

# ***INTRO TO NETWORKING***

## **PART 1: Basic concepts (full)**





# Agenda

- Connections
- Concept of Packet
- Network Stack Models (TCP/IP - ISO/OSI)
- Internet Protocol and IP Address Space
- Ethernet and Physical Address
- Speed, Bandwidth, Latency, Throughput
- High Speed (and Low Latency) Networks
- **LINUX** commands (configuration and diagnostic)

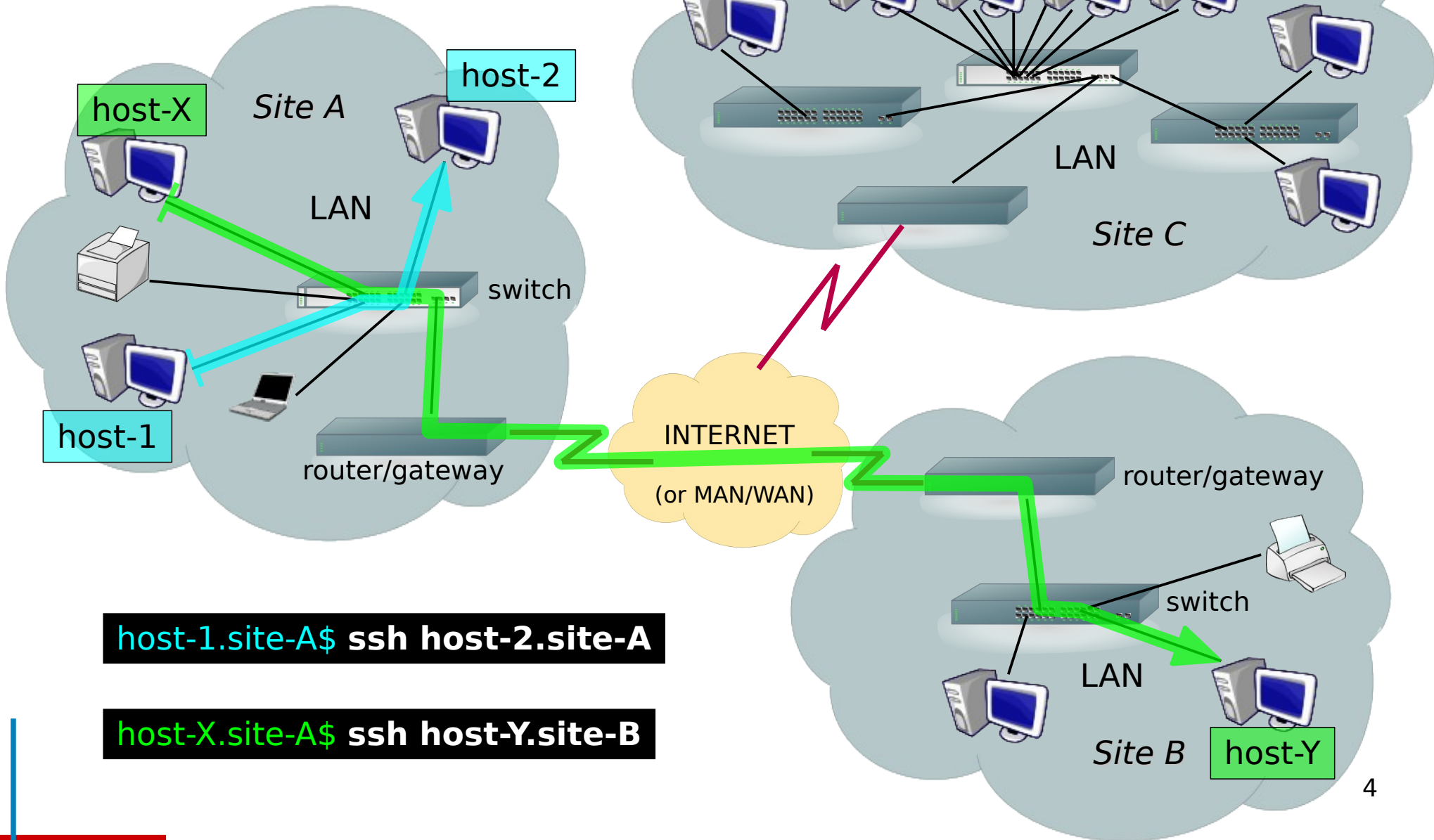


