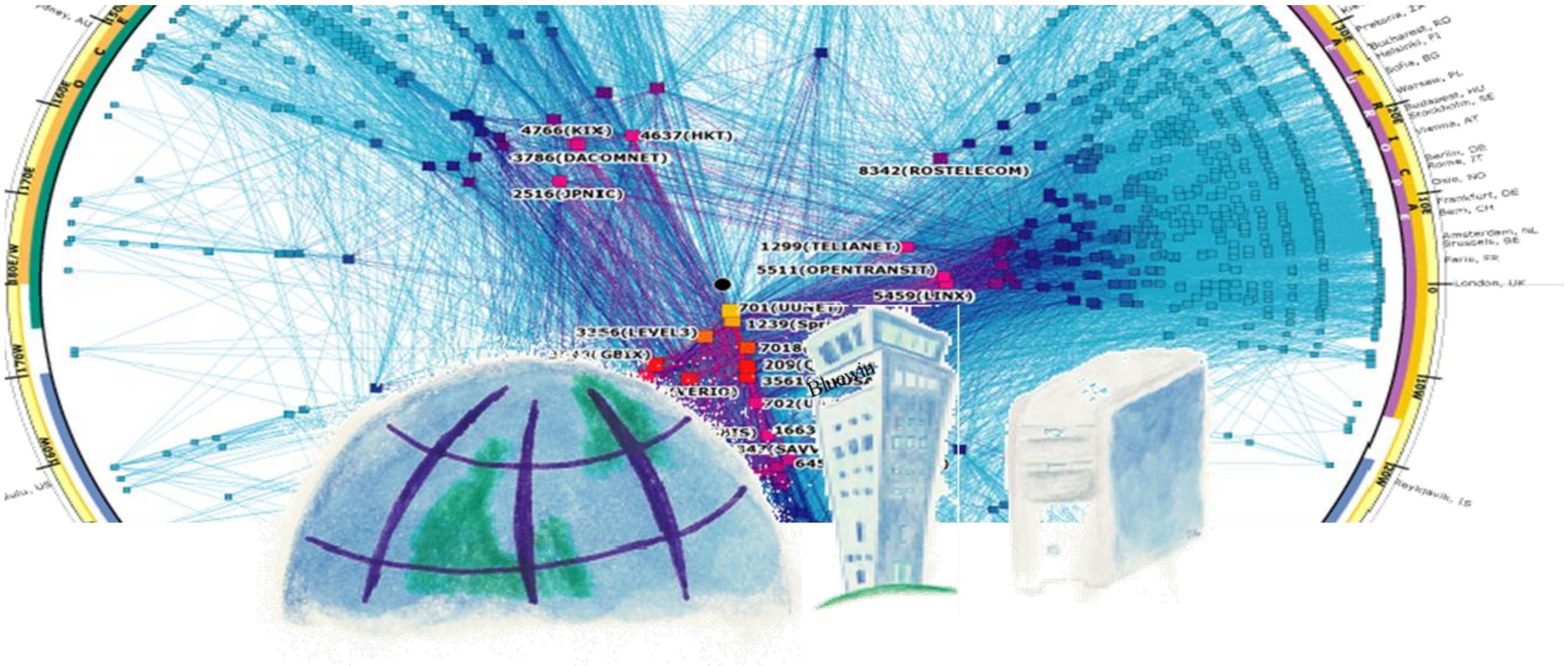


The TCP/IP Architecture

Jean-Yves Le Boudec

2020

EPFL

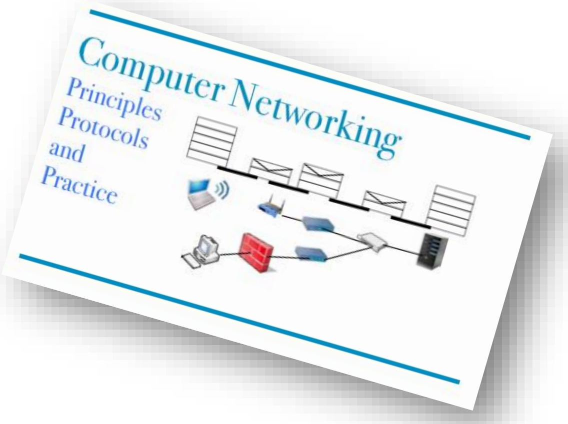
Objective

Understand Layered Model of Communication Systems

Know what MAC, IP addresses and DNS names are

Textbook

Chapter 2: Introduction of edition 1



TCP/IP is a layered architecture

Why ?

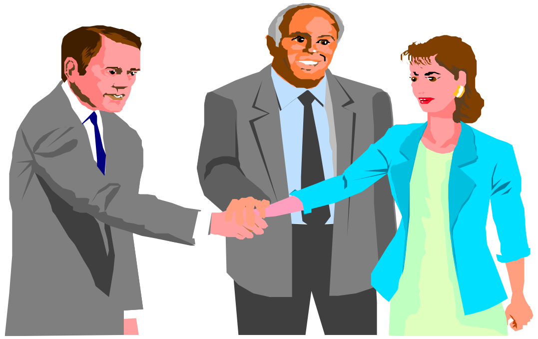
Divide and conquer – make things manageable

What is it ?

Communication

Application

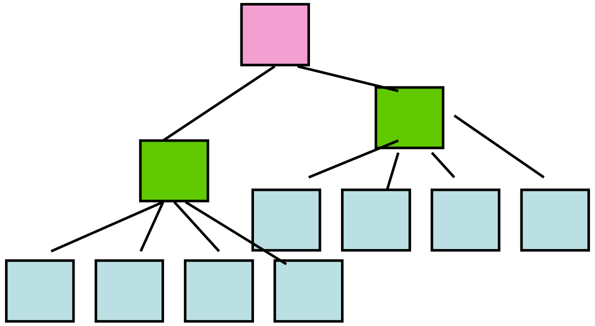
Transport



Interconnection

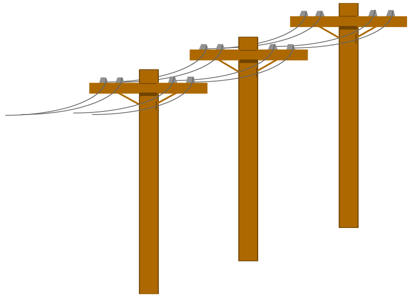
Network

MAC



Distance

Physical

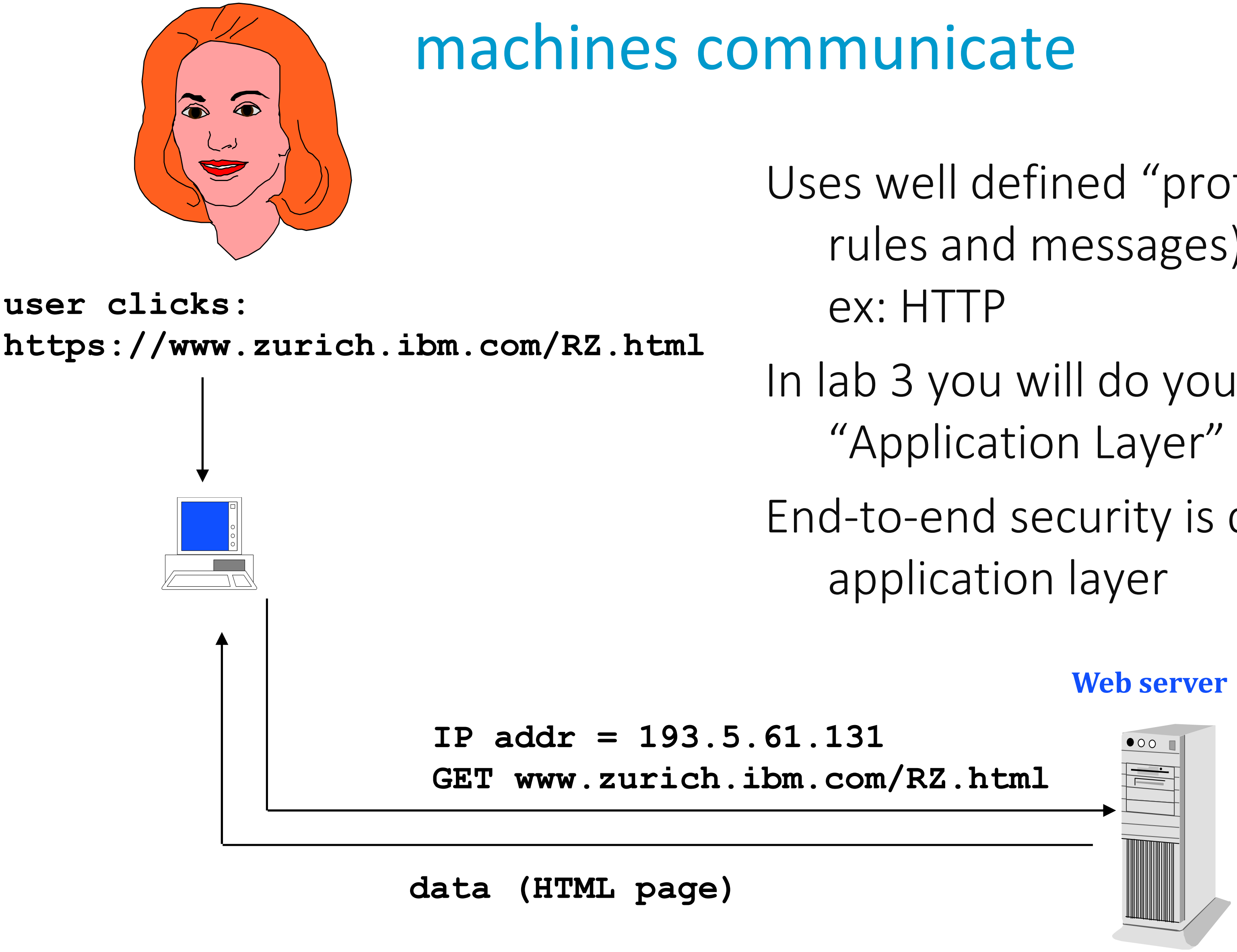


Application Layer helps people and machines communicate

Uses well defined “protocols” (set of rules and messages)
ex: HTTP

In lab 3 you will do your own “Application Layer”

End-to-end security is done in the application layer



user clicks:
https://www.zurich.ibm.com/RZ.html

Web server

IP addr = 193.5.61.131
GET www.zurich.ibm.com/RZ.html

data (HTML page)

Transport Layer helps Application layer

Transport Layer provides programming interface to application layer

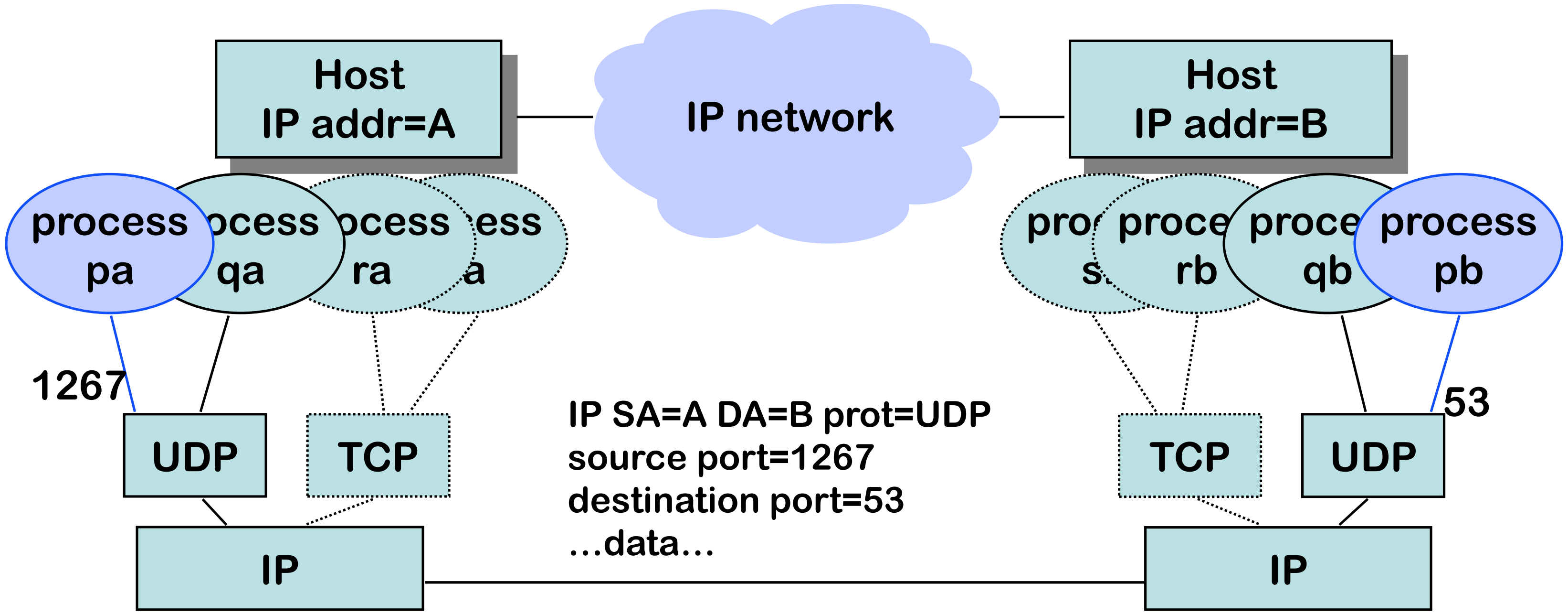
Exists in two versions

- UDP (User Datagram Protocol)
  Unit of information is a message
  Unreliable (message may be lost) -- No sequence guarantee
- TCP (Transmission Control Protocol)
  Reliable: if some data is lost somewhere, TCP retransmits it
  Stream service: the data is delivered at destination in the order it was sent by source (sequence guarantee)
  Unit of information is a byte; grouping of data into blocks may be different at destination than at source
- We will also study QUIC, which is a sort of “super-TCP” and is over UDP

