✓
3	1	✓
3	2	✓
3	3	↯

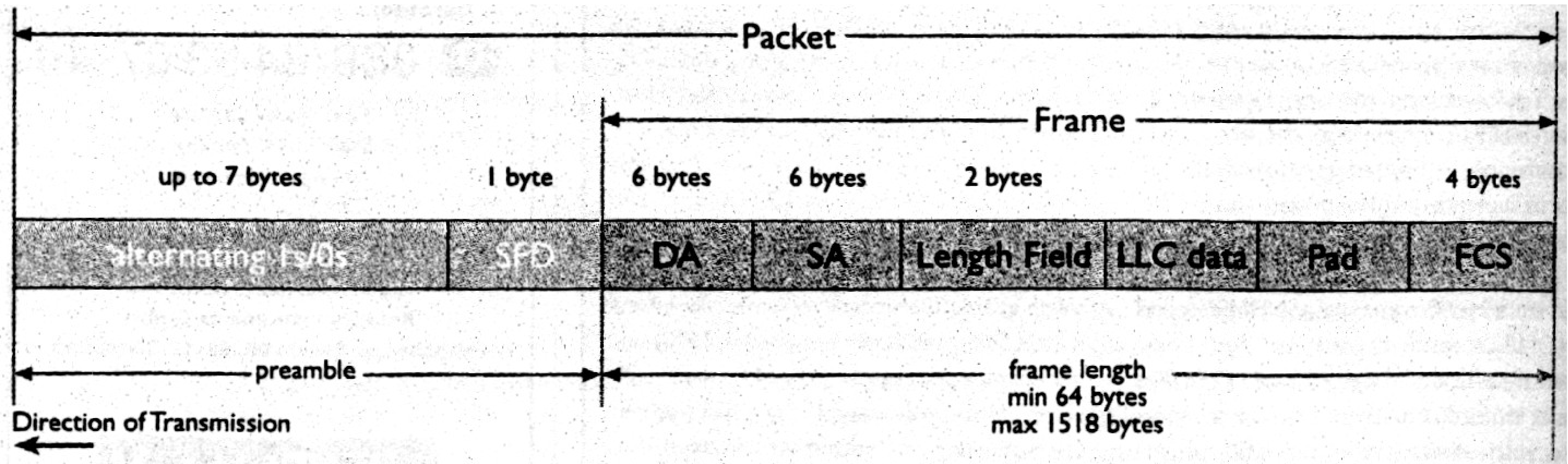
3 devices, $n = 1$

A	B	C	
0	0	0	↯
0	0	1	↯
0	1	0	↯
0	1	1	✓
1	0	0	↯
1	0	1	✓
1	1	0	✓
1	1	1	↯

$\frac{5}{8}$



Ethernet-Frame



SFD = Start of Frame Delimiter
DA = Destination Address
SA = Source Address

LLC = Logical Link Control
FCS = Frame Check Sequence
(also called Cyclic Redundancy Check, or CRC)

Typ	Protokoll
0x0800	IP
0x0806	ARP

[DesEle]

Ethernet Frame

- Structure of an ethernet datagram
  - Preamble:
    - Up to 7 bytes alternating 0/1 (synchronisation)
      (some of these bits will be lost if the signal passes „repeater“s)
    - SFD: 1 byte. "start of frame delimiter" 101010**11**
  - Frame:
    - DA: 6 bytes. "Destination Address", MAC-address.
      FFFFFFFFFFFF is broadcast: Message to everybody.
    - SA: 6 bytes. "Source Address"
    - LEN: 2 bytes. IEEE 802.3: Length of the data field (46-1500)
      Before 802.3: Type: 0x800: IP, 0x806: ARP
    - DATA: up to 1500 bytes: Data and protocol info of higher layers
    - Pad: 0..46 bytes. Padding bytes if the length is less than 46 bytes.
    - FCS: 4 bytes. "Frame check sequence" (CRC "cyclic redundancy check")

Command line: Windows: ipconfig /all others: ifconfig

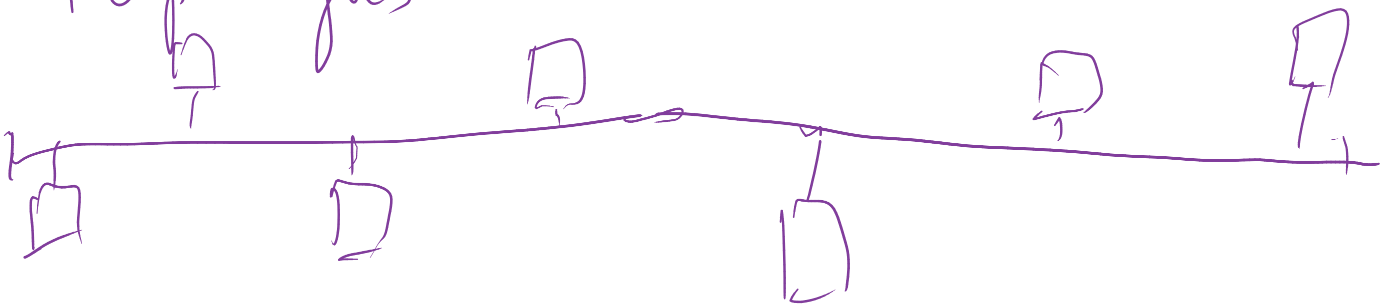
MAC-Address

Q: Who did your network cards? Q: Who has ID ∅?

- Worldwide unique address. The bits are devided into groups:
  - #47,#46: reserved
  - #45 .. #24: Manufacturer-ID (4.194.302 manufacturers)
  - #23.. #0: serial number of the manufacturer (16.777.214 adapters)
- Bits 46,47 have the following meaning:
  - Destination address:
    - #47=1: group address, =0: individual address
    - #46=1: local address, not IEEE conformant; =0: global, unique, IEEE
  - Source address:
    - #47=0 defined
    - #46=1: local address, not IEEE conformant; =0: global, unique, IEEE

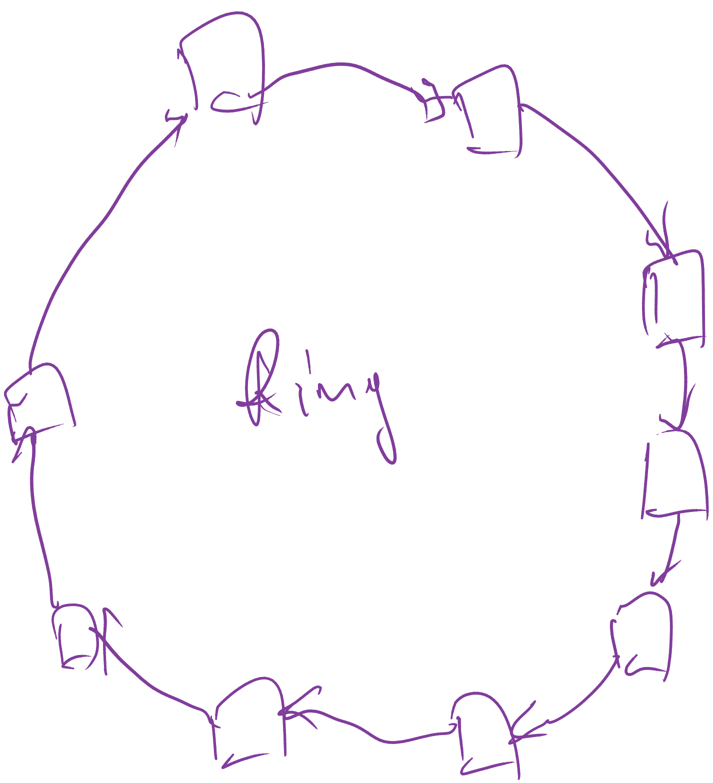
With #46=1 you can define local, private numbers that need not be unique with respect to all others in the world.