# Connections





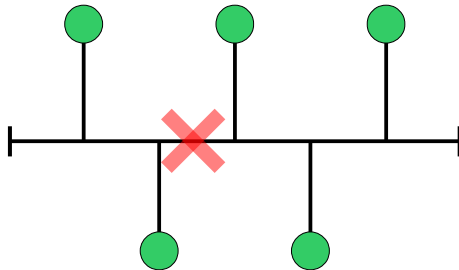
# Connections



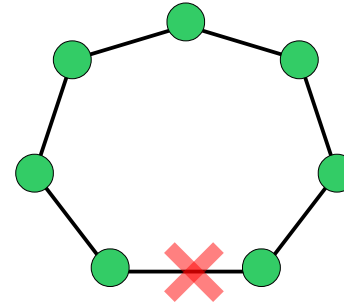


# Physical Network Topologies

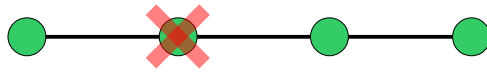
BUS



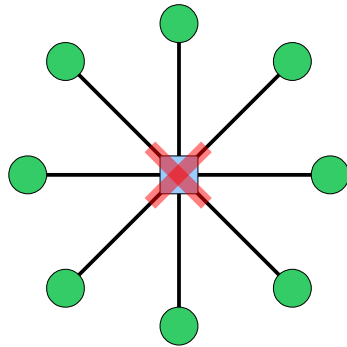
RING



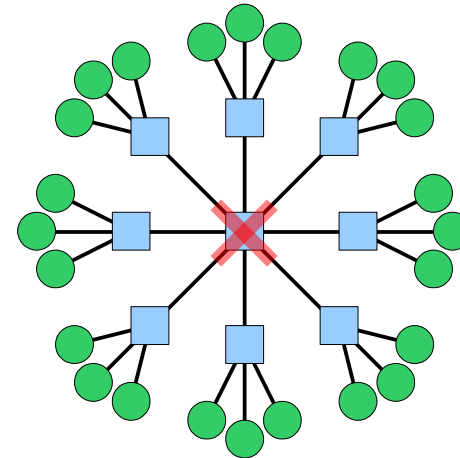
LINEAR



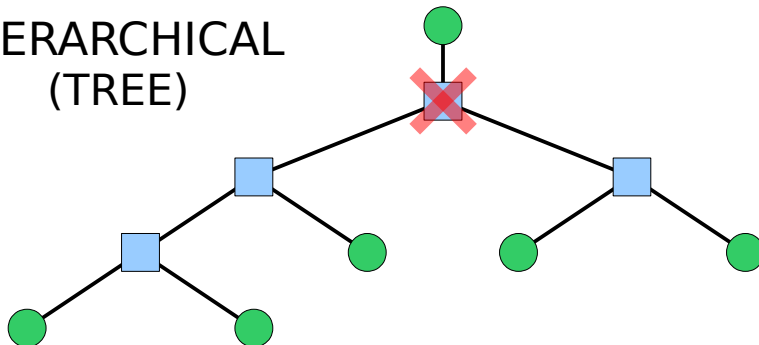
STAR



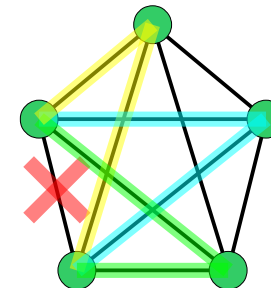
EXTENDED  
STAR



HIERARCHICAL  
(TREE)