Transport Layer Uses Port Numbers

Port numbers allow to differentiate source / destination processes on one machine

Source and destination port numbers are carried in UDP/TCP header

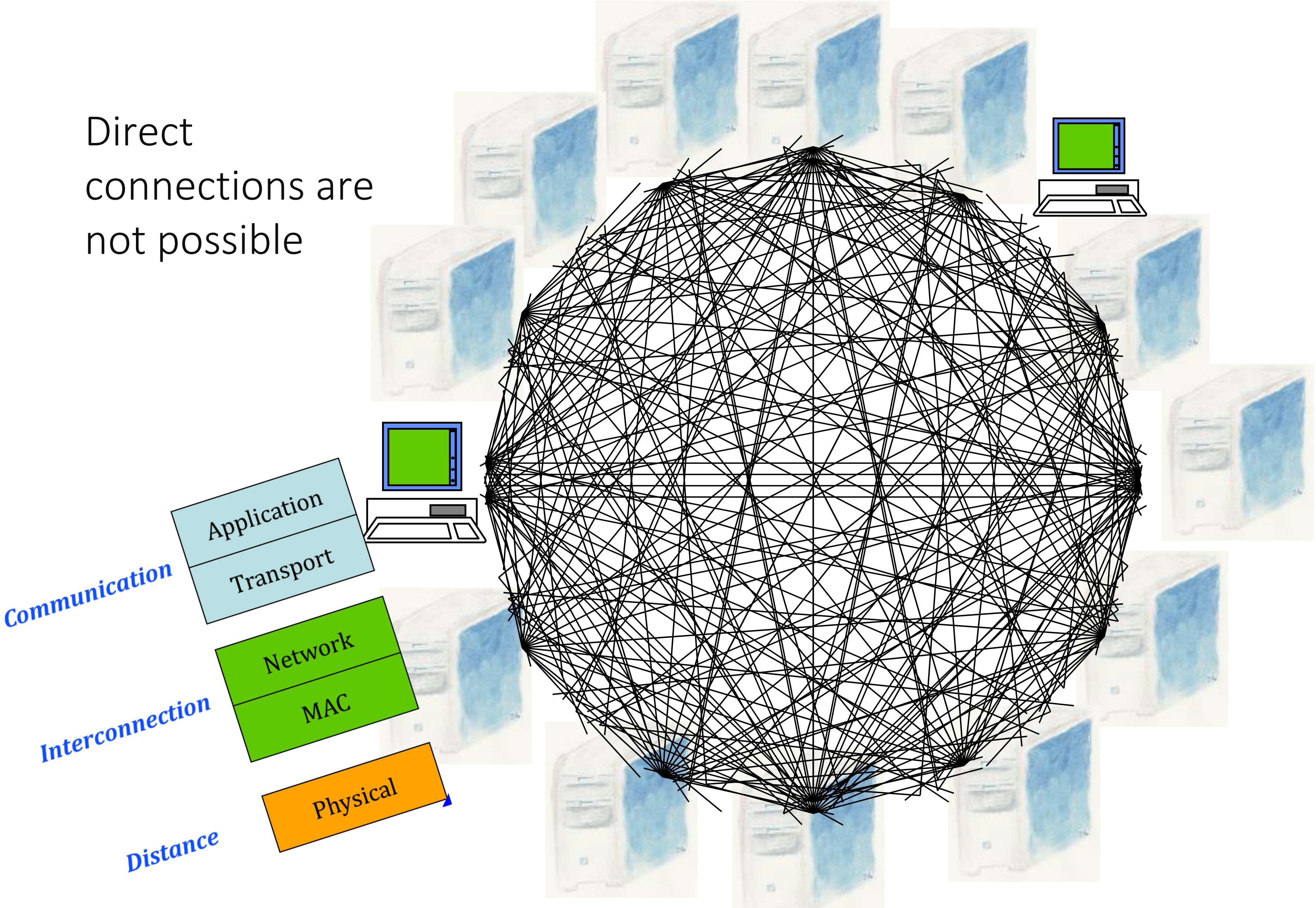


IP SA=A DA=B prot=UDP
source port=1267
destination port=53
…data…

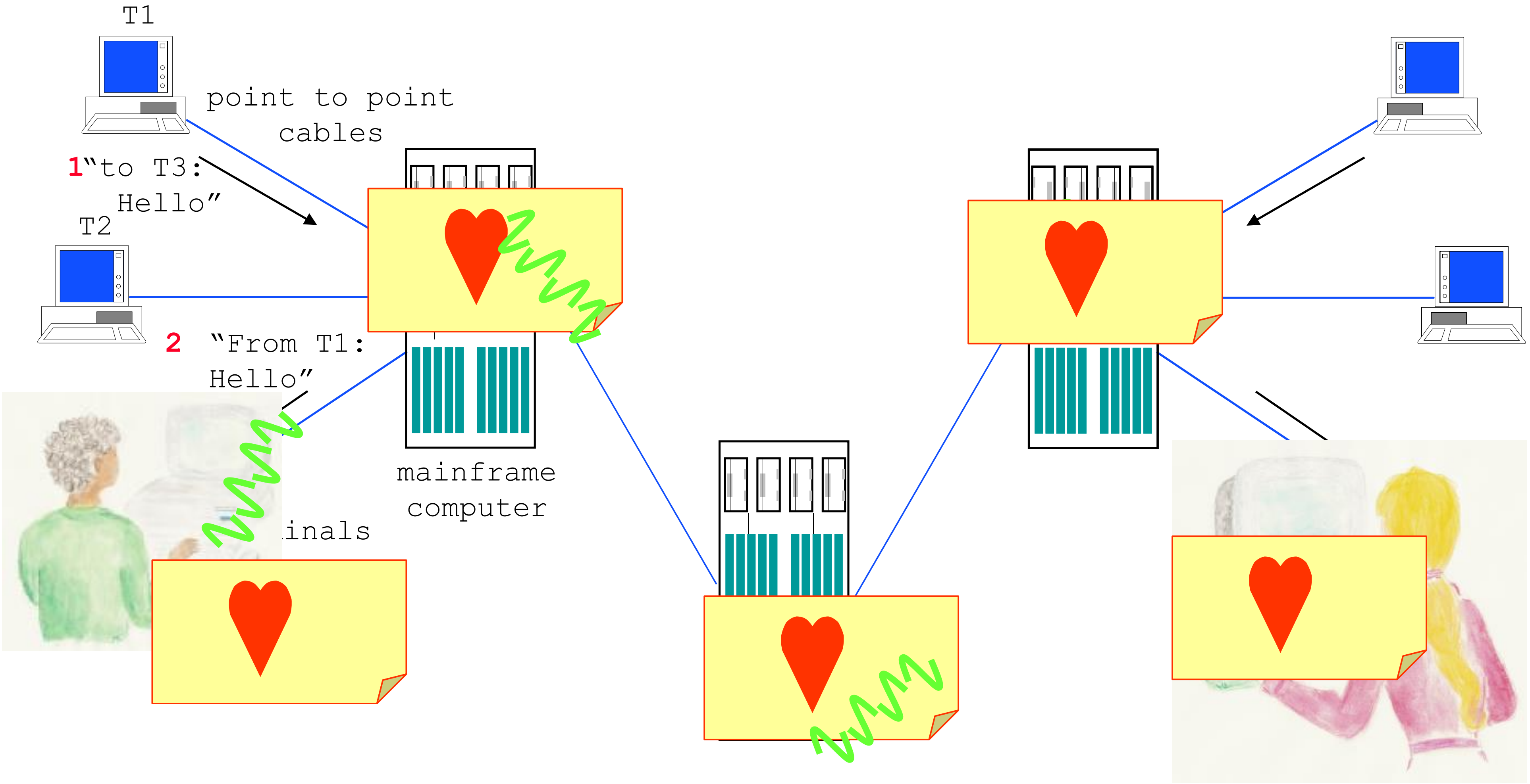
E.g.: Process pa on machine A sends a request to machine B's "DNS Server" process pb.

Network Layer provides full connectivity

Direct connections are not possible



The Very First Computer Networks (Bitnet, SNA) used Store and Forward



T1

point to point
cables

1"to T3:
Hello"

T2

2 "From T1:
Hello"

mainframe
computer

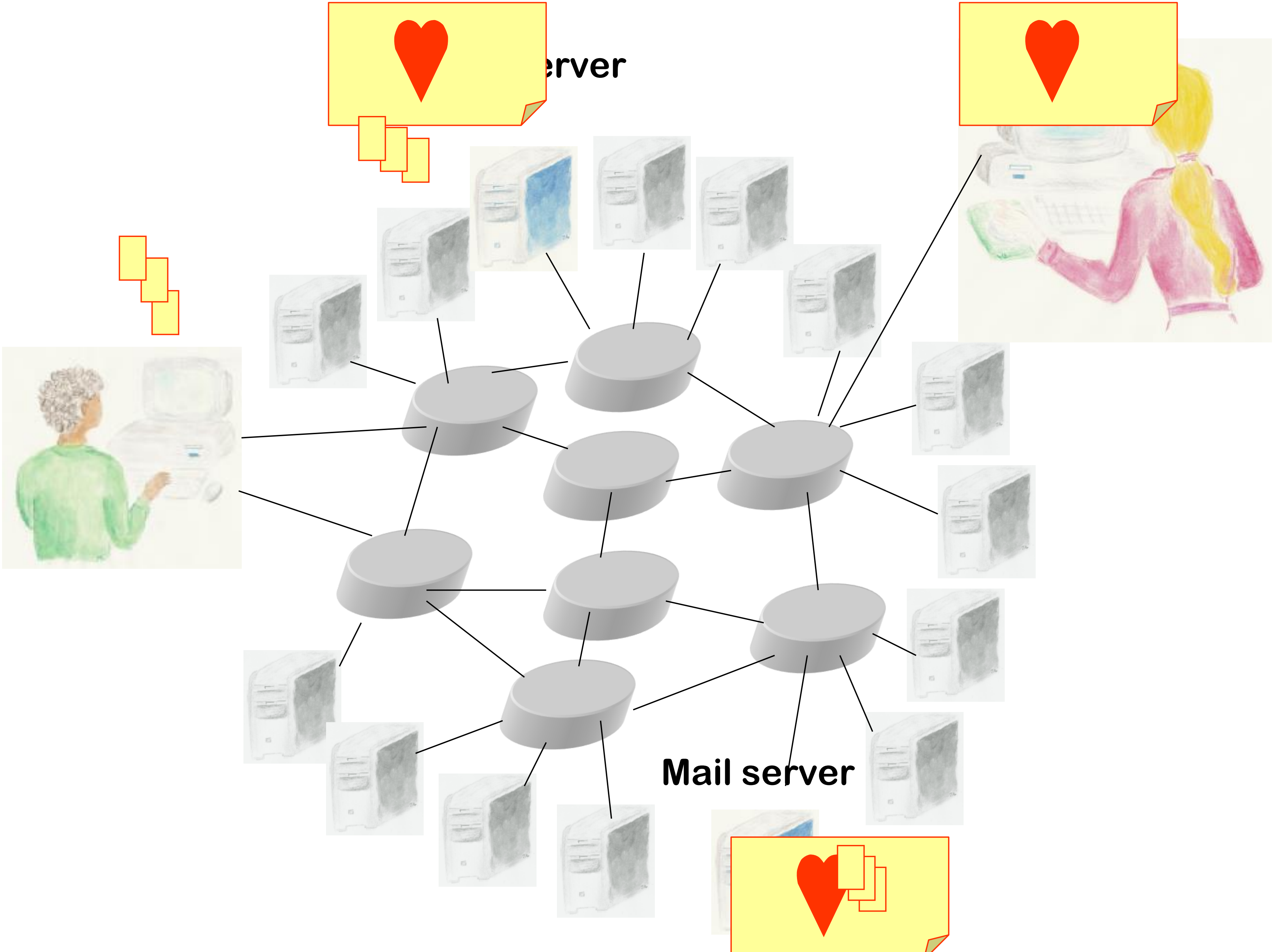
inals

The Internet Uses Packet Switching

Data is broken into chunks called IP packets of size ≤ 1500 bytes

One packet ≈ postcard, contains source and destination addresses

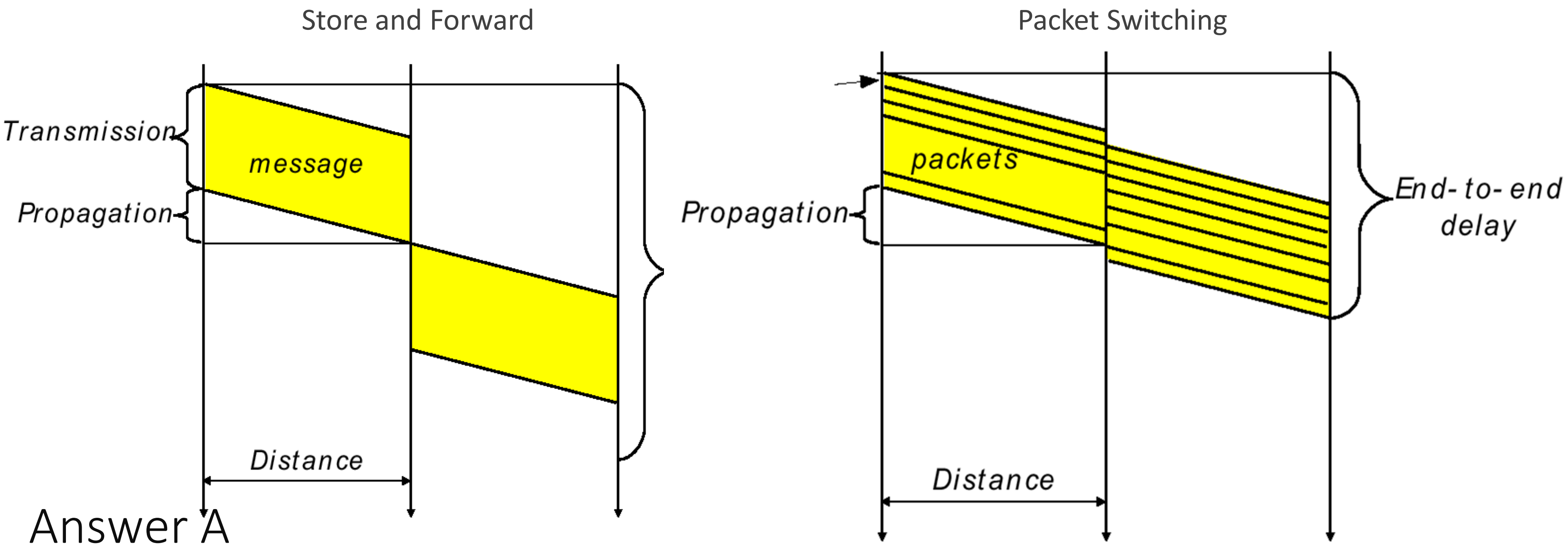
Louis Pouzin 1973, first datagram network, Cyclades, France
Vint Cerf and Bob Kahn, TCP/ IP,May 74


erver
Mail server

Why packet switching?