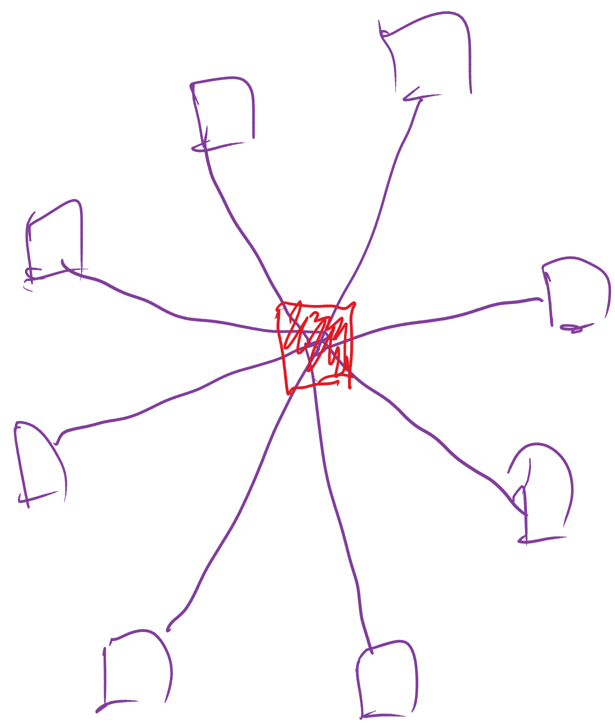
Topologies



line / bus

passive star
active

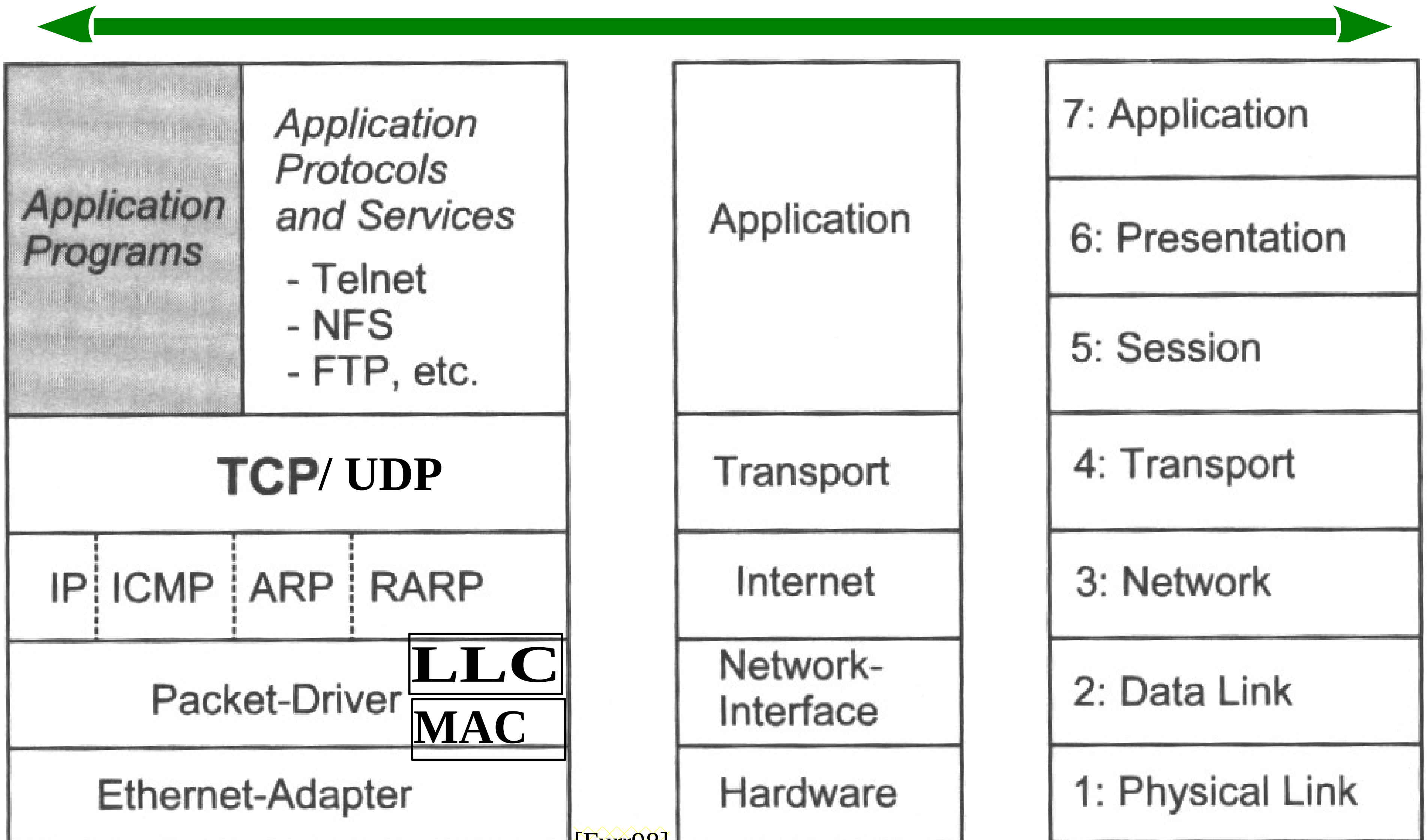




TCP/IP

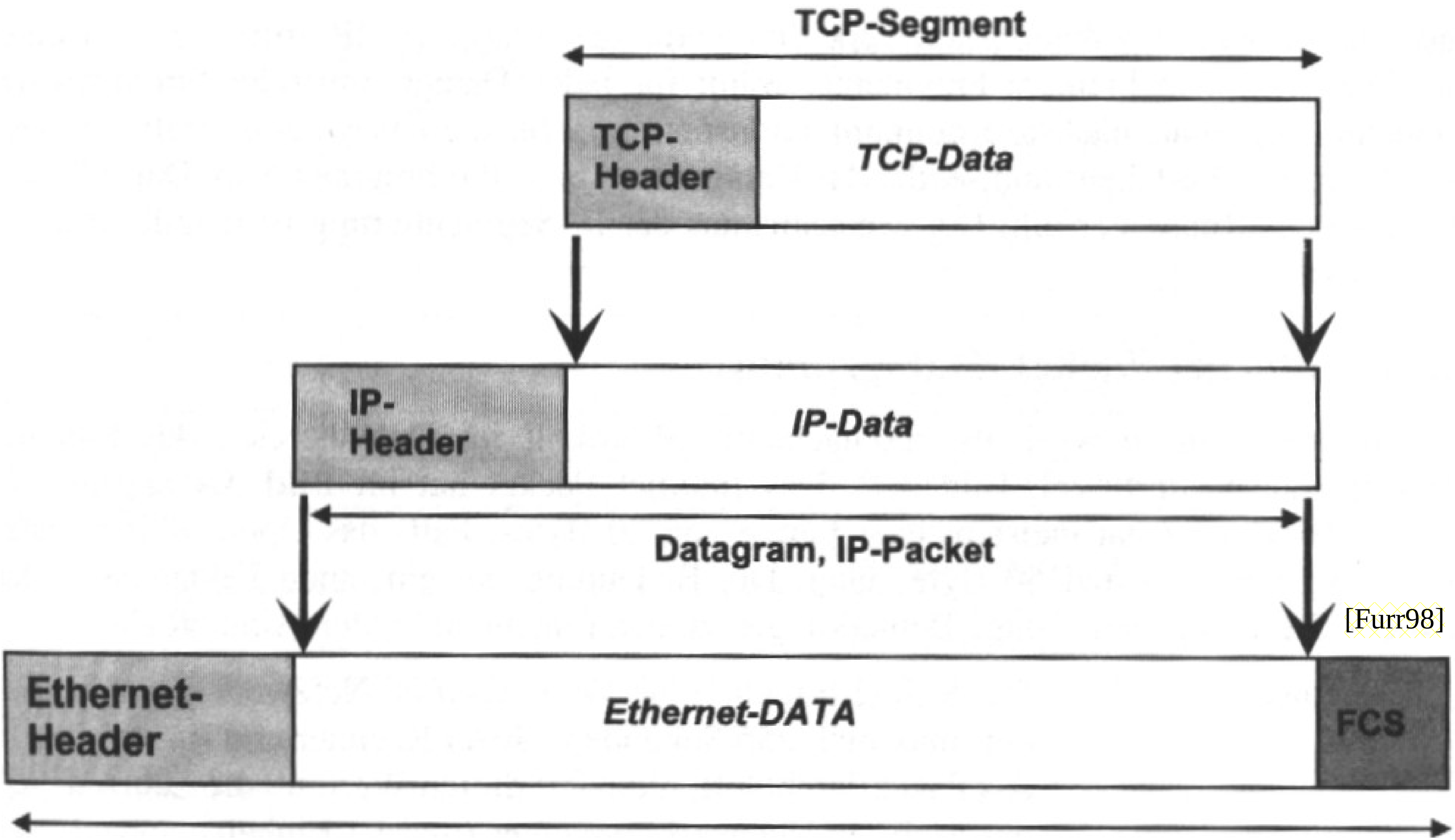
- TCP/IP is a whole family of protocols
- History:
  - Early 1970s: US defense government and ARPA start with some research. These requirements were defined:
    - Usage of different transmission pathes and media.
    - Independent of a specific operating system or hardware of any manufacturer.
    - Standardised functions and protocols.

ISO/OSI – Layers and TCP/IP



[Furr98]

Datagrams and Headers



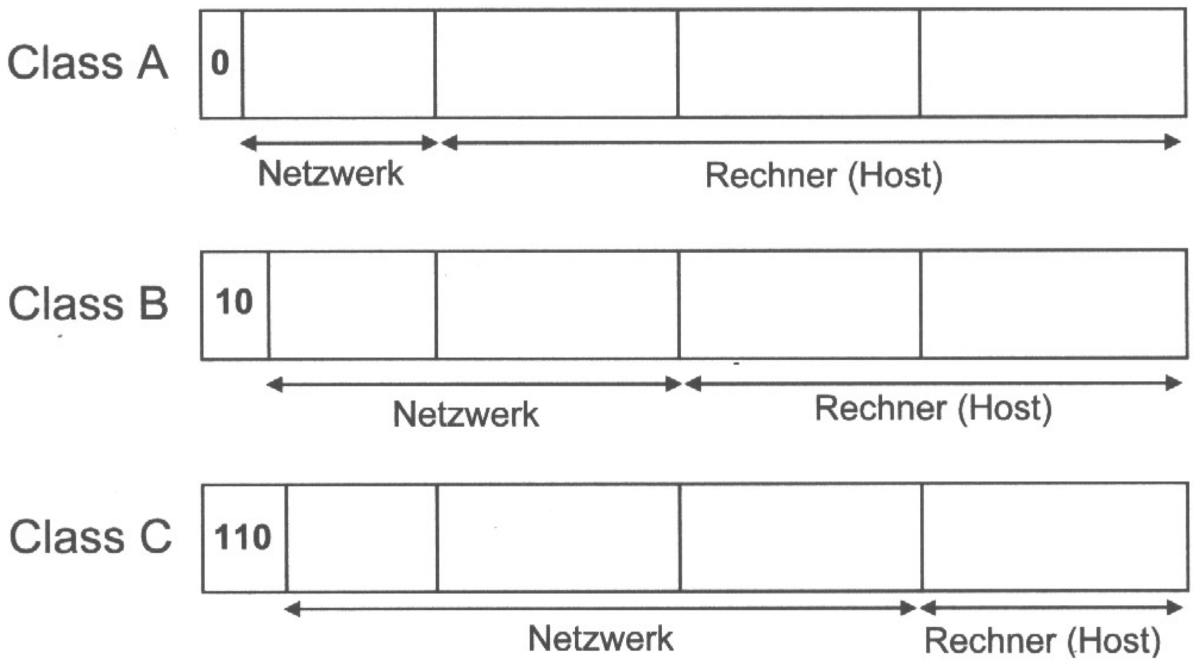
IP: "Internet Protocol"

- Here: Version 4, most widely used today.
  New: Version 6: IPnG ("next Generation")
- **Network layer.**
- Contains functionality for
  - Addressing of nodes
  - Routing
  - Fragmenting / defragmenting of the datagrams IPv4 only
- IP is ***not able*** to:
  - Guaranty a reliable connection (reliability depending on layers below, e.g. Ethernet)
  - No acknowledgements are sent

IP Datagrams

- Structure of a IP datagram (multiple of 32 bits)
  - version, 4 bits, e.g. "4"
  - header length in multiples of 4 bytes, 4bit (without options: 5; otherwise 5.. 15)
  - service-Type, 8 bits
  - length, 16 bit, Header+Data
  - id, 16 bit, for segmentation
  - flags, 3 bit, -"-
  - fragment offset, 13 bit, -"-
  - life time, 8 bit
  - protocol, 8bit
  - checksum of the header, 16bit
- IP source address, 32 bit
- IP destination address, 32bit
- options, 0..320bit
- padding, 0..32, filling bits to a multiple of 32 bits.
- data, 0 .. 65.516 bits

IP Adresses



[Furr98]

- IP addresses "dotted decimal": e.g.: 193.196.87.145
- class A: nets: 1.x.x.x - 126.x.x.x (126 nets à 16.646.144 hosts)
- class B: nets: 128.1.x.x - 191.254.x.x (16.256 nets à 65.024 hosts)
- class C: nets: 192.0.1.x - 223.255.254.x (2.080.768 nets à 254 hosts)
- class D, class E: (Starting bits: 1110 / 1111) reserved

Subnet Mask

- Subnetwork-masks also are dotted decimal.
- Defines logical subnets.
  (You will need routers between subnets. Enables big nets to be devided in separate subnets.)
- Bit: 1 means this is a bit of the net address, 0 is a bit of the host address.
- class A: 255.0.0.0 /8
- class B: 255.255.0.0 /16
- class C: 255.255.255.0 /24
- Example: Huge B-net is devided into subnets by mask 255.255.192.0
  (11111111 11111111 11000000 00000000)
  This results in th following subnet start addresses: 190.136.0.0, 190.136.64.0, 190.136.128.0, 190.136.192.0
  Hosts in the second subnet have addresses between IP 190.136.64.1 and 190.136.127.254

Given is a class B network: 190.136.x.y

subnet mask: 255.255.0.0

Task: Divide into 4 subnets. => 2 additional network bits.