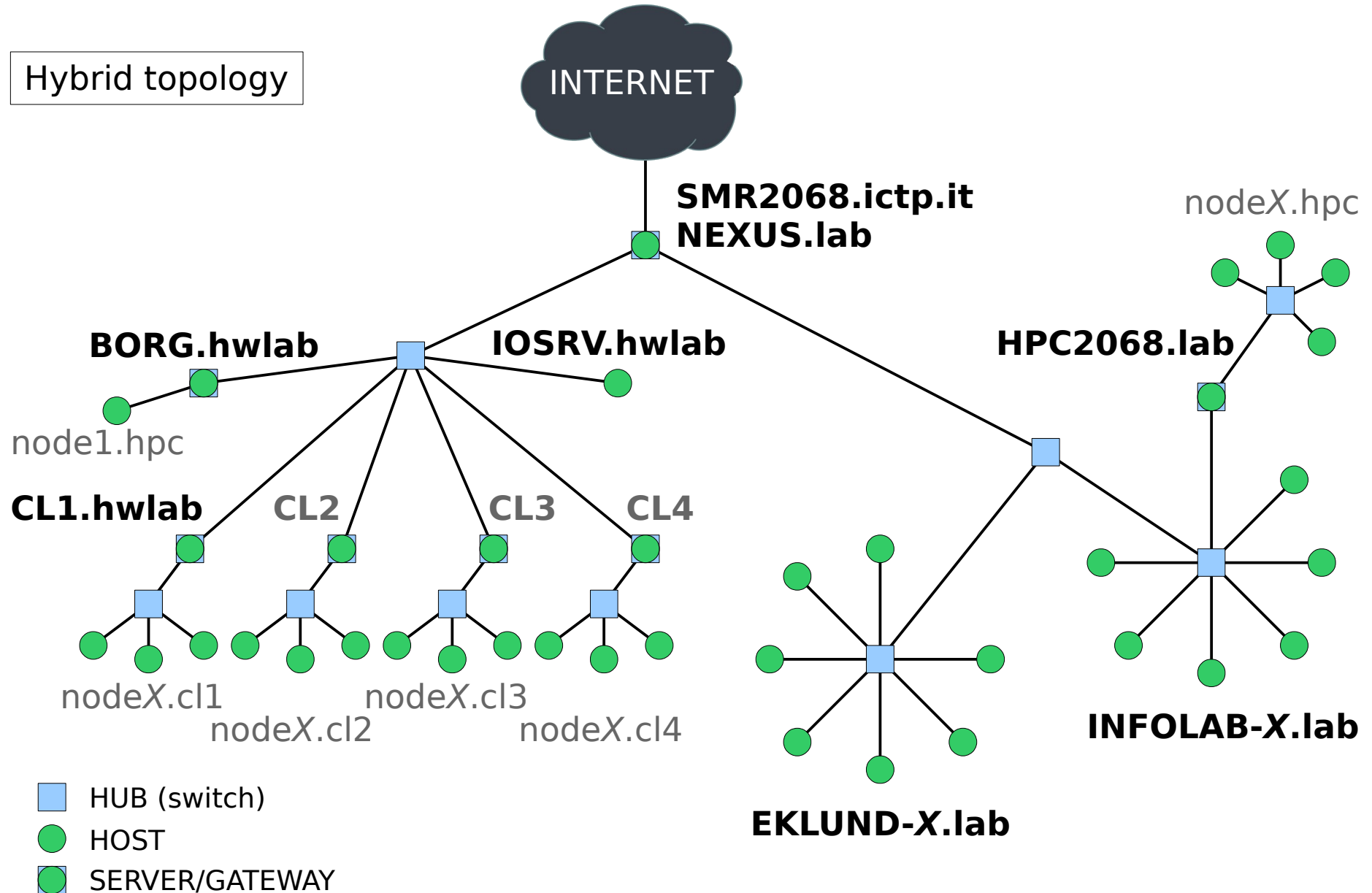
MESH  
(PARTIAL or FULLY  
CONNECTED)