A. It reduces buffer required in routers
B. It reduces the bit error rates
C. It increases capacity
D. I don’t know

Solution



Answer A

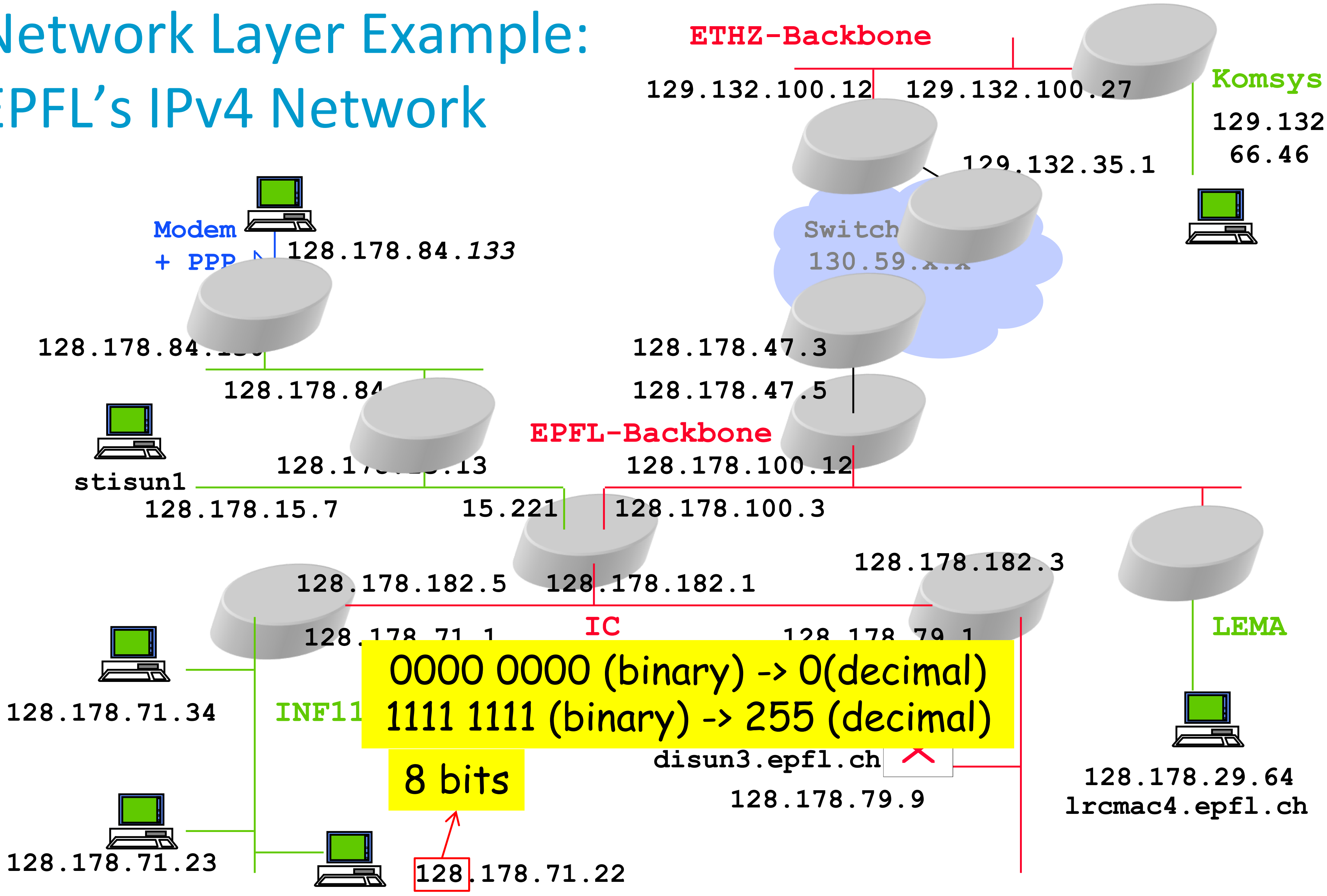
required buffer at intermediate systems is reduced (“cut-through”)

delay is also reduced.

Bit error rate remains the same but the probability that a packet is affected by a bit error is smaller than for an entire message (as a message is typically larger than a single packet).

Capacity is not increased (see “Transport”).

Network Layer Example: EPFL's IPv4 Network



0000 0000 (binary) -> 0(decimal)
1111 1111 (binary) -> 255 (decimal)

8 bits

There are two network layers: IPv4 and IPv6

The old numbering plan is IPv4 – 32 bits
uses dotted decimal notation – one number in {0, 1,…, 254, 255} = 8 bits
an EPFL address: 128.178.156.23
private addresses: 192.168.1.23, 172.16.3.4, 10.201.121.98.

The current numbering plan is IPv6 – 128 bits
uses hexadecimal notation – one hexadecimal digit in {0,1,…, e,f} = 4 bits
an EPFL public address: 2001:0620:0618:01a6:0a00:20ff:fe78:30f9
an EPFL private address: fd24:ec43:12ca:01a6:0a00:20ff:fe78:30f9

IPv4 and IPv6 network layers are distinct and incompatible
→ see later

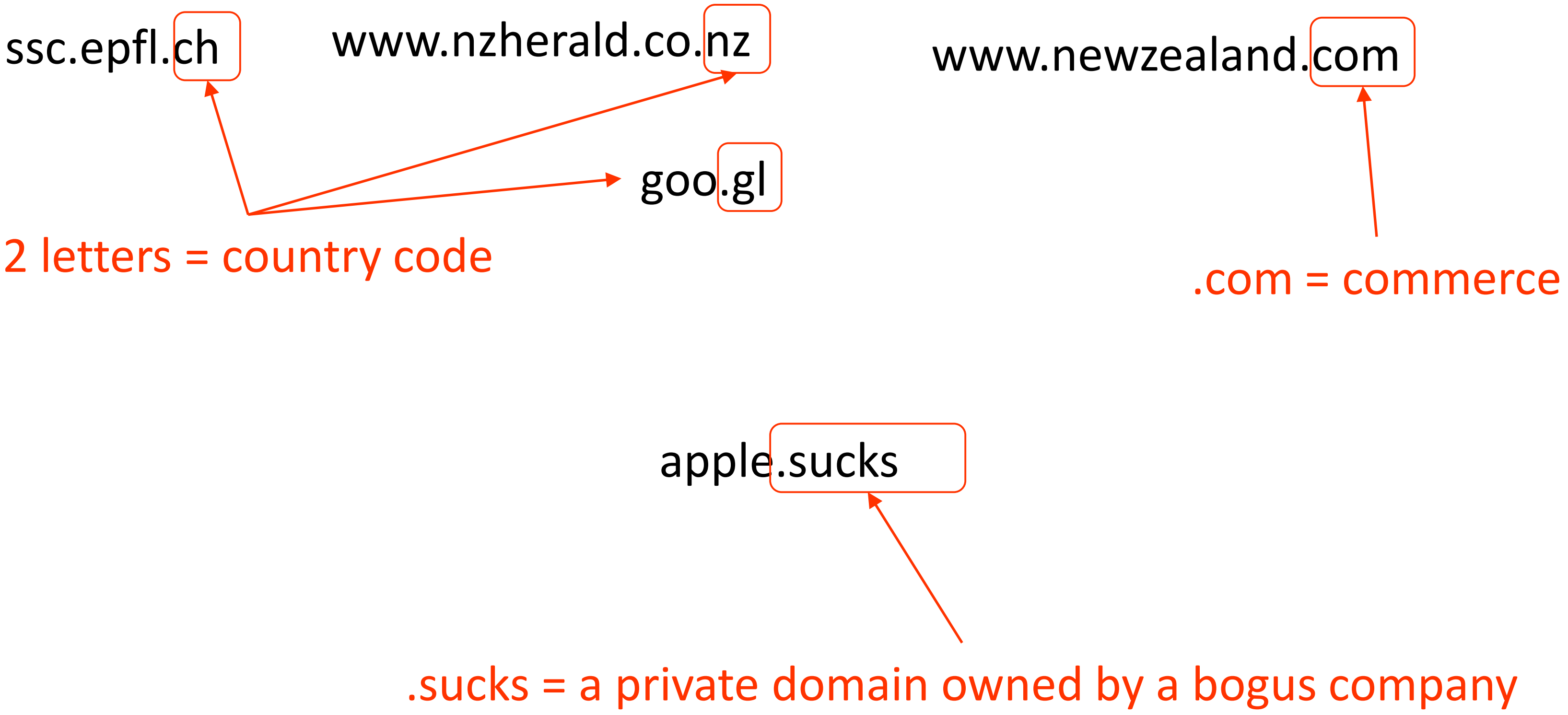
Adresses and Names

Names are human readable synonyms for IPv4 or IPv6 addresses

Examples:

ssc.epfl.ch

smtp.sunrise.ch



ssc.epfl.ch

www.nzherald.co.nz

www.newzealand.com

goo.gl

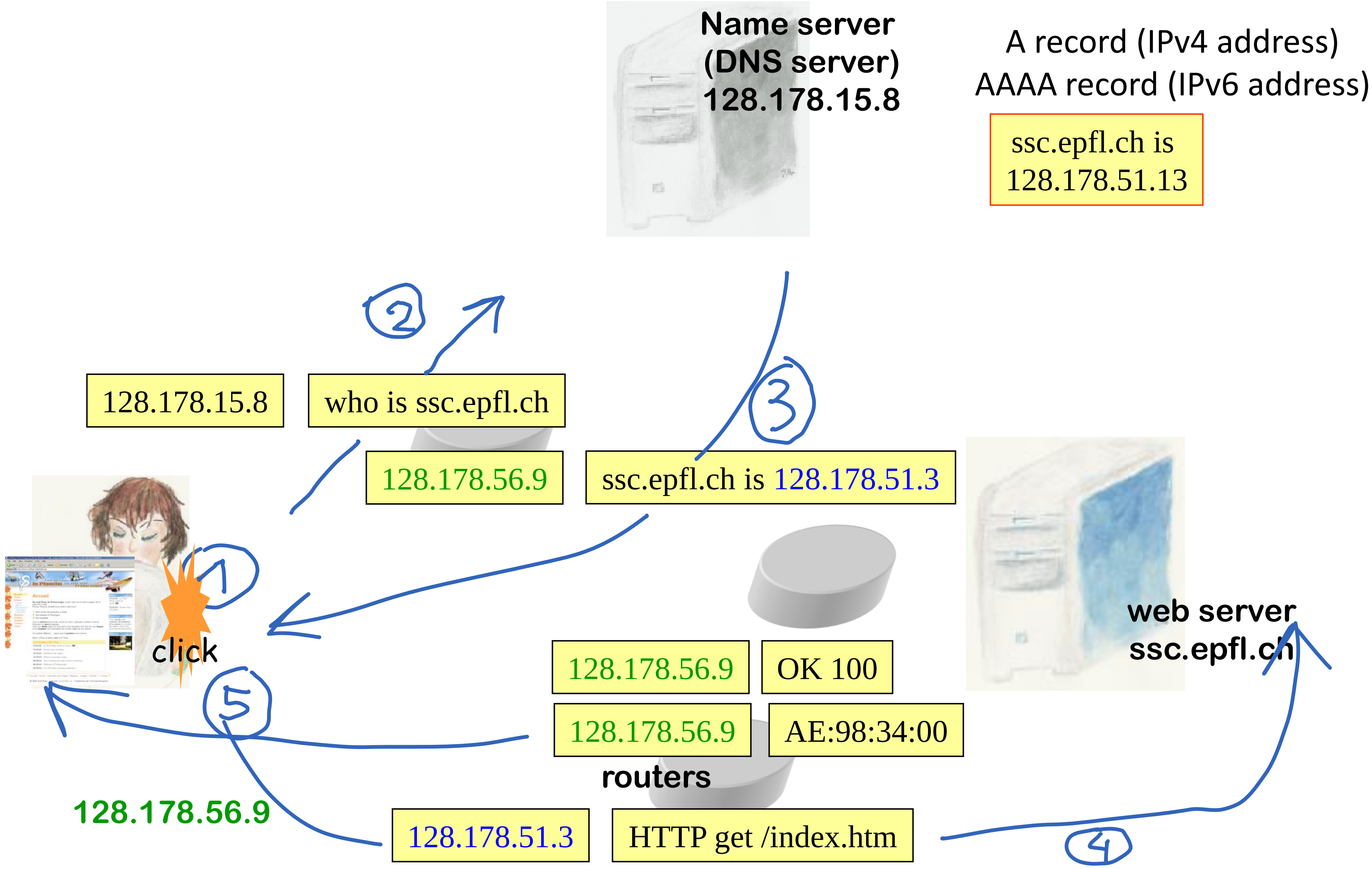
2 letters = country code

.com = commerce

apple.sucks

.sucks = a private domain owned by a bogus company

Names are mapped to addresses by DNS servers – not present in IP headers



Name server
(DNS server)
128.178.15.8

A record (IPv4 address)
AAAA record (IPv6 address)