1111 1111 . 1111 1111 . 1100 0000 . 0000 0000 /18

255 . 255 . 192 . 0

Question: what are the IP ranges for these subnets?

	from IP	to IP
00	190.136.0.0	190.136.63.255
01	190.136.64.0	190.136.127.255
10	190.136.128.0	190.136.191.255
11	190.136.192.0	190.136.255.255

Given: class C network 192.168.1.x subnet 255.255.255.0

Task: Please divide into 2 subnets. What is the new subnetmask?

What are the IP ranges?

new subnetmask: 255.255.255.128

	from	to
0	192.168.1.0	192.168.1.127
1	192.168.1.128	192.168.1.255

Given network: 10.10.x.y subnetmask 255.255.0.0

Please divide into min. 6 subnets.

- What is the new subnet mask?
- What are the IP ranges? Please use the numerically lowest possibilities.

=> 3 additional bits

=> New subnet mask: 255.255.224.0

	from	to
0 0 0	10.10.0.0	10.10.31.255
0 0 1	10.10.32.0	10.10.63.255
0 1 0	10.10.64.0	10.10.95.255
0 1 1	10.10.96.0	10.10.127.255
1 0 0	10.10.128.0	10.10.159.255
1 0 1	10.10.160.0	10.10.191.255
1 1 0	10.10.192.0	10.10.223.255
1 1 1	10.10.224.0	10.10.255.255

Special IP Addresses

IPv6: 128 bit
hexadecimal
:: insert Ø here

- 127.x.x.x: Local addresses inside the same node. They are not transmitted to the bus. Loopback address: 127.0.0.1 IPv6 ::1
- 255: Broadcast addresses better: highest address in network.
  Used to send broadcast messages to all stations of this net.
  e.g.: 126.255.255.255 or 239.1.2.255
- 0: local
  Leading zeros define „this network“
  e.g. 0.0.0.4 host 4 inside this network.
- RFC 1918 defines reserved, local addresses, that are not routed to the internet. For usage in local intranets. Example: class A: 10.x.x.x

IP - Fragmentation

- The MTU (Max. Transfer Unit) is defined by the layer 2 used: Token Ring (16MBit/s) 17914 Bytes, Token Ring (4Mbit/s) 4464 Bytes, Ethernet 1500 Bytes, IEEE 802.3 1492 Bytes, X.25 576 Bytes. => Fragmentation needed
- Flag: 3 bits: Highest bit not used yet, middle: "Don't fragment" bit, lowest: "more-flag" (last datagram contains 0)
- In an unfragmented datagram the more flag is 0 as well as the fragment offset.
- Do not fragment if the don't-fragment flag is not set.
- Fragmentation means division into full IP datagrams
  - Information set: More, Fragment-Offset, Options copied, checksum calculated, length defined

ICMP
Internet Control Message Protocol

- Is on top of IP as well as part of IP definition.
- Used for transmission of errors.
- Not reliable since it uses IP.
- Complete IP header, ICMP information in data bytes.
- There are no errors about ICMP-messages to avoid endless loops.

IPv6: NDP Neighbour Discovery Protocol

ARP
Address Resolution Protocol

Only IPv4

- Is not routed.
- Looks for the MAC (Ethernet)-address for a given IP address.
- Does a node not find the MAC address in it's ARP cache, it sends a ARP broadcast to find the MAC address
- Every station in this subnet listens to these broadcasts. If one node has this IP address, it answers this message and delivers it's MAC address. The asking station adds the new information to it's cache.
- Switches and bridges are layer 2 devices and therefore transmit these broadcasts.

Each device keeps an arp table:

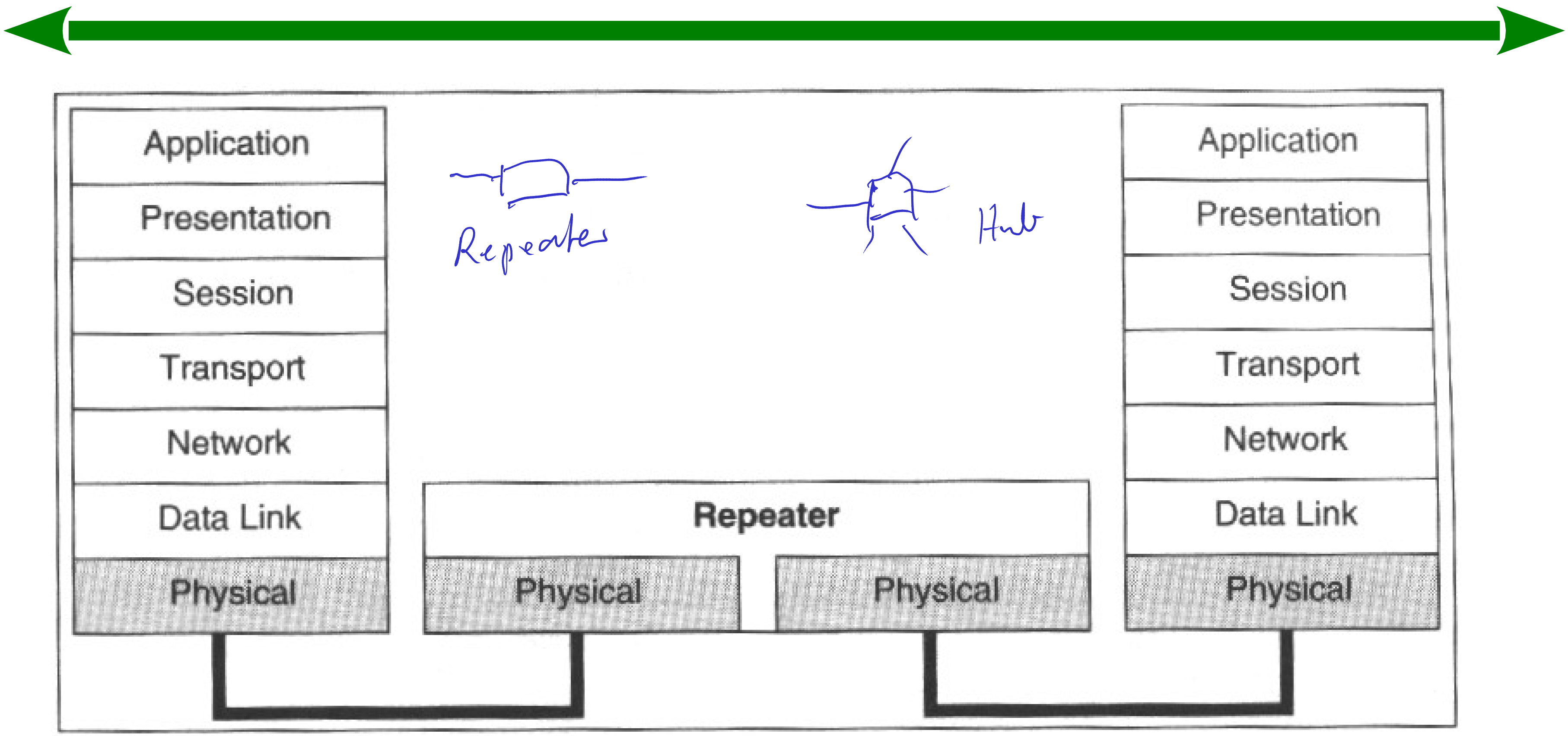
Command line: arp

ping ip/name sends test messages.

RARP
Reverse Address Resolution Protocol

- Same structure as ARP
- Nearly identical functionality
- But now the MAC address is available and the corresponding IP address unknown.