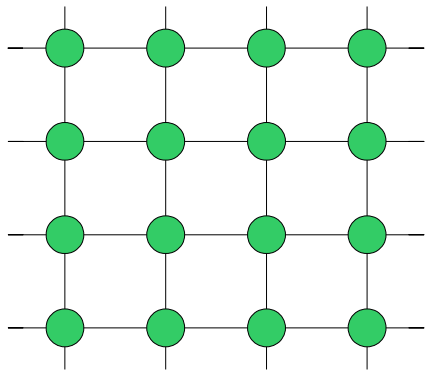
# Example: the lab network

Hybrid topology

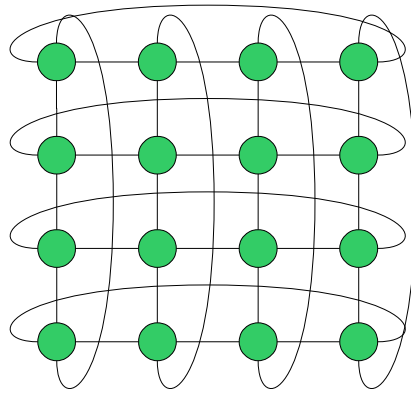




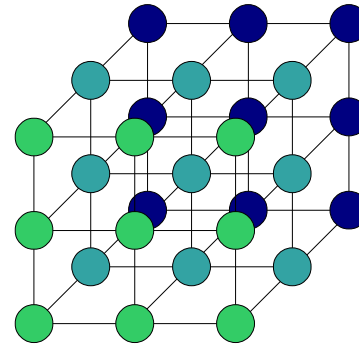
# Clustering topologies (HPC)



2D Mesh



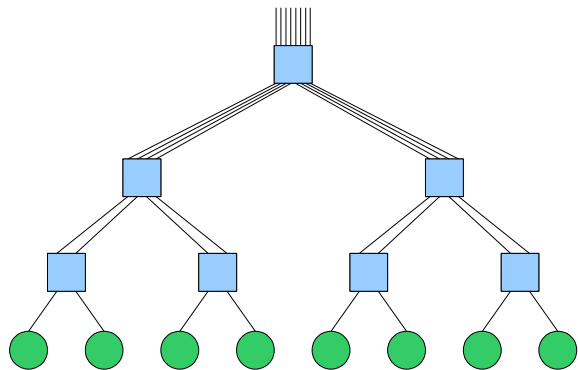
2D Torus



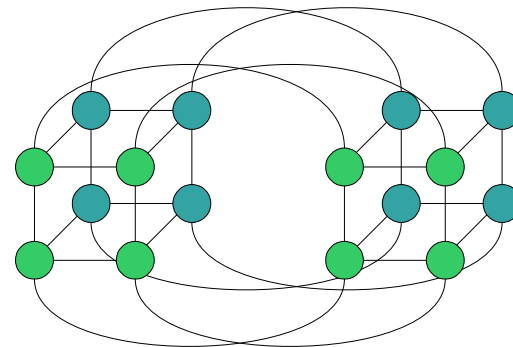
3D Mesh

*Imagine a 3D  
mesh with the  
ends connected.  
Thanks ;)*

3D Torus



FAT TREE



Hypercube (4-cube)



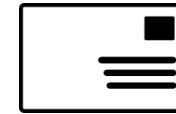
# Concept of Packet







# Addressing and Multiplexing

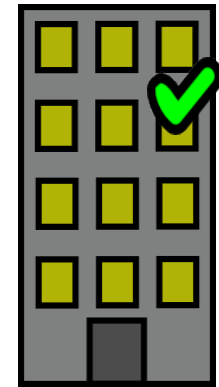


## From Address:

Country  
City  
Street and Number  
Name

## To Address:

Country  
City  
Street and Number  
Name/Apartment/Floor



0100110100010010



## Source Address:

hostname: **host-a**  
domain: **example.com**  
IP address: **192.0.32.10**  
protocol: **TCP**  
port: **35432**

## Destination Address:

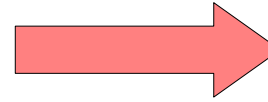
hostname: **host-b**  
domain: **example.org**  
IP address: **192.0.2.44**  
protocol: **TCP**  
port: **25 (SMTP)**