ssc.epfl.ch is 128.178.51.13

128.178.15.8 | who is ssc.epfl.ch

128.178.56.9

ssc.epfl.ch is 128.178.51.3

click

web server
ssc.epfl.ch

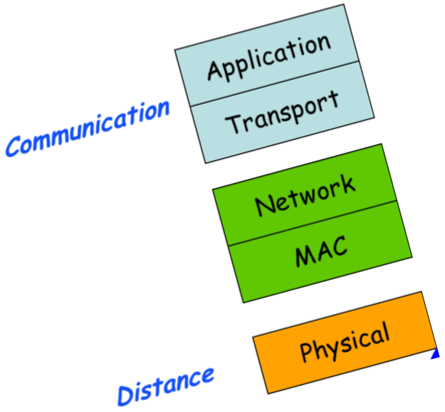
128.178.56.9 | OK 100

128.178.56.9 | AE:98:34:00

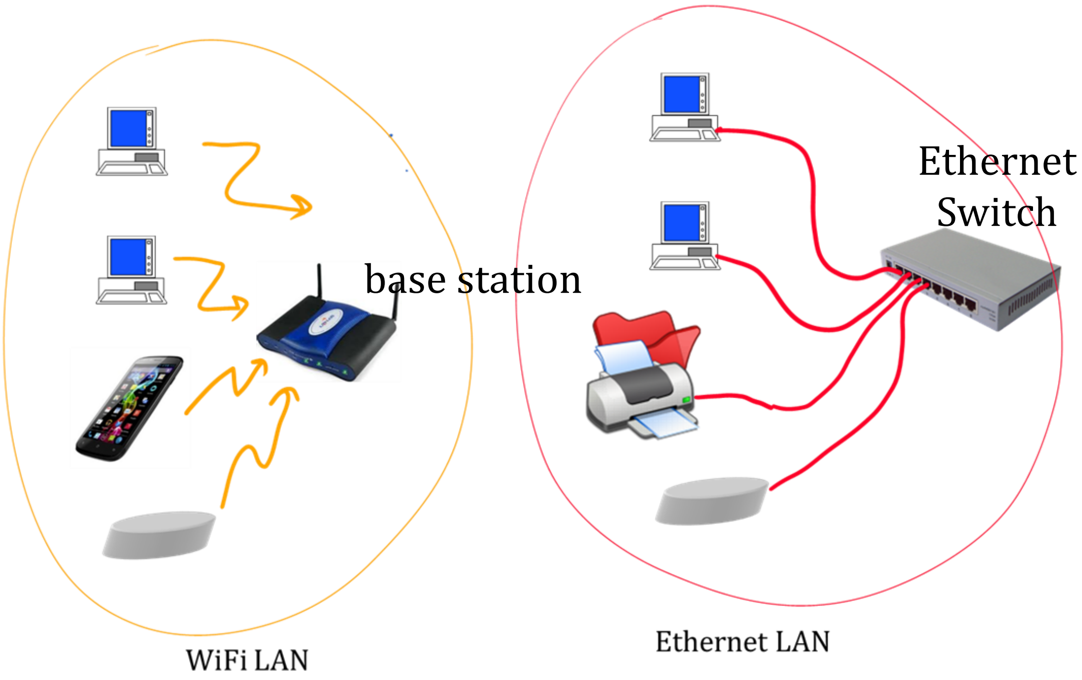
routers

128.178.56.9

128.178.51.3 | HTTP get /index.htm



Link Layer = MAC layer interconnects a small number of devices without any configuration



Using either Wireless or Cabled (Ethernet) or combination

Uses a method to avoid collisions (see later) + uses *MAC addresses*

MAC = Medium Access Control

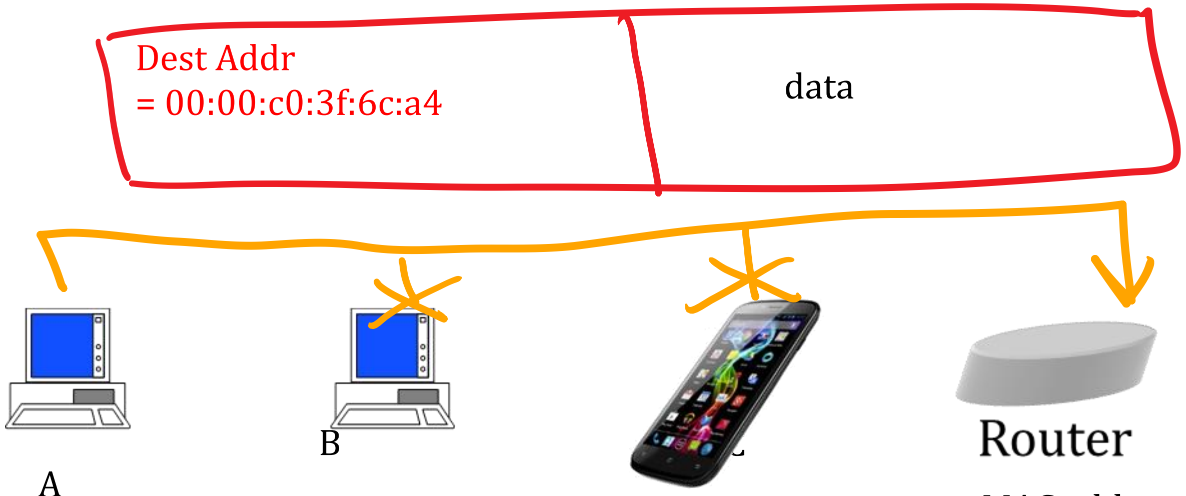
MAC Addresses are Hardware Addresses

MAC address: 48 bits = set by manufacturer, unique, in principle

sender puts destination MAC address in a frame

all stations within the local area read all frames; keep only if destination address matches (true for WiFi as well as Ethernet)

Destination MAC Address is sent in the clear, no encryption (but data can be encrypted)



Dest Addr
= 00:00:c0:3f:6c:a4

data

WiFi radio system
or
Ethernet cabling system

A
MAC address =
08:00:20:71:0d:d4

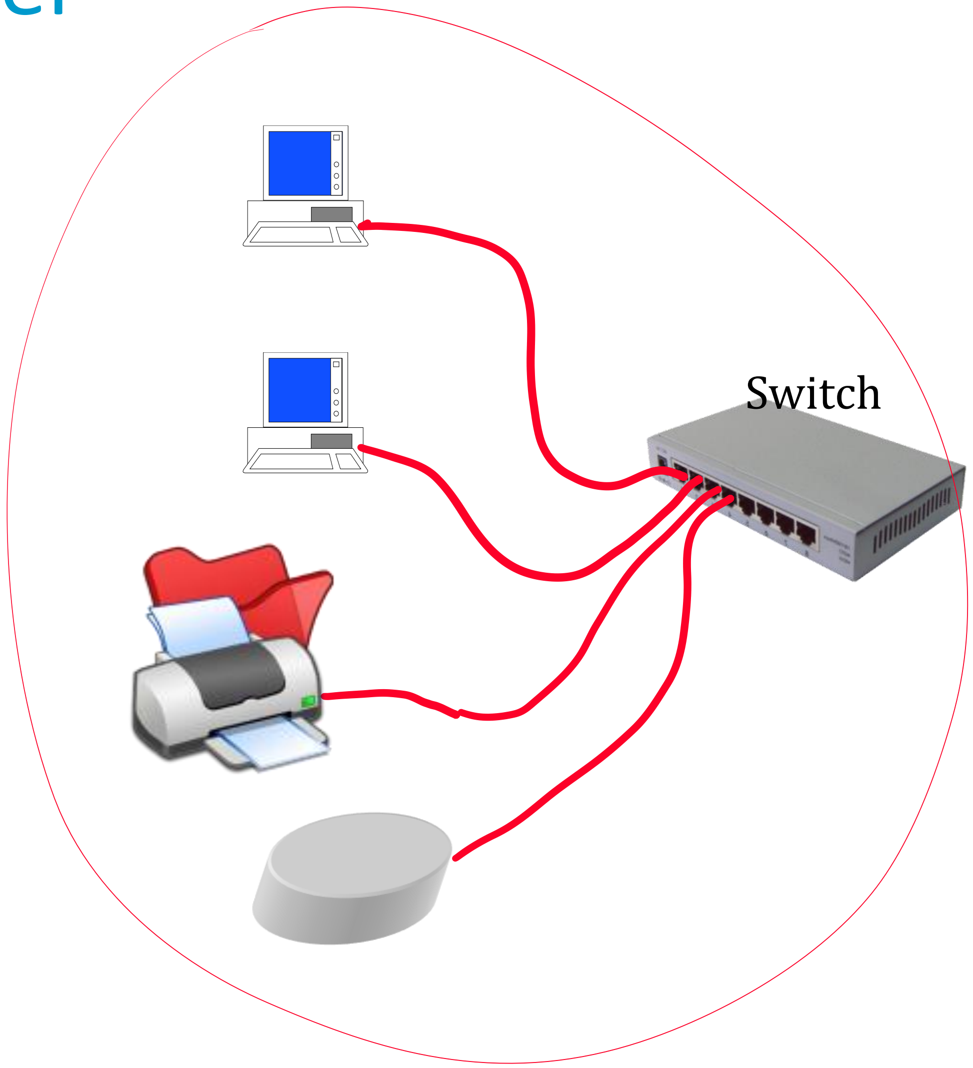
B

Router

MAC address =
00:00:c0:3f:6c:a4

Local Area Network = A set of devices that are connected at the MAC layer

WiFi LAN

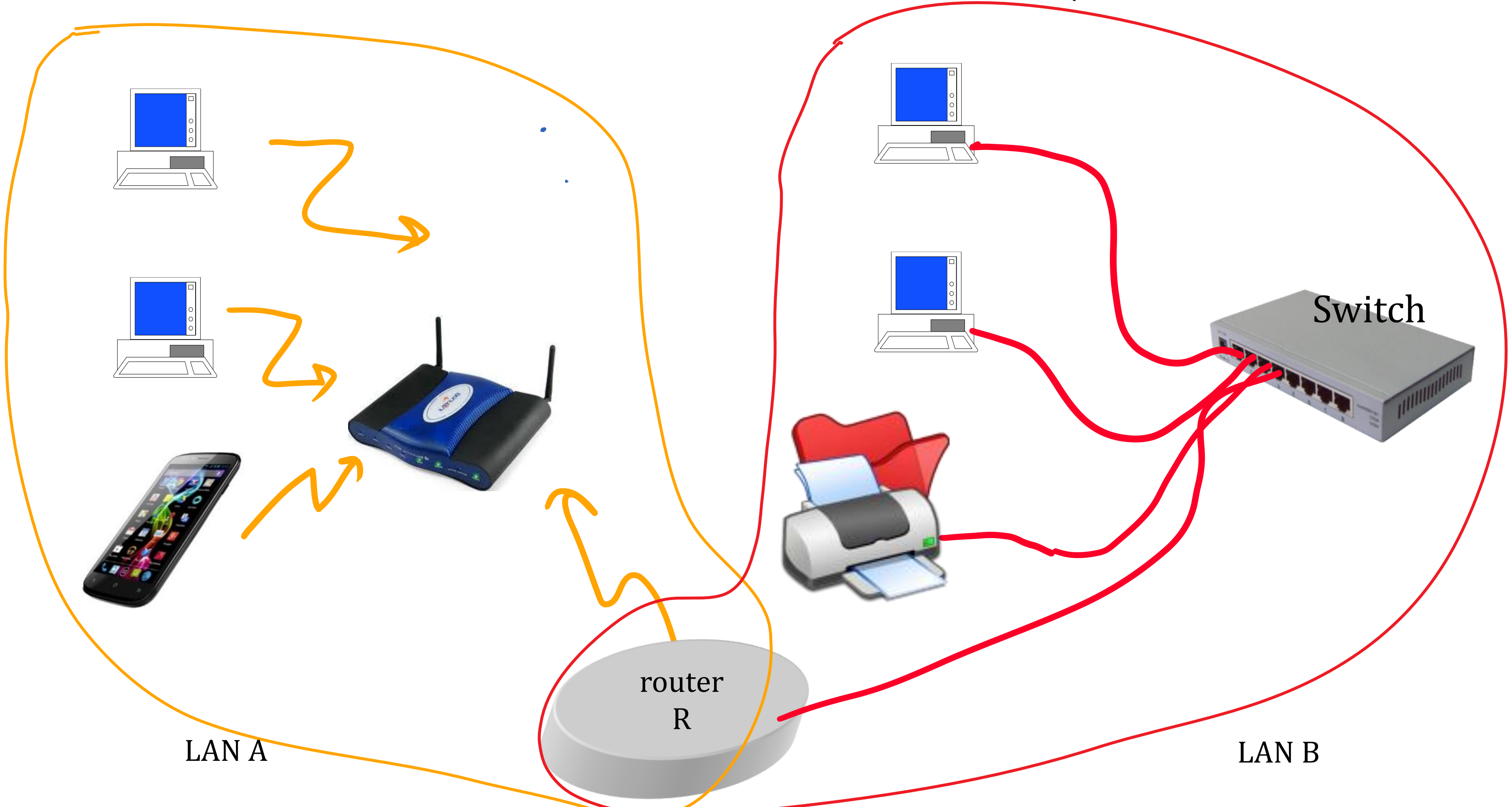


Ethernet LAN

How MAC and IP interact

LANs can be interconnected by ***routers*** = devices that forward packets based on IP addresses