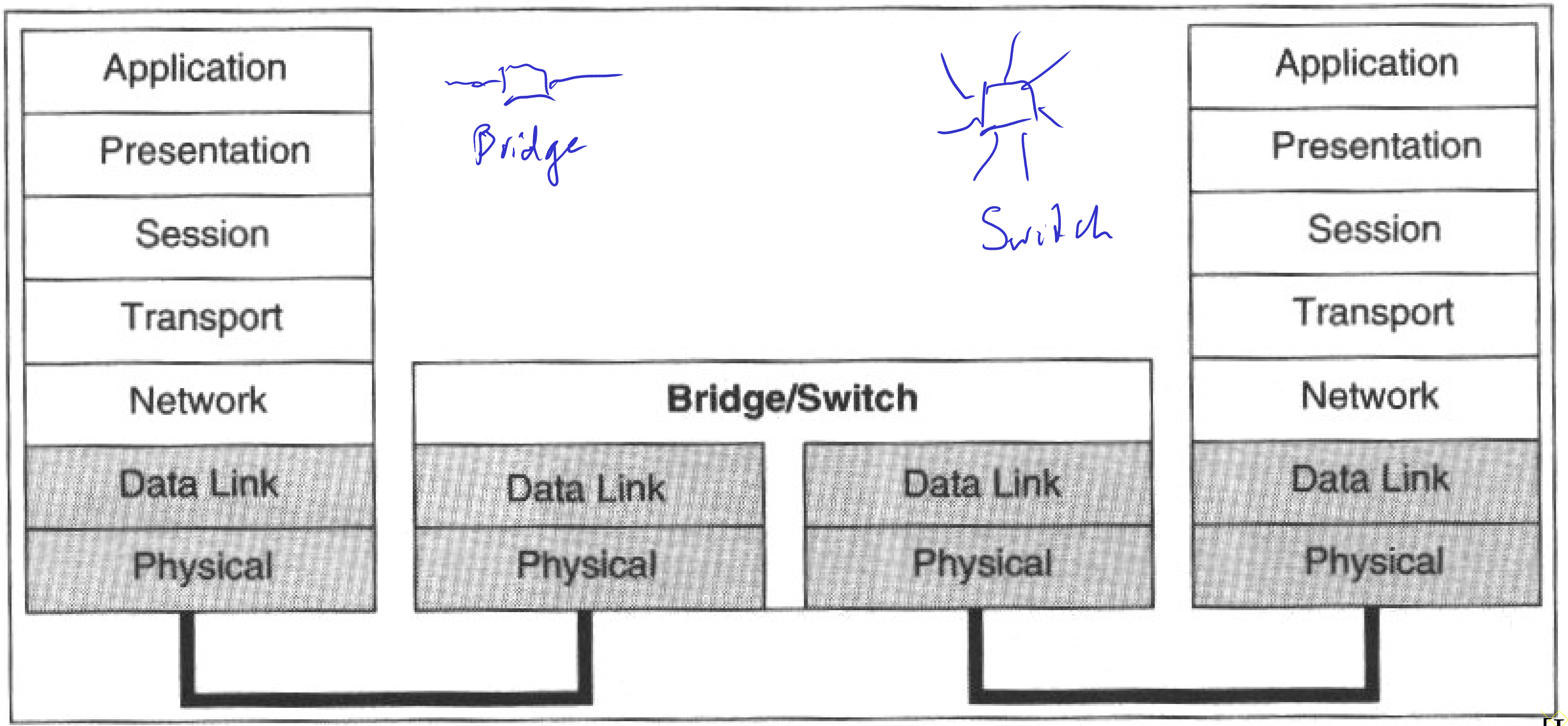
Repeater / Bridges / Router / Gateways



[Lien00]

Enhance the signal quality.
Errors in the messages are not detected and transmitted.

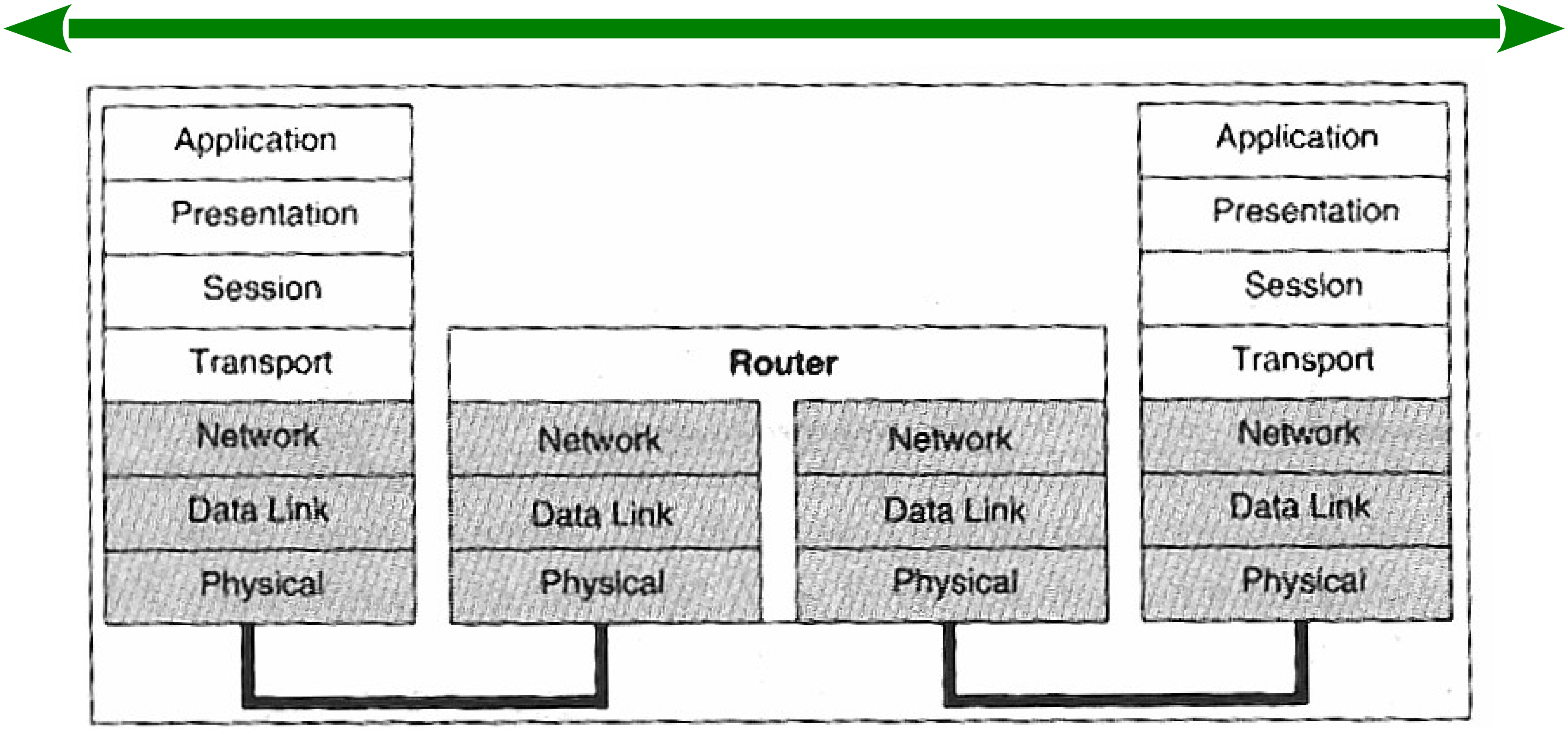
Repeater / **Bridges** / Router / Gateways



[Lien00]

Links two segments of a net. Error handling and load balancing possible.
Collisions are limited to one segment.
Usually transmits several higher protocols.

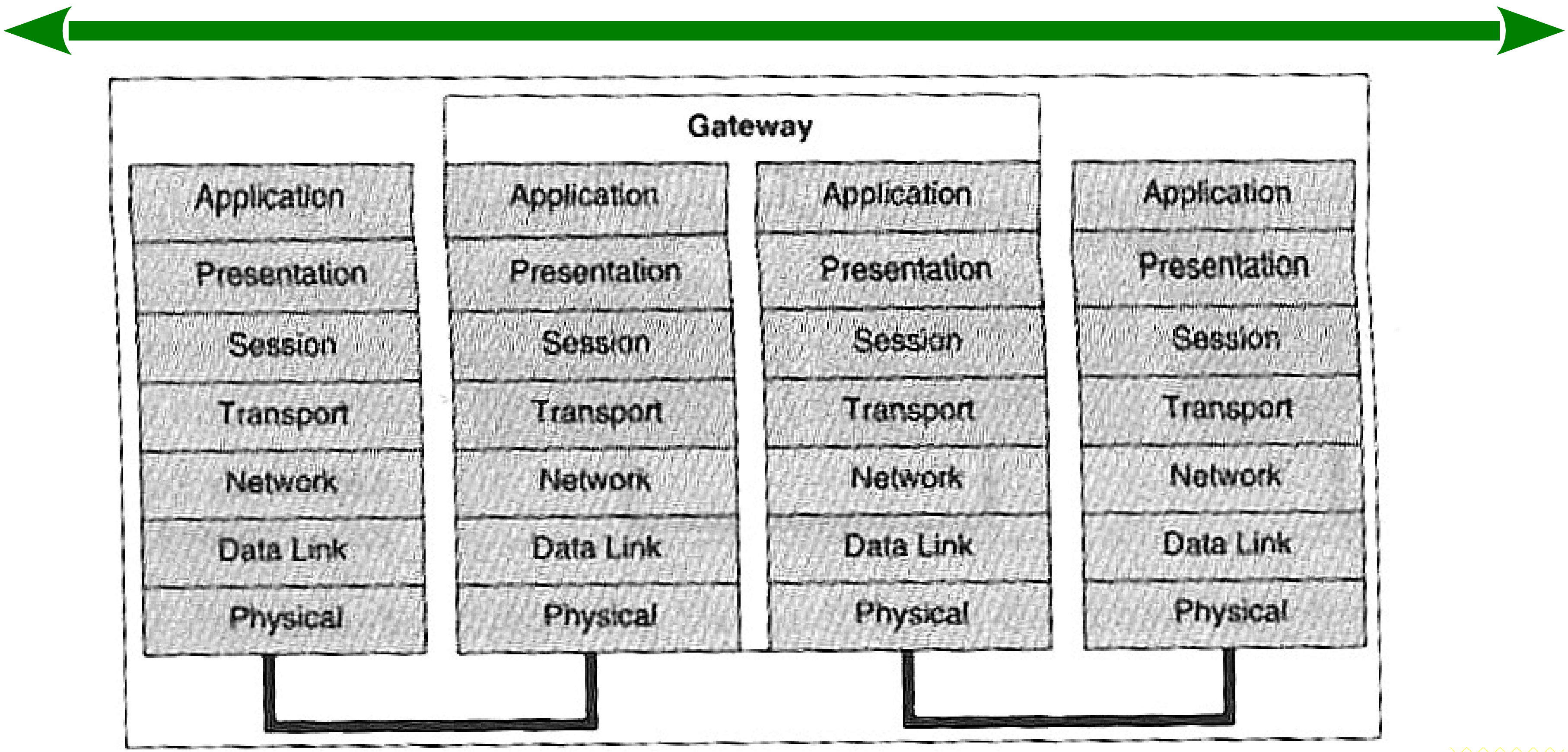
Repeater / Bridges / **Router** / Gateways



[Lien00]

Connects two logically separated nets.
Usually restricted to a defined protocol.