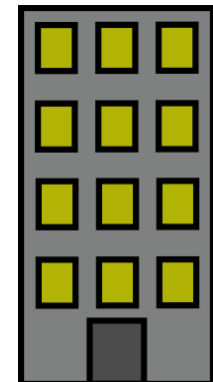
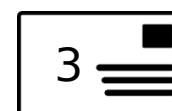
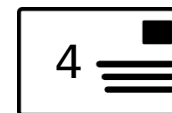
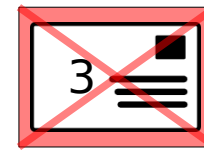
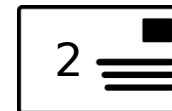
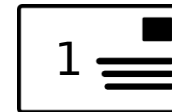
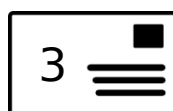
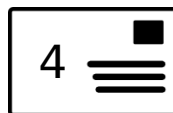
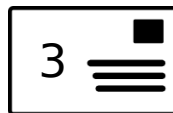
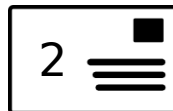
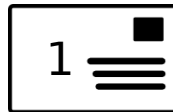
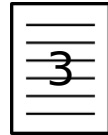
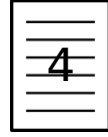
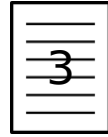
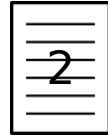
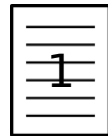


# Fragmentation and Windowing

1



2

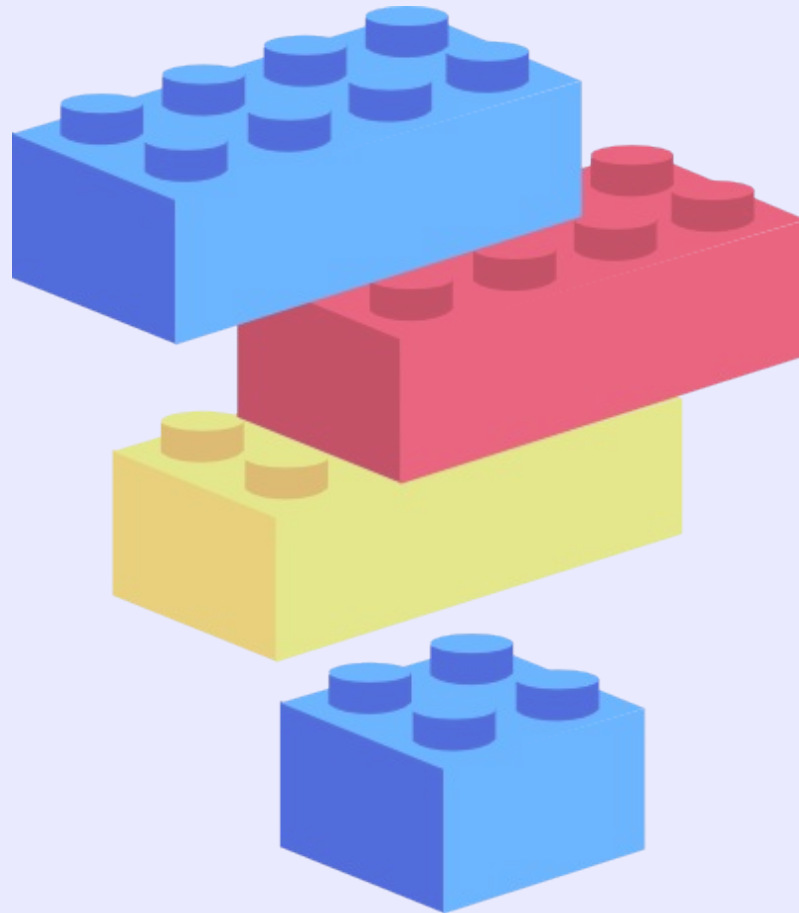


NETWORK CONNECTIONS ARE (OFTEN) NOT RELIABLE  
BANDWIDTH IS NOT FREE AND IS NOT UNLIMITED

In case of failure, sending twice a large amount of data has a cost, both in terms of money and time. Network protocols split and fragments the data stream, TCP uses sequence numbers to reassemble the data in case they reach the destination out