Every machine must know the IP address of the next router (called "default gateway")

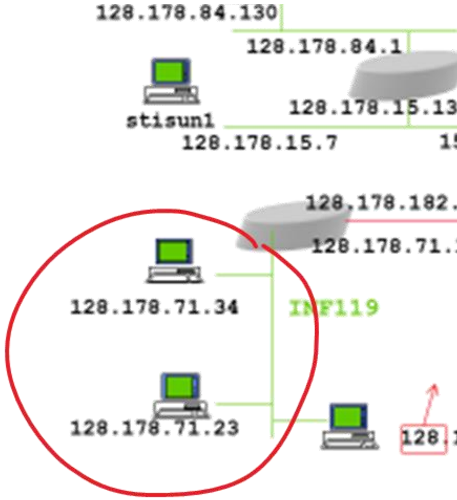


Network Masks

For IP, LAN = subnetwork

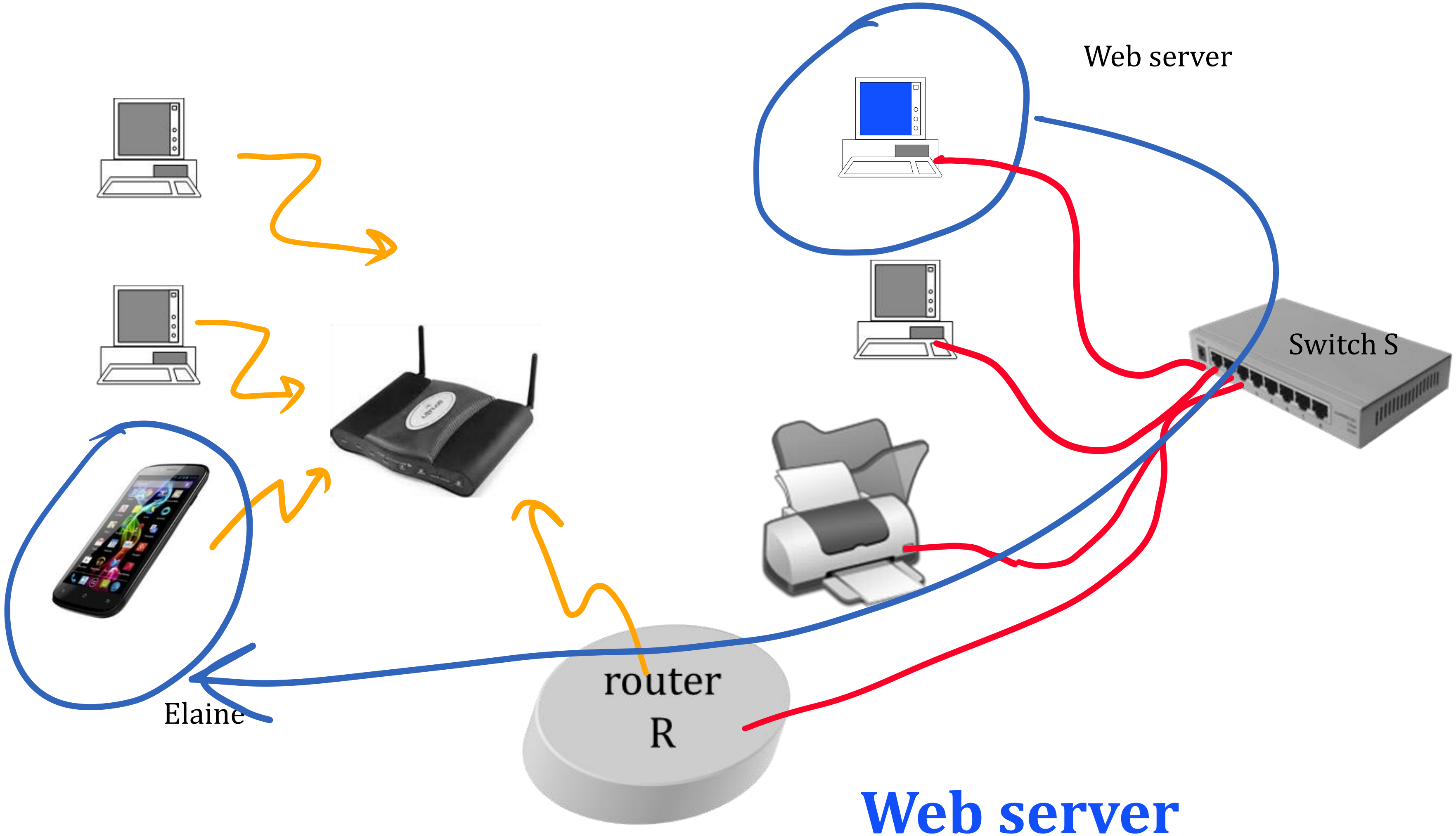
The IP addresses of all machines in one subnetwork must have same subnet prefix ex: 128.178.71

The size (in bits) of the subnet prefix is not always the same; must be specified in the configuration;
EPFL-IPv4: 24 bits: Example: 128.178.71.34 /24
ETHZ IPv4: 26 bits
At EPFL/IPv6: 64 bits: Example 2001:620:618:1a6:0a00:20ff:fe78:30f9/64



The size of IP subnet prefix is (still) often specified using a network mask
Mask = sequence of bits where 1s indicate the position of the prefix.
EPFL-IPv4, network mask is
1111 1111 1111 1111 1111 1111 0000 0000
which is written in decimal notation as 255.255.255.0;
Example: address =128.178.71.34, mask =255.255.255.0

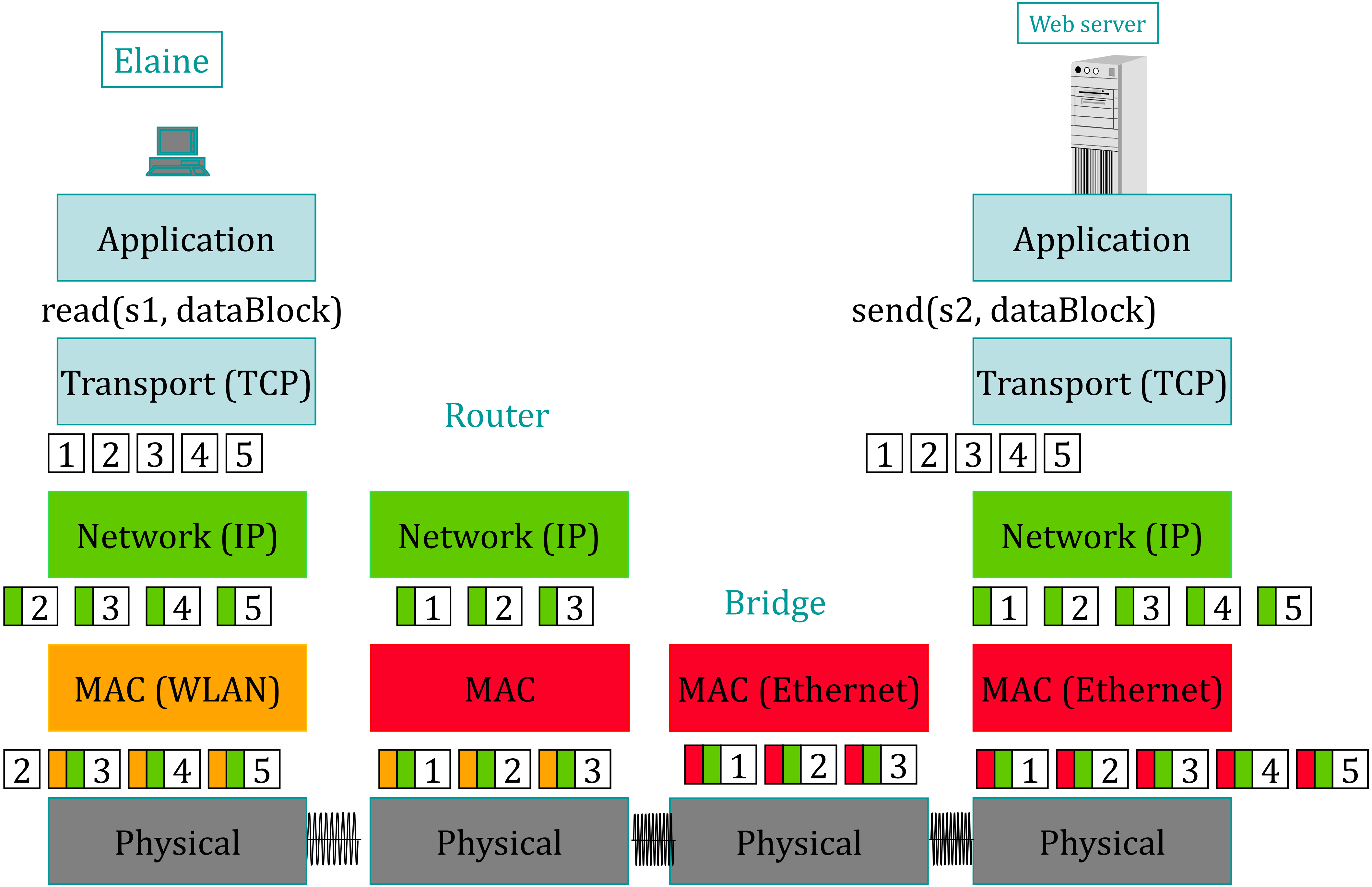
Putting Things Together



Web server sends a file to Elaine

Router = a system (or program) that forwards packets based on IP addresses

Bridge = Switch = a system (or program) that forwards packets based on MAC addresses



Elaine

Web server

Application

Application

read(s1, dataBlock)

send(s2, dataBlock)

Transport (TCP)

Transport (TCP)

Router

1 2 3 4 5

1 2 3 4 5

Network (IP)

Network (IP)

Network (IP)

2 3 4 5

1 2 3

Bridge

1 2 3 4 5

MAC (WLAN)

MAC

MAC (Ethernet)

MAC (Ethernet)

2 3 4 5

1 2 3

1 2 3

1 2 3 4 5

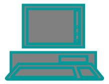
Physical

Physical

Physical

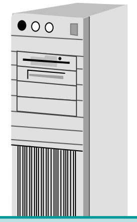
Physical

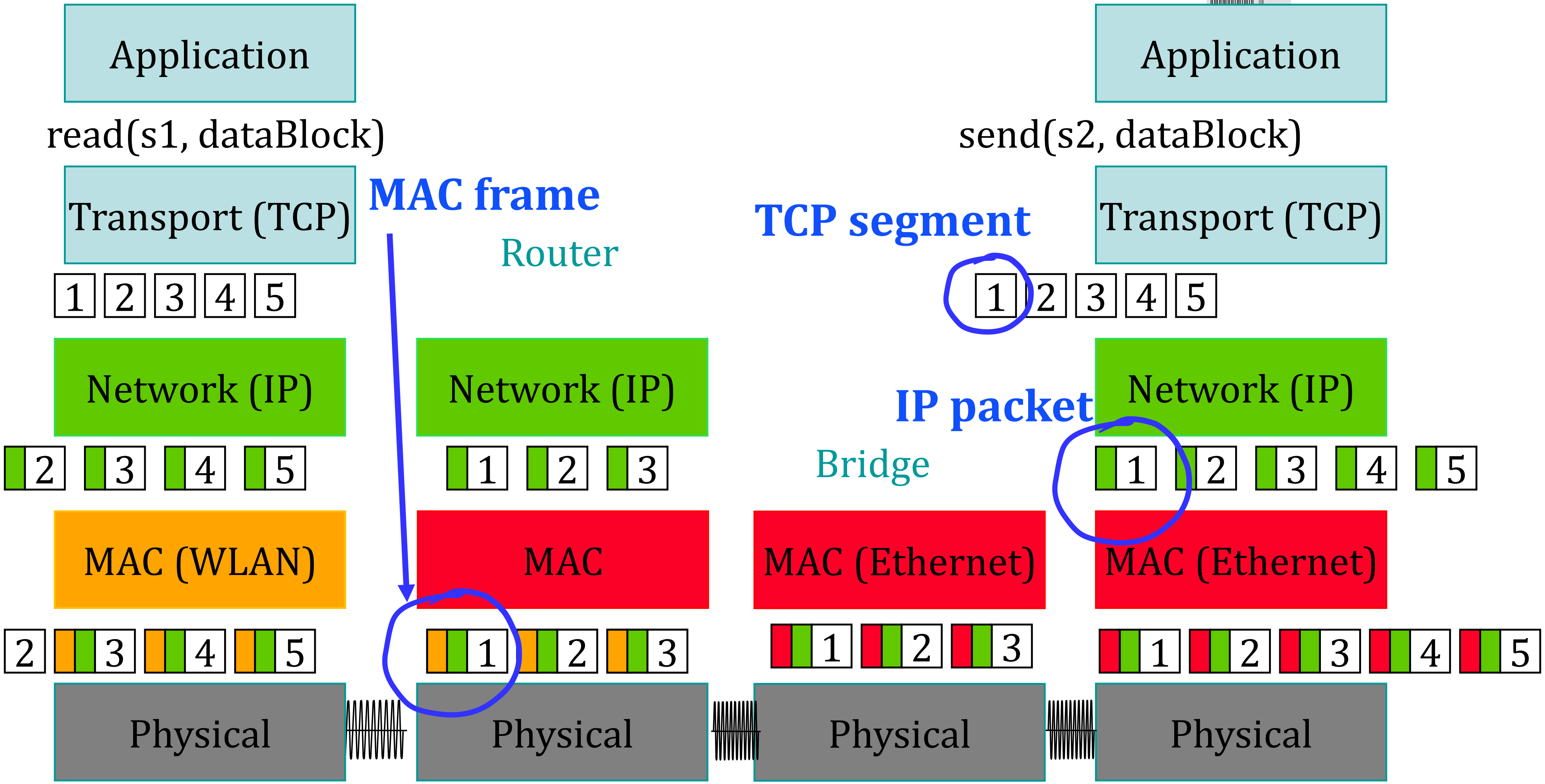
Elaine



Terminology

Web server





Application

read(s1, dataBlock)