Repeater / Bridges / Router / **Gateways**



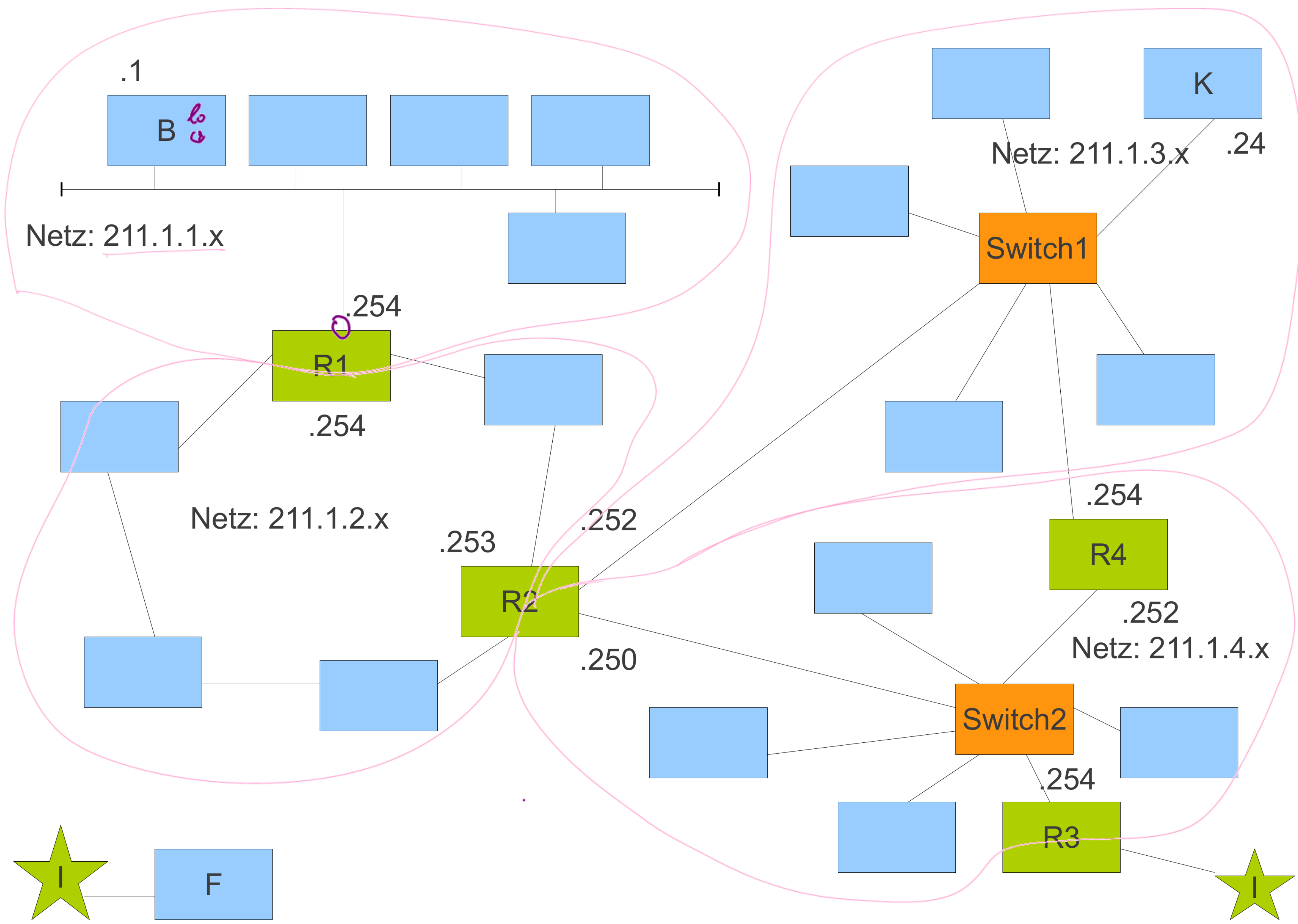
[Lien00]

Two different nets are connected that have different protocols used. The translation has to be done over lots of layers.

Routing

- Routing table:
- Routers communicate continuously about the best and cheapest route using routing protocols.
- Positive information is spread quickly, negative information spreads slowly. Disadvantage: Lot of time consumed in case of router damage
- Routing protocols:
  - RIP Routing Information Protocol (old version, no costs, only HOPs)
  - OSPF Open Shortest Path First (newer, with weights)
  - HELLO, IGRP (Cisco), IS-IS, BGP

Ziel-Netz	Route über
10.136.0.0	lokal
10.137.0.0	10.136.10.1
10.138.0.0	10.136.20.1
0.0.0.0 (default)	10.136.1.1



.1
B
lo
Netz: 211.1.1.x
.254
R1
.254
K
.24
Netz: 211.1.3.x
Switch1
Netz: 211.1.2.x
.253
.252
R2
.250
.254
R4
.252
Netz: 211.1.4.x
Switch2
.254
R3
I
F
I

Routing Table for B

route print
netstat -r

Destination IP	Destination Subnetmask	Router	Interface
127.0.0.1	255.0.0.0	–	lo
211.1.1.0	255.255.255.0	–	B.1
(default) 0.0.0.0	0.0.0.0	211.1.1.254 (R1.1)	B.1

Here (!) we name interfaces like that: name.netwno
with netwno 3. byte of IP

~~Routing Table of R1.~~

Repetition: Subnet mask

it masks out the network part of the IP

ip & sm → network address.

Are ip1 and ip2 in the same network?

if (ip1 & sm == ip2 & sm) .....

✓

Routing Table of R1:

Dest-IP	subnetmask	router	interface
127.0.0.0	255.0.0.0	–	lo
211.1.1.0	255.255.255.0	–	R1-1
211.1.2.0	255.255.255.0	–	R1-2
0.0.0.0	0.0.0.0	211.1.2.253 (R2)	R1.2

Routing Table of R2:

127.0.0.0	255.0.0.0	–	lo
211.1.2.0	255.255.255.0	–	R2.2
211.1.3.0	255.255.255.0	–	R2.3
211.1.4.0	255.255.255.0	–	R2.4
0.0.0.0	0.0.0.0	211.1.4.254 (R3)	R2.4
211.1.1.0	255.255.255.0	211.1.2.254 (R1)	R2.2

Routing Table for R4

IP Destination	Subnetmask	Router	Interface
127.0.0.1	255.0.0.0	–	lo
211.1.3.0	255.255.255.0	–	R4.3
211.1.4.0	255.255.255.0	–	R4.4
0.0.0.0	0.0.0.0	211.1.4.254 (R3)	R4.4
211.1.2.0	255.255.255.0	211.1.3.252 (R2)	R4.3
211.1.1.0	255.255.255.0	211.1.4.250 (R2)	R4.4

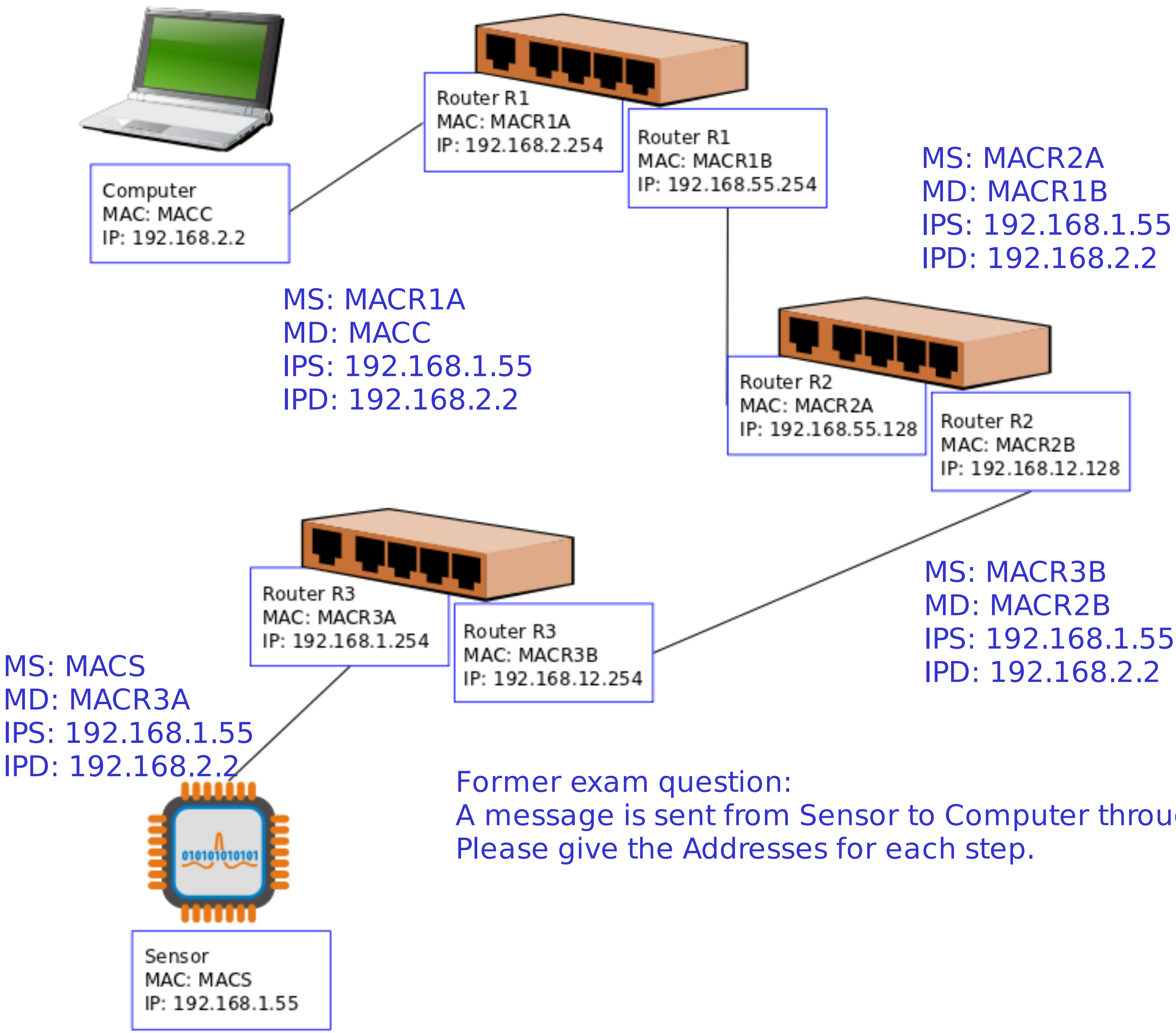
Now we send a message from B to K.

Q: What are the addresses for each step?