Transport (TCP)

1 2 3 4 5

Network (IP)

2 3 4 5

MAC (WLAN)

2 3 4 5

Physical

MAC frame

Router

Network (IP)

1 2 3

MAC

1 2 3

Physical

TCP segment

IP packet

Bridge

MAC (Ethernet)

1 2 3

Physical

Application

send(s2, dataBlock)

Transport (TCP)

1 2 3 4 5

Network (IP)

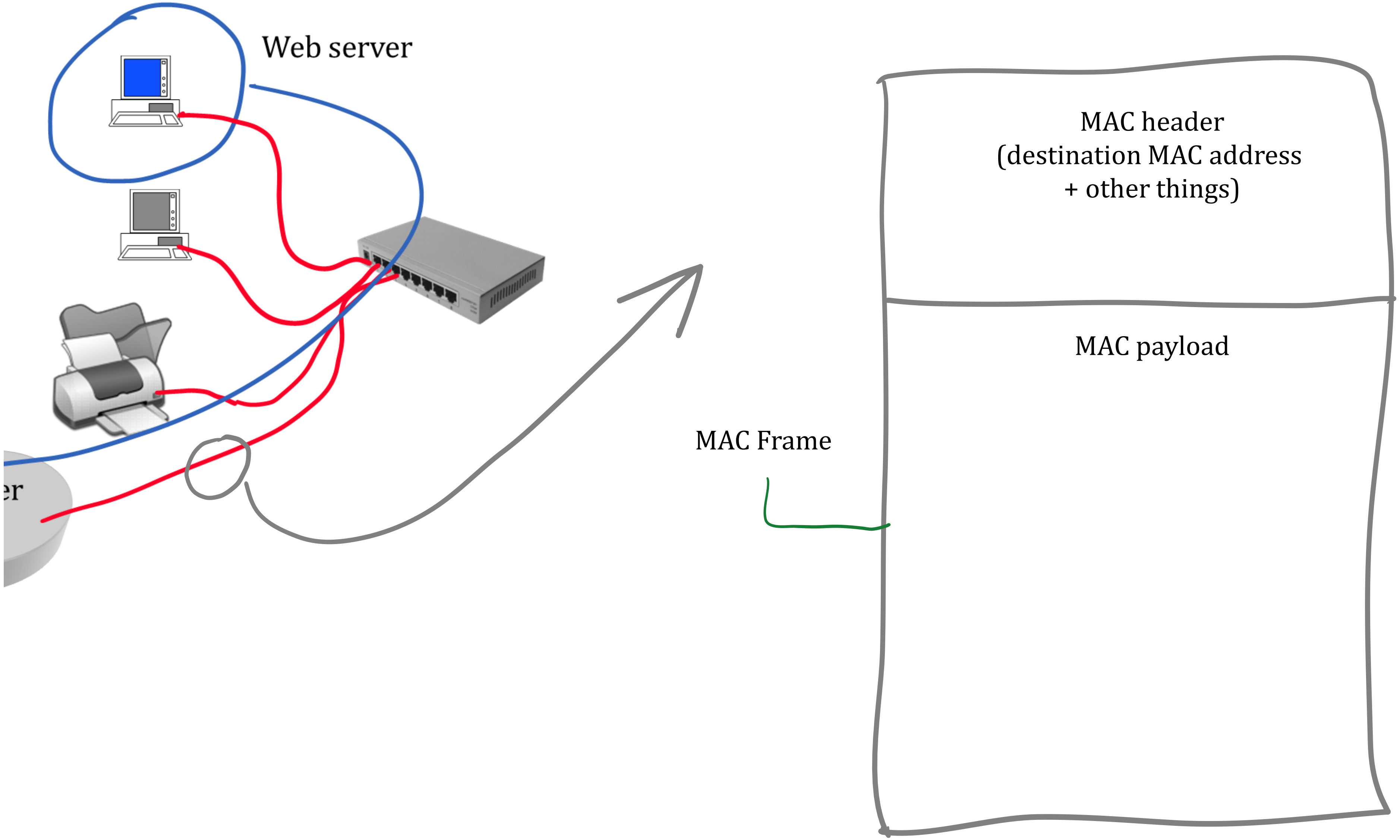
1 2 3 4 5

MAC (Ethernet)

1 2 3 4 5

Physical

The Onion View: header and payload

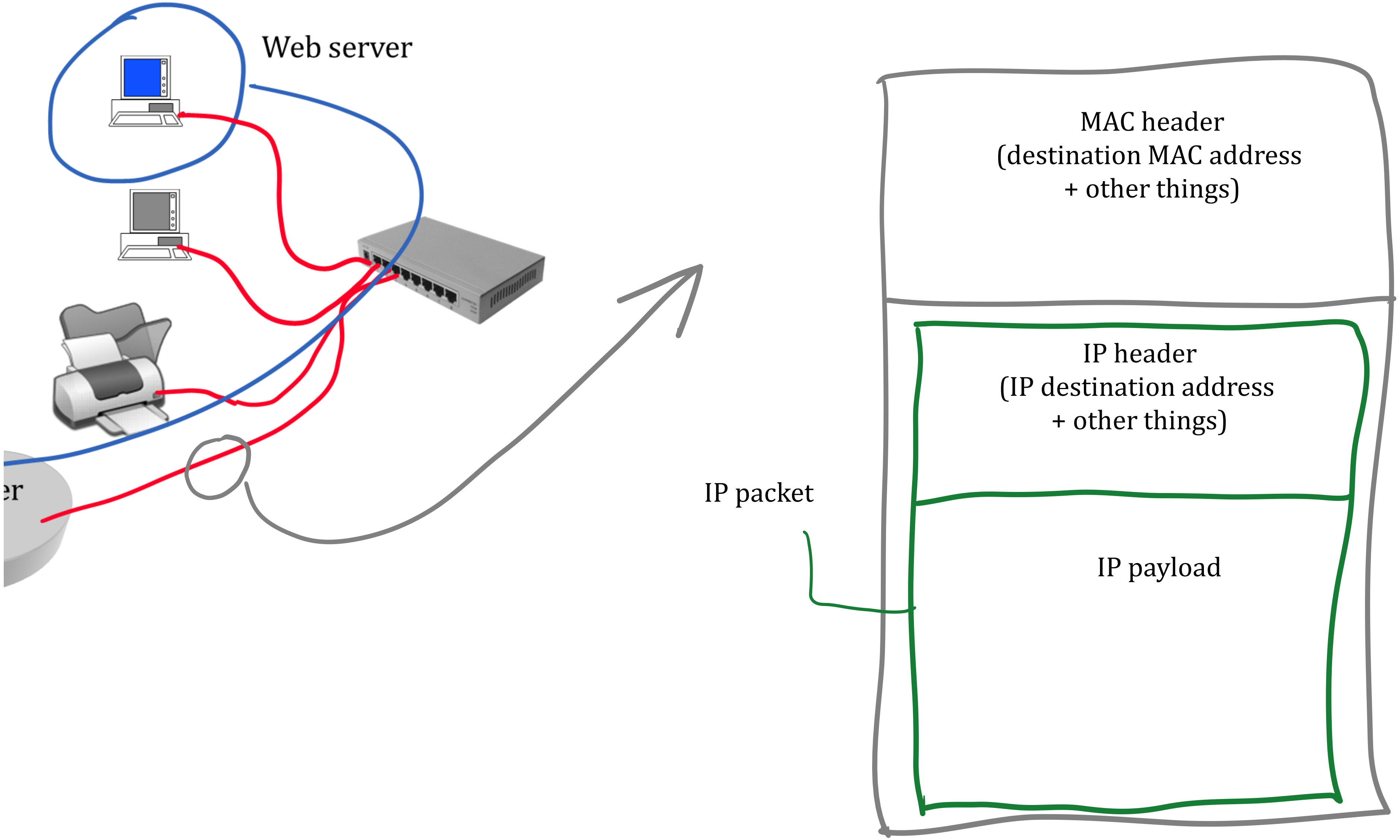


MAC header
(destination MAC address
+ other things)

MAC payload

MAC Frame

The Onion View: header and payload

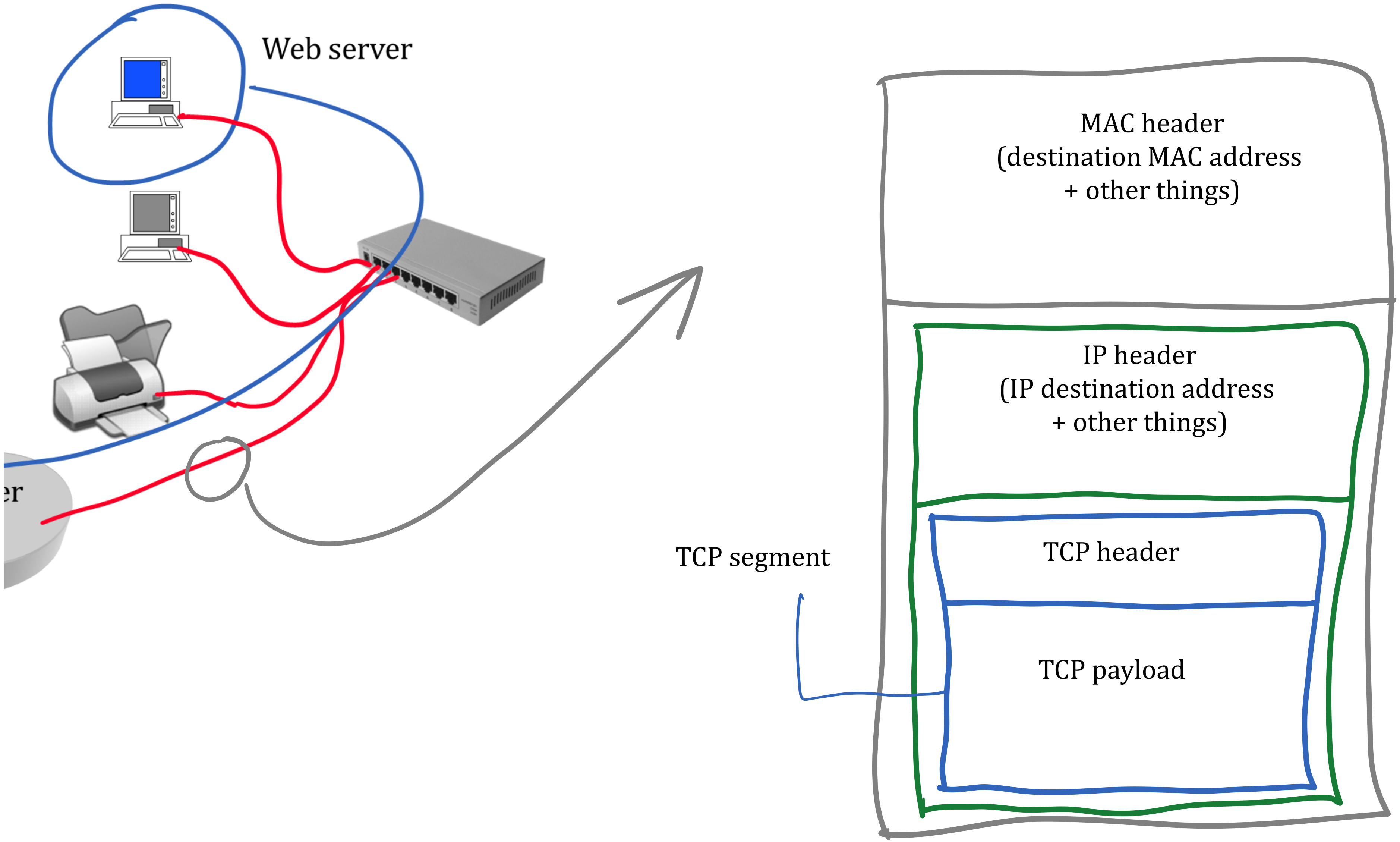


MAC header
(destination MAC address
+ other things)

IP header
(IP destination address
+ other things)