("Invitation to dinner")

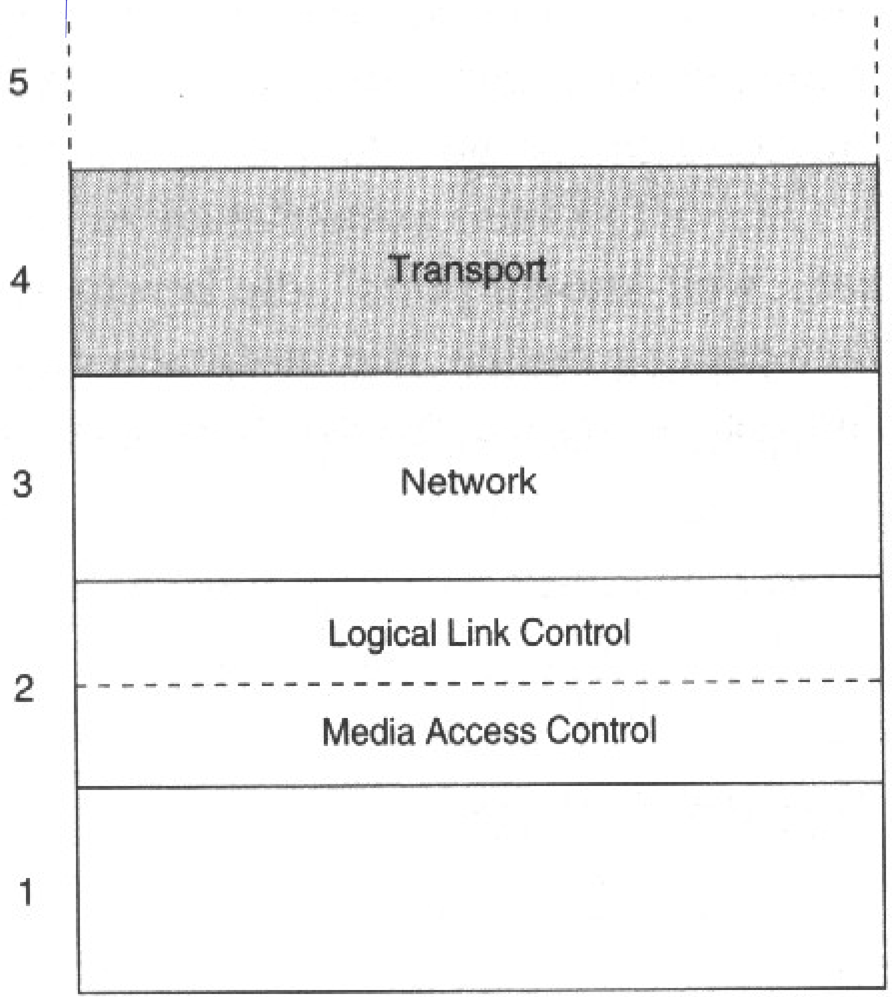
Please take care that K needs to reply!

	B → R1	R1 → R2	R2 → K
MAC Source	MAC(B)	MAC(R1.2)	MAC(R2.3)
MAC Dest.	MAC(R1.1)	MAC(R2.2)	MAC(K)
IP Source	211.1.1.1	211.1.1.1	211.1.1.1
IP Dest.	~~211.1.1.254~~ 211.1.3.24	211.1.3.24	211.1.3.24



Former exam question:
A message is sent from Sensor to Computer through all the routers.
Please give the Addresses for each step.

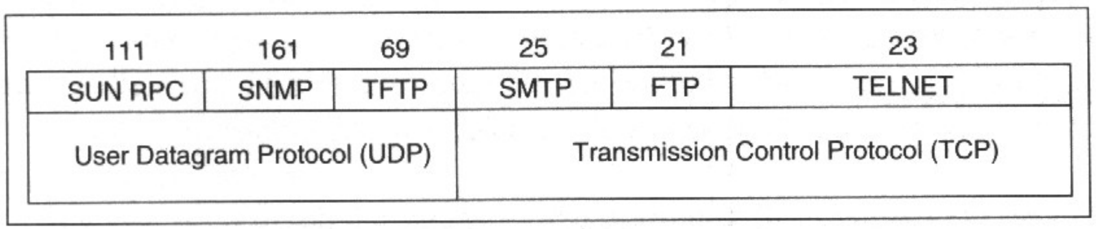
Layer 4: Transport

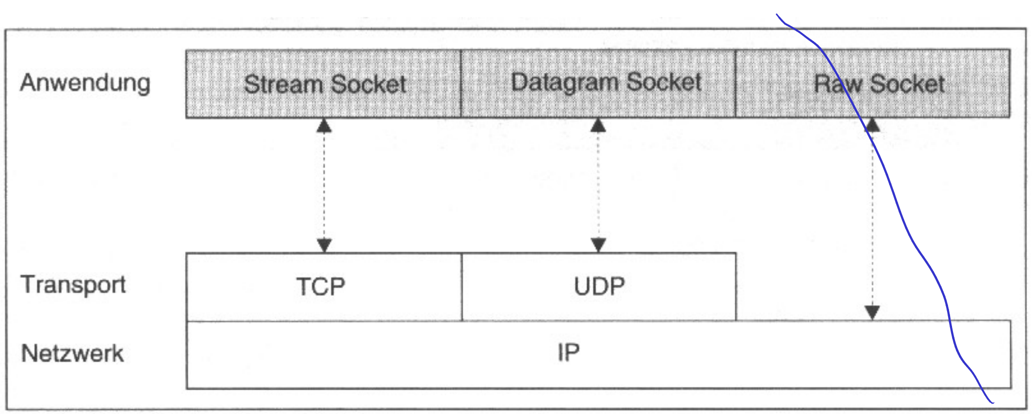


[Lien00]

/etc/services

\windows\system32\drivers\etc\services



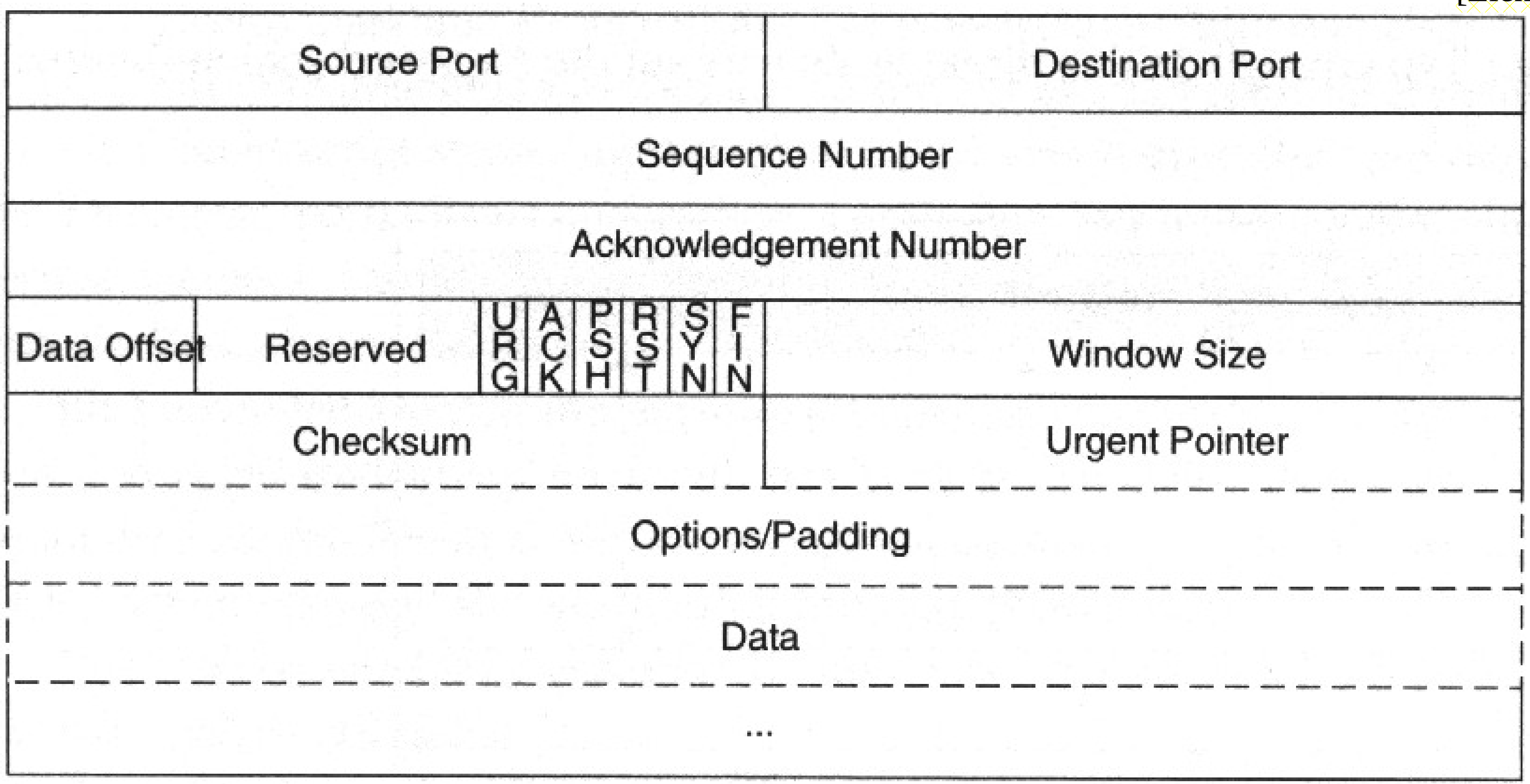


TCP
Transport Control Protocol

- Responsible for
  - Transmission of data streams
  - Virtual full duplex transmission
  - Control of the data flow
  - Error detection
  - Priority handling
  - Establish, close and maintain connections
  - Buffering
- The data stream is divided into several segments (according to the MSS: Maximum Segment Size)
  - The MSS is chosen during session setup.
  - Segments get numbers (in 32bit-units)

TCP-Datagramm

[Lien00]