IP payload

The Onion View: header and payload



Web server

er

TCP segment

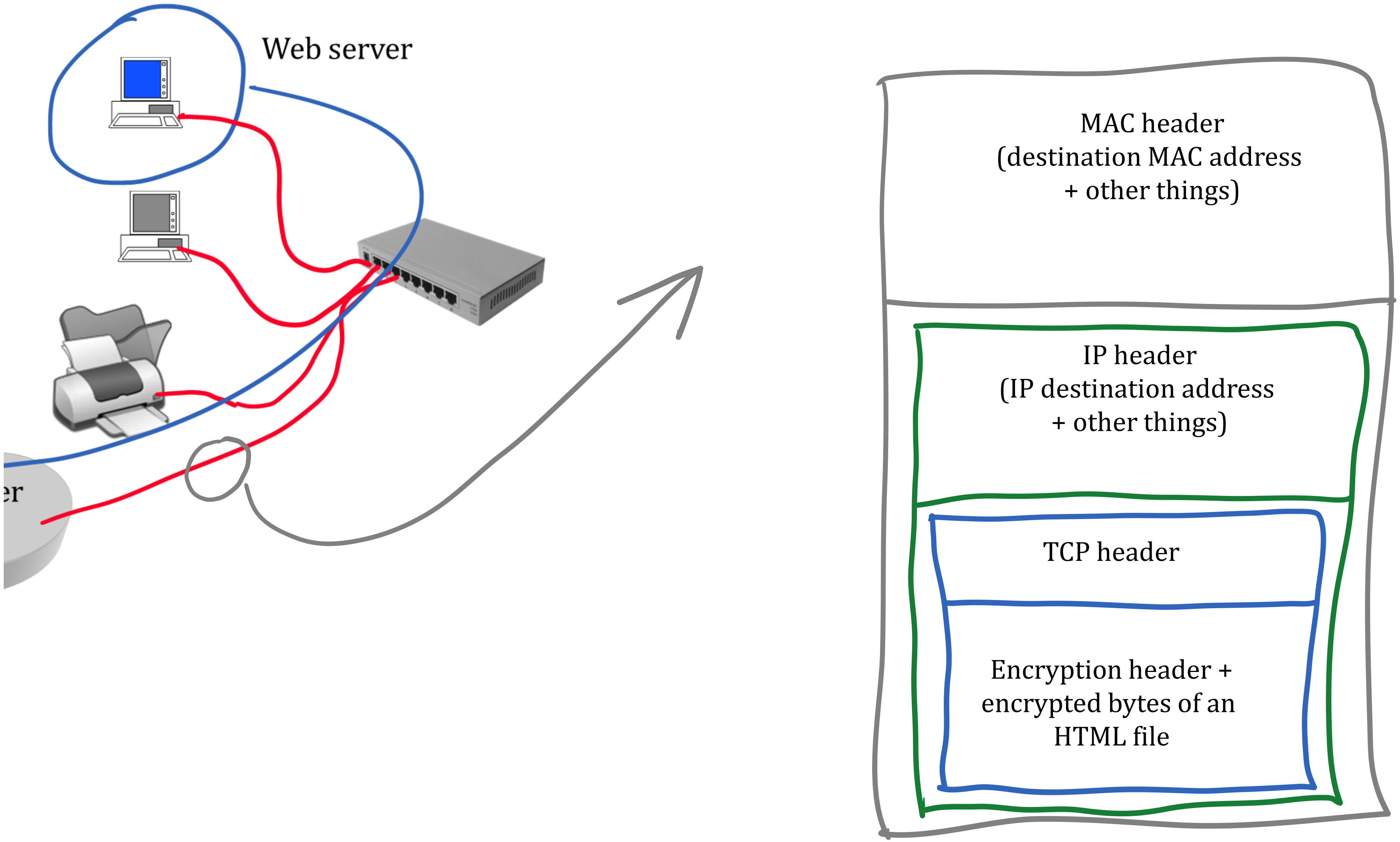
MAC header
(destination MAC address
+ other things)

IP header
(IP destination address
+ other things)

TCP header

TCP payload

The Onion View: header and payload



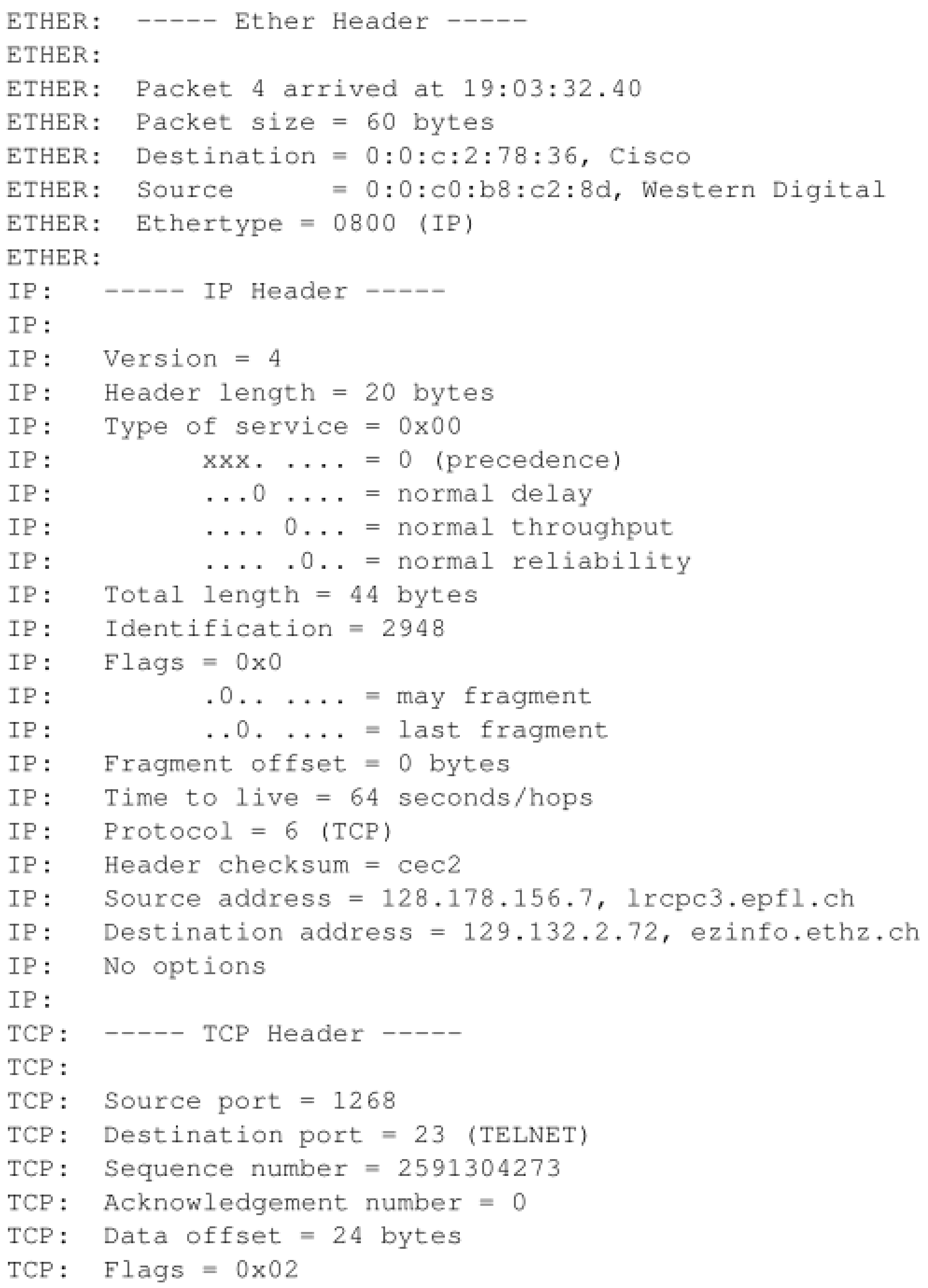
MAC header
(destination MAC address
+ other things)

IP header
(IP destination address
+ other things)

TCP header

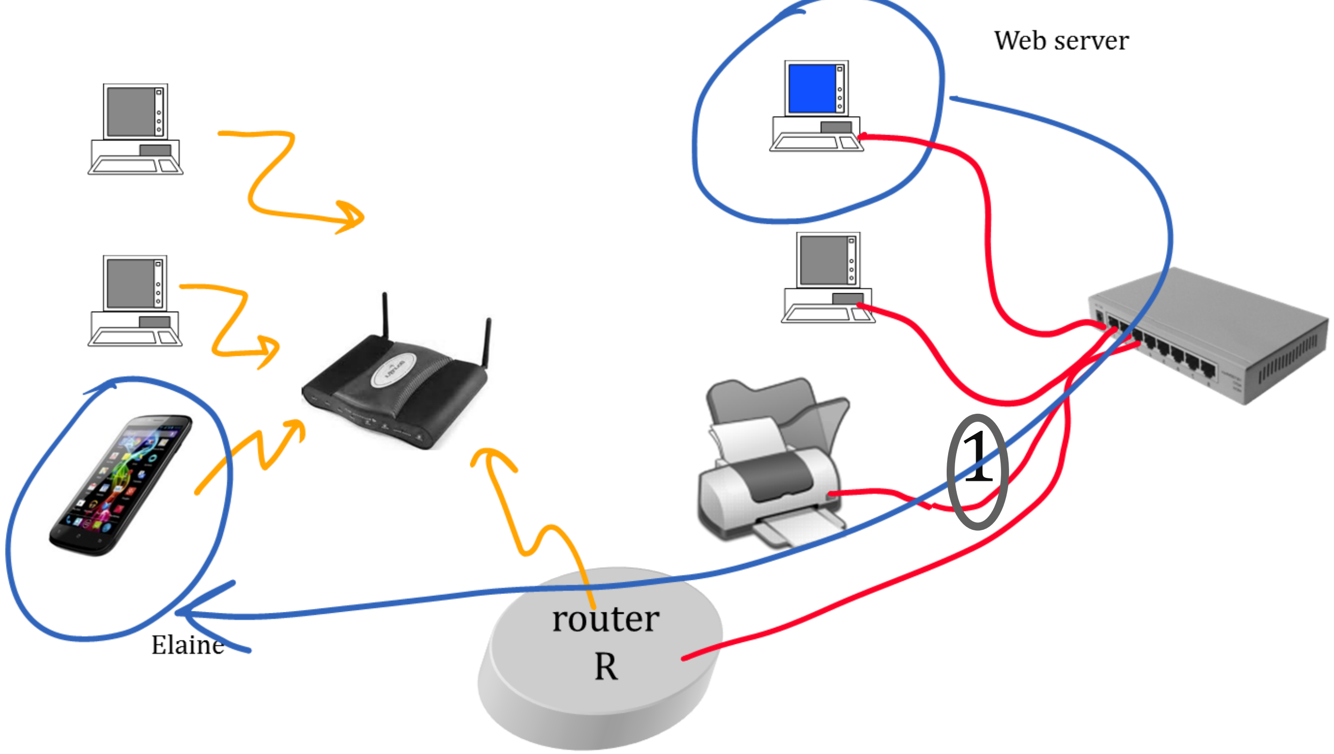
Encryption header +
encrypted bytes of an
HTML file

A Packet captured and prettily displayed



```
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 4 arrived at 19:03:32.40
ETHER:  Packet size = 60 bytes
ETHER:  Destination = 0:0:c:2:78:36, Cisco
ETHER:  Source      = 0:0:c0:b8:c2:8d, Western Digital
ETHER:  Ethertype = 0800 (IP)
ETHER:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:   Total length = 44 bytes
IP:   Identification = 2948
IP:   Flags = 0x0
IP:         .0.. .... = may fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live = 64 seconds/hops
IP:   Protocol = 6 (TCP)
IP:   Header checksum = cec2
IP:   Source address = 128.178.156.7, lrcpc3.epfl.ch
IP:   Destination address = 129.132.2.72, ezinfo.ethz.ch
IP:   No options
IP:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 1268
TCP:  Destination port = 23 (TELNET)
TCP:  Sequence number = 2591304273
TCP:  Acknowledgement number = 0
TCP:  Data offset = 24 bytes
TCP:  Flags = 0x02
```

We observe a packet from Web server to Elaine at 1; Say what is true



A. The destination MAC address is the MAC address of the router

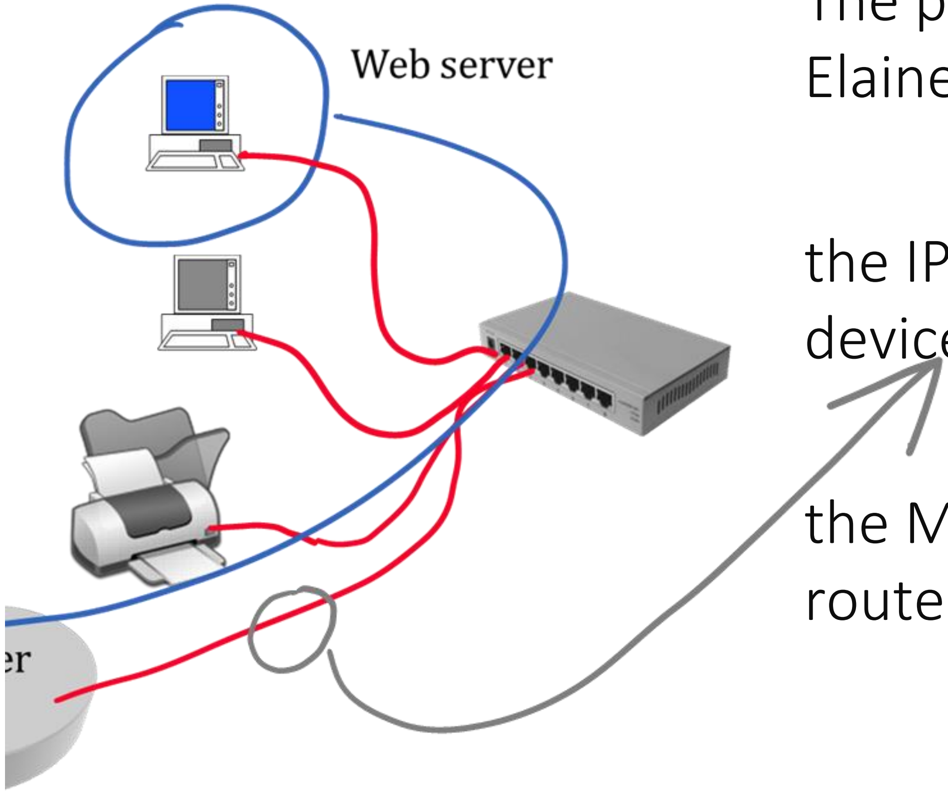
B. The destination IP address is the IP address of the router