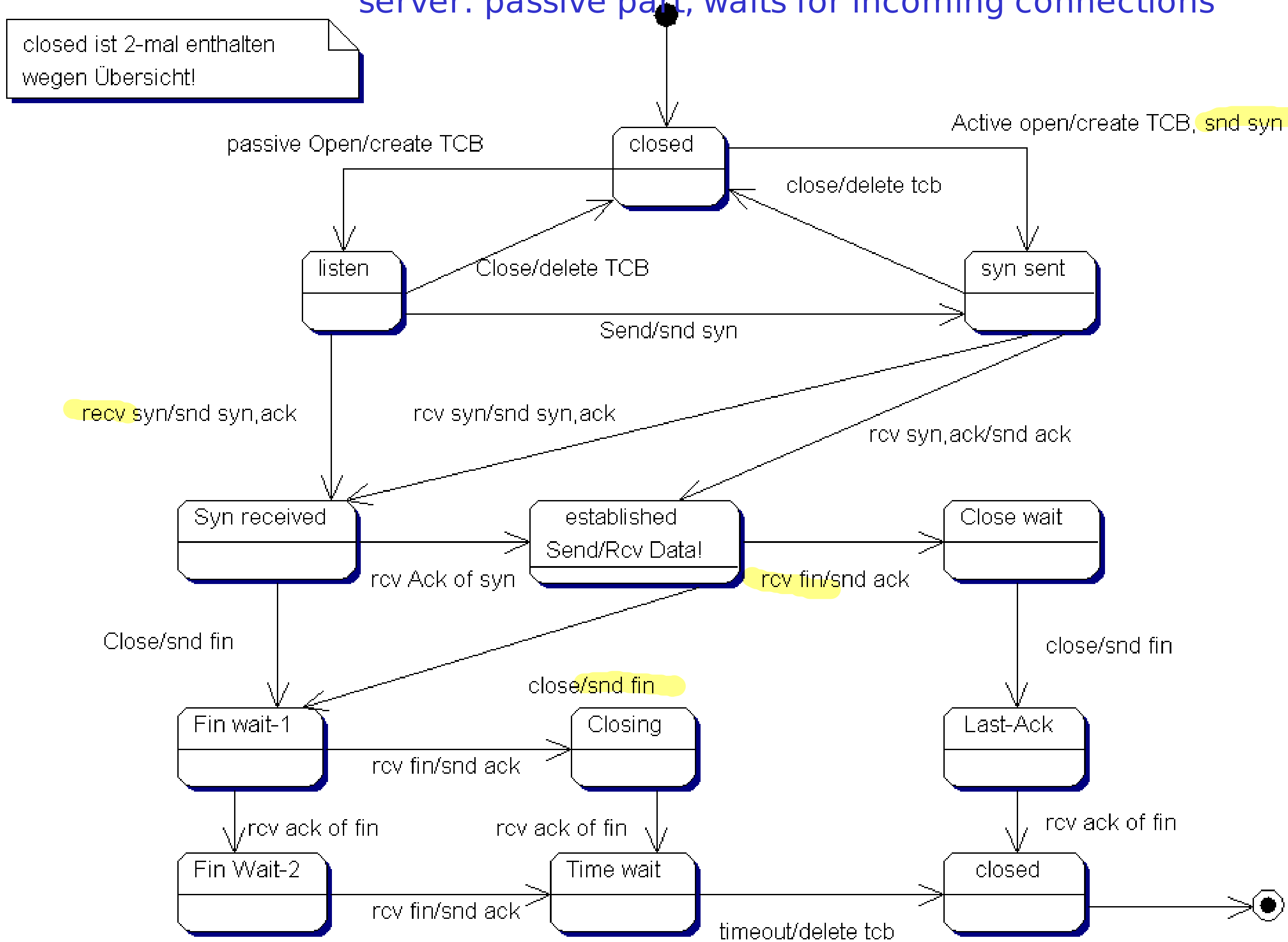


Statechart of a Session

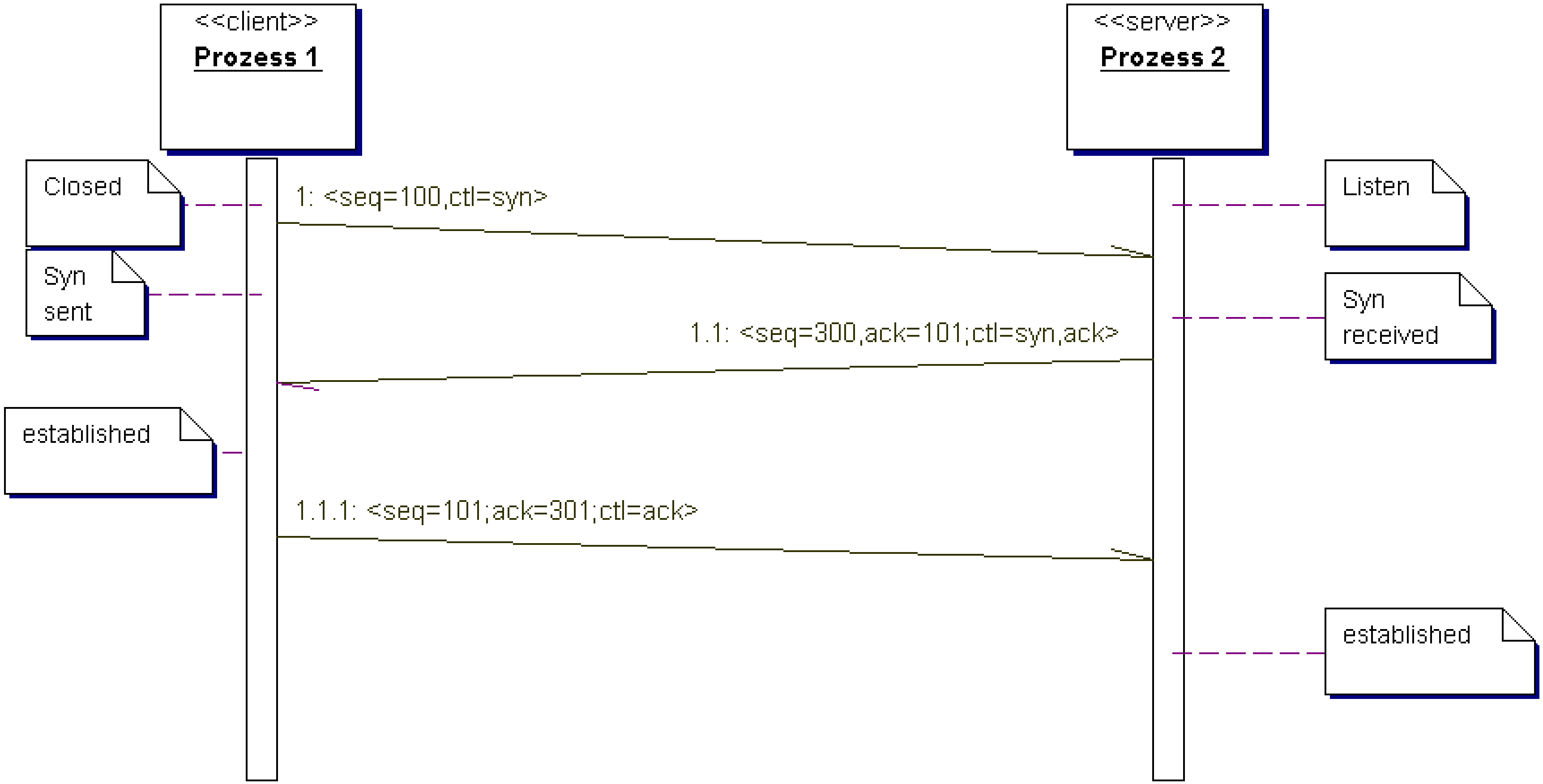
"client-server" as long as the session is established. Afterwards there is no difference anymore.

client: active part, initiates the session

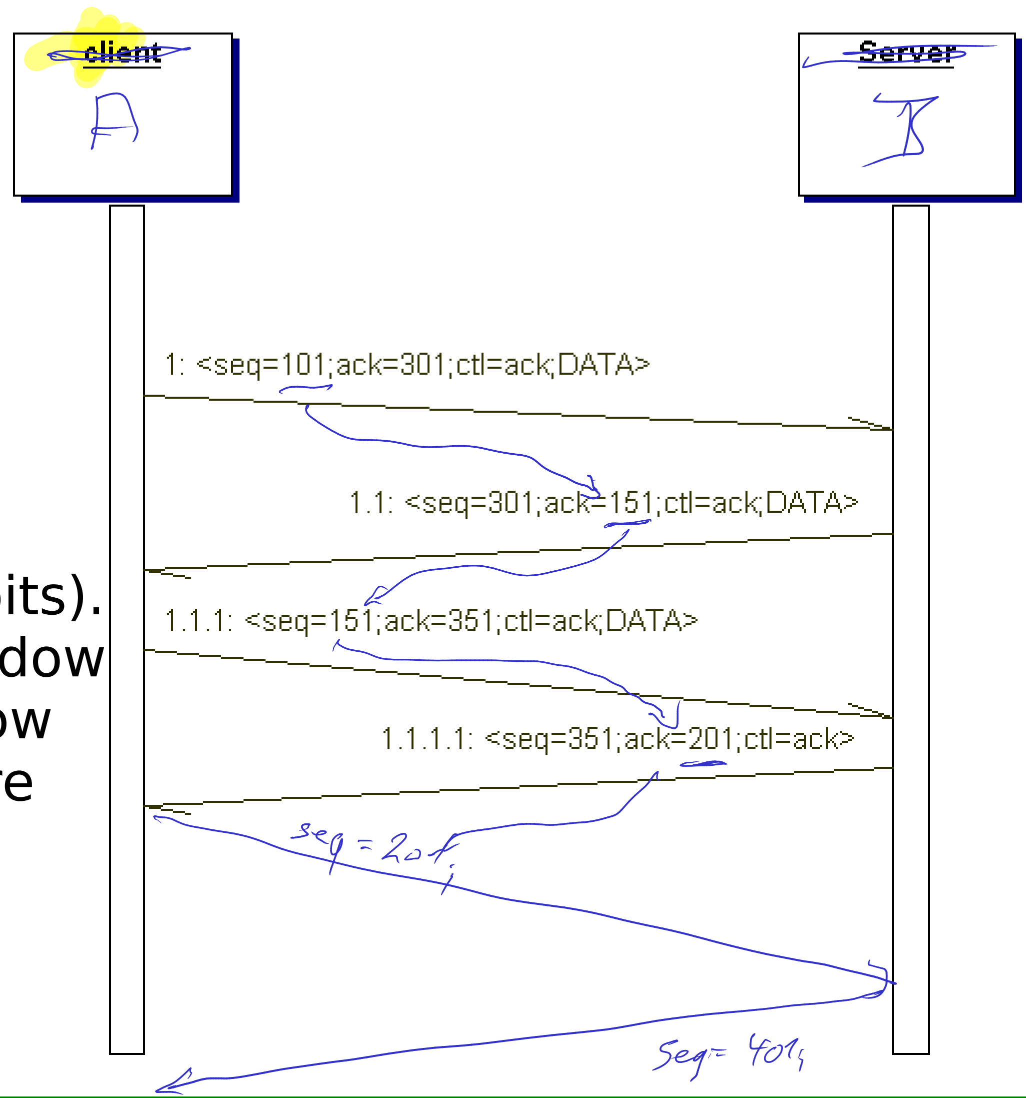
server: passive part, waits for incoming connections