C. Both A and B

D. None

E. I don't know

Solution



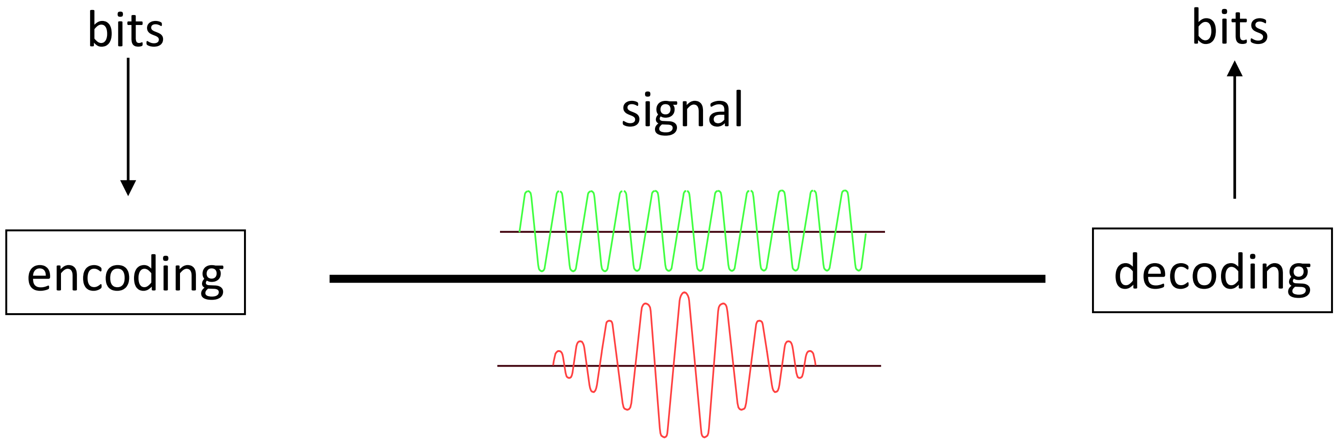
Answer A

The packet is sent by Web server to Elaine;

the IP destination address is Elaine’s device address (network layer is global)

the MAC destination address is the router’s MAC address (MAC layer is local)

Physical Layer Transforms Bits and Bytes into Electromagnetic Waves



Encoding of bits as physical signals, usually electromagnetic

Is technology specific: there are several Ethernet physical layers, several WLAN 802.11 physical layers

Acoustic instead of electromagnetic used under water

Bit rate of a channel = number of bits transmitted per time unit; is measured in **b/s**, 1 kb/s = 1000 b/s, 1 Mb/s = 10^6 b/s, 1Gb/s=10^9 b/s; also (improperly) called “bandwidth”

Bit Rate and Bandwidth

The bit rate of a channel is the number of bits per second. The bandwidth is the width of the frequency range that can be used for transmission over the channel. The bandwidth limits the maximal bit rate that can be obtained using a given channel. Information theory gives a bound on the achievable bit rate on a given channel.

For example: Shannon-Hartley law: for a channel of bandwidth B (Hz) submitted to Gaussian noise, the capacity in b/s is:

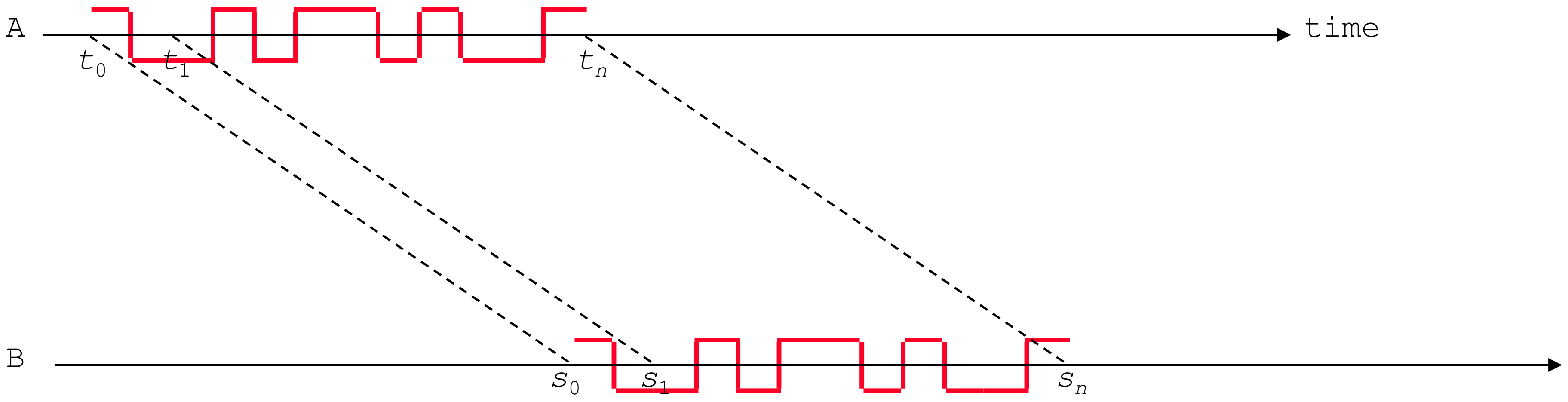
$$C = B \log2(1 + SNR)$$

with SNR= signal to noise ratio (ratio of power of emitted signal over power of noise); for example: ADSL Line: $B = 1$ MHz, SNR =45 dB, C =15 Mb/s

In computer science, many people use “bandwidth” instead of “bit rate

Propagation

Propagation between A and B = time for the head of signal to travel from A to B



$D = s_i - t_i = \frac{d}{c} = \frac{distance}{speed\ of\ light}$(propagation delay for non acoustic channels)

In copper: *c*= 2.3e+08 m/s; in glass optical fiber: *c*= 2e+08 m/s;

Rule of thumb: 5 µs/km; around the globe = 200 msec

Time it takes to send one packet of 1kB (8000bits)

	data center	*ADSL*	*modem*	*Internet*
distance	20 m	2 km	20 km	20’000 km
bit rate	1Tb/s	10Mb/s	10kb/s	1Mb/s
propagation	0.1μs	0.01ms	0.1ms	100ms
transmission	0.008 μs	0.8ms	800ms	8ms
total	0.108 μs	0.81ms	800.1ms	108ms

Throughput

Throughput = number of useful data bits / time unit

It is *not* the same as the bit rate. Why ?

- protocol overhead: all protocols like UDP use some extra bytes to transmit protocol information.
- protocol waiting times.

Same units as a bit rate

- b/s, kb/s, Mb/s

Pigeon outruns South African ADSL

11 September 2009 | 14:28

A South African information technology company has proved it's faster for them to send data by carrier pigeon than using the country's leading internet provider.

A South African information technology company has proved it's faster for them to transmit data by carrier pigeon than to send it using Telkom, the country's leading internet service provider.

Internet speed and connectivity in Africa's largest economy are poor because of a bandwidth shortage. It is also expensive.

Winston the pigeon has easily outpaced South Africa's leading broadband network it moving data (AAP)

An 11-month-old pigeon, Winston, took one hour and eight minutes to fly the 80 km (50 miles) from Unlimited IT's offices near Pietermaritzburg to the coastal city of Durban with a data card strapped to its leg.

Including downloading, the transfer took two hours, six minutes and 57 seconds – the time it took for only four percent of the data to be transferred using a Telkom line.

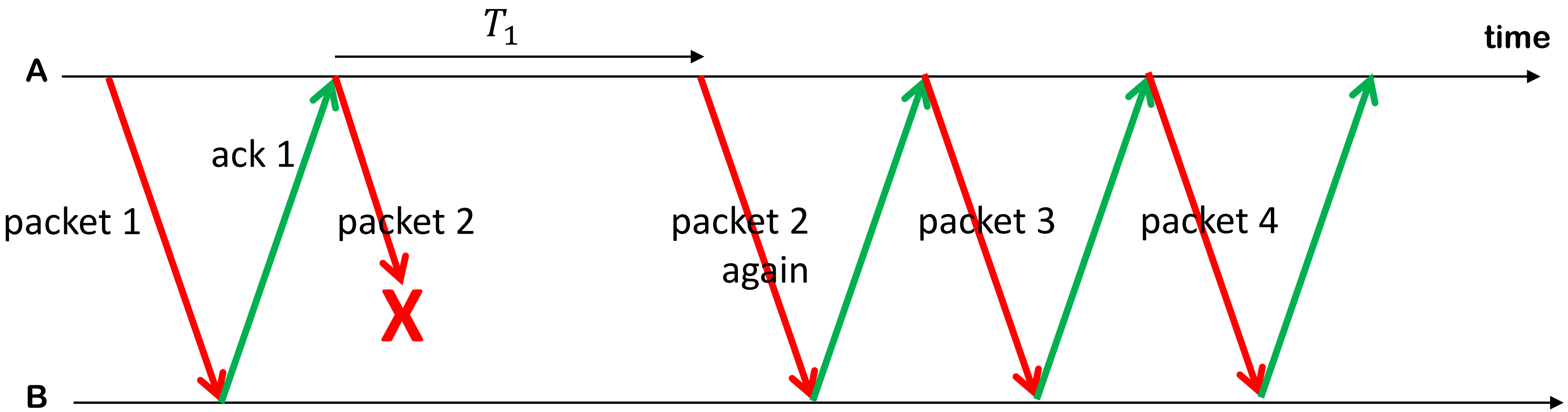
Example. The Stop and Go Protocol

A simple protocol used to repair packet losses.

A sends packets to B; B returns an acknowledgement packet immediately to confirm that B has received the packet;

A waits for acknowledgement before sending a new packet; if no acknowledgement comes after a delay T_1, then *A* retransmits

What is the throughput when there is no loss ?



Performance of the Stop and Go Protocol

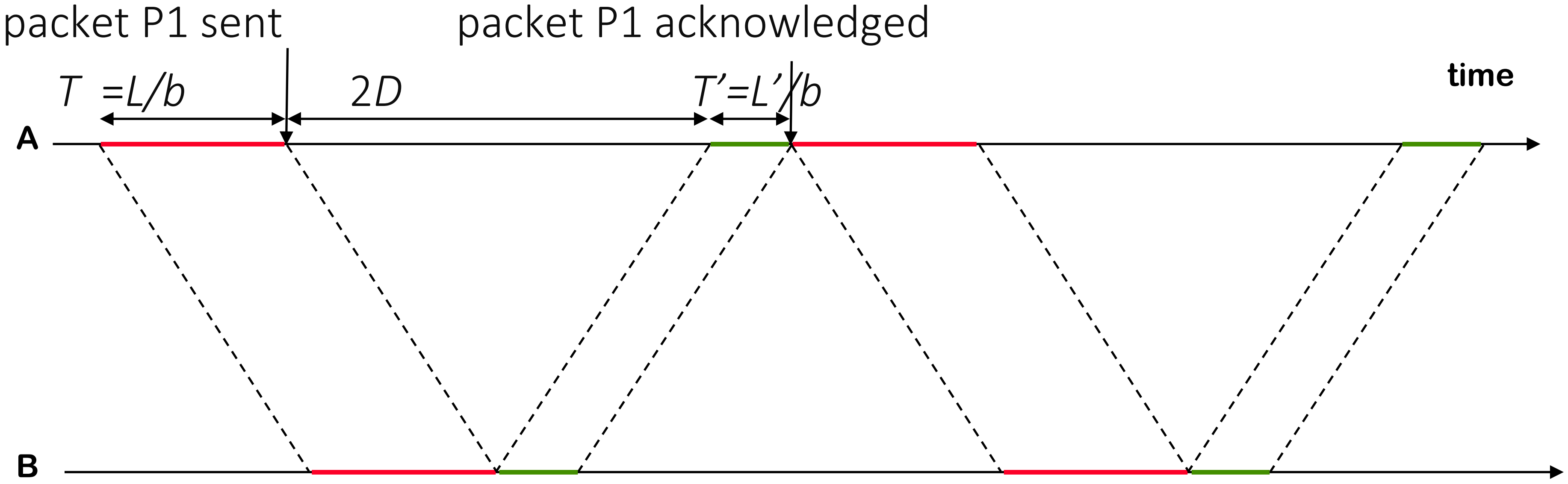
L = packet size; b = channel bit rate; D = propagation delay

Best case: always one packet to transmit, no loss.

- In one cycle, L useful bits are transmitted.
- The cycle lasts $T + 2D + T'$.
- Throughput $= \frac{L}{T+2D+T'} = \frac{b}{1+\frac{L'}{L}+\frac{2Db}{L}}$

overhead

“Bandwidth-Delay Product”



Throughput of Stop and Go

	data center	*ADSL*	*modem*	*Internet*
distance	20 m	2 km	20 km	20’000 km
bit rate	1Tb/s	10Mb/s	10kb/s	1Mb/s
propagation	0.1μs	0.01ms	0.1ms	100ms
transmission	0.008 μs	0.8ms	800ms	8ms
total	0.108 μs	0.81ms	800.1ms	108ms
bw delay product	200kb	200b	2b	200kb
throughput of Stop and Go*	3.8%	97.56%	99.98%	3.8%

We will see that TCP does better than Stop and Go by using a smarter scheme (sliding window)

* with packets of size 1kB=8’000 bits and assuming overhead is negligible