Three way handshake



TCP Transfer

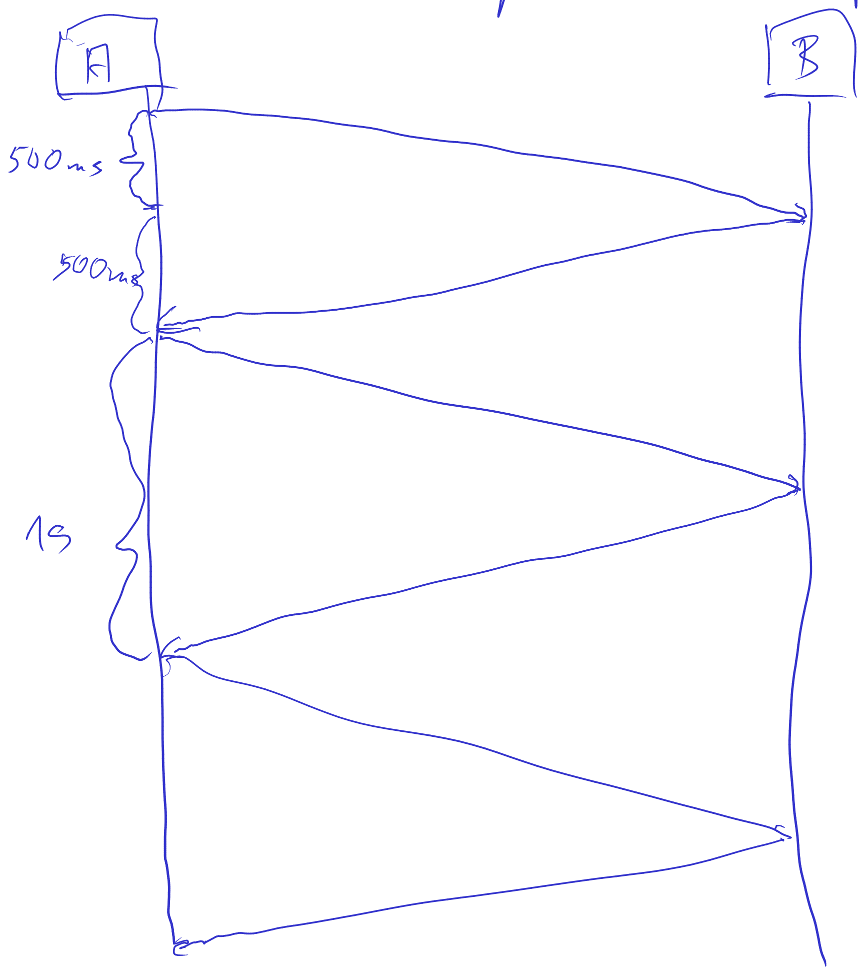


ack-no is the next sequence no that we want to receive.

So we acknowledge ack - 1

Sequence number counts
4 bytes units (32 bits).
The maximum window size determines how many segments are sent before an acknowledgement is needed.

We want to transfer a 4,5GB file



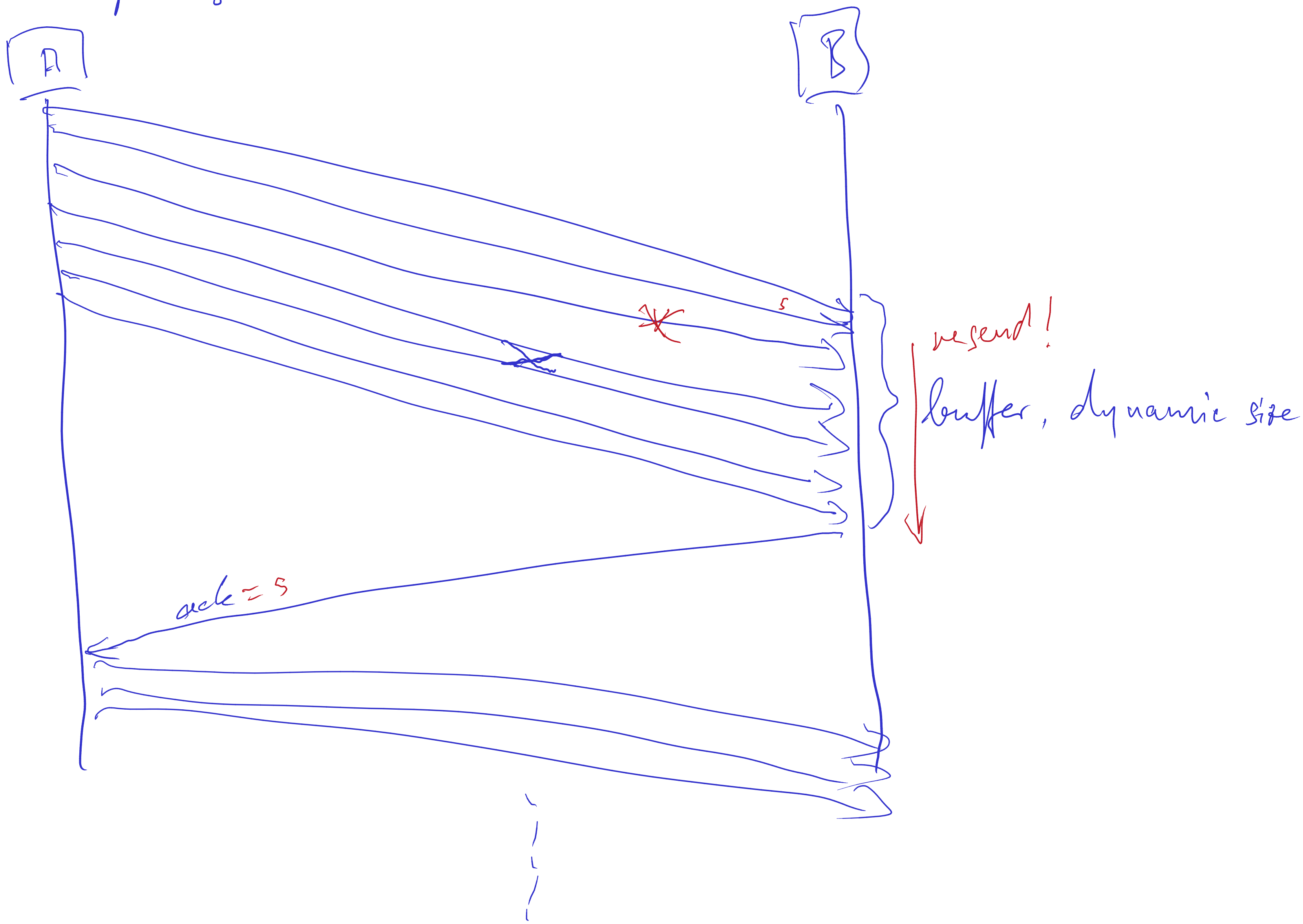
How long will it take to download the file if we wait for acknowledgment after each packet?

packet size 1.5kB (ethernet)

1,5 kB/s

for full file: 3 000 000 s

With buffering:



UDP
User Datagram Protocol

- Sessionless, not reliable
- Same properties as IP
- But ports: Several processes possible
- UDP defines an API to use IP
- UDP services are e.g.: TFTP, DNS, SNMP, RPC

UDP: Datagramm

[Lien00]

Source Port	Destination Port
Length	Checksum
Data	
...	

UDP-Datagramm-Format

Protocols of Layer 7

- SMTP (TCP, port 25)
- POP3 (TCP, port 110)
- Telnet (TCP, port 23)
- ftp (TCP, port 21)
-

SMTP
Simple Mail Transfer Protocol

RECV FROM post.strato.de <<<<	220 post.webmailer.de ESMTP Sendmail 8.9.3/8.8.7mail 8.9.3/ 8.8.7mail 8.9.3/8.8.70 (MET DST)..
SEND TO post.strato.de >>>>	HELO Leize866.Leize.de
RECV FROM post.strato.de <<<<	250 post.webmailer.de Hello dialinpool.tiscali.de [62.246.9.40](may be forged), pleased to meet you..
SEND TO post.strato.de >>>>	MAIL FROM: Leize@Leize.de
RECV FROM post.strato.de <<<<	250 Leize@Leize.de... Sender ok..
SEND TO post.strato.de >>>>	RCPT TO: Leize@Leize.de
RECV FROM post.strato.de <<<<	250 Leize@Leize.de... Recipient ok..
SEND TO post.strato.de >>>>	DATA
RECV FROM post.strato.de <<<<	354 Enter mail, end with "." on a line by itself..
SEND TO post.strato.de >>>>	Date: Sat, 27 Jun 2002 20:02:05 CEST
SEND TO post.strato.de >>>>	To: Thorsten@Leize.de
SEND TO post.strato.de >>>>	Subject: Testmail mit telnet!
SEND TO post.strato.de >>>>	
SEND TO post.strato.de >>>>	Hier beginnt der Mailtext. Zwischen Header und tesxt muss ein
SEND TO post.strato.de >>>>	Leerzeichen stehen.
SEND TO post.strato.de >>>>	Gruss und vile =viel Spass beim ausprobieren.
SEND TO post.strato.de >>>>	.
RECV FROM post.strato.de <<<<	250 UAA07771 Message accepted for delivery..
SEND TO post.strato.de >>>>	QUIT
RECV FROM post.strato.de <<<<	221 post.webmailer.de closing connection..

HTTP
Hypertext Transfer Protocol

- HTTP-Request:
  - Method SP Request-URL SP HTTP-Version CR+LF
  - Header SP Value CR+LF (n*)
  - CR+LF
  - Data
  - CR+LF
- HTTP-Response
  - HTTP-Version SP Status-code SP Reason-Phrase CR+LF
  - Header SP Value CR+LF (n*)
  - CR+LF
  - Data
  - CR+LF

You may use wireshark to see the messages on the network
Please install wireshark.

telnet google.de 80
GET / HTTP/1.0

empty line

References

- [Furr98] Frank J. Furrer: "Ethernet-TCP/IP für die Industrieautomation", Hüthig, 1998
- [DesEle] Zeitschrift Design & Elektronik, Extraheft: µC-Webserver
- [Internet] Tons of information.
- [Lien00] Gerhard Lienemann: "TCP/IP-Grundlagen", Heise, 2000
- [Hein01] Mathias Hein, Michael Reisner: "TCP/IP Ge-packt", mitp, 2001
- [Lip98] Klaus Lipinski: "Lexikon TCP/IP Internetworking", itp, 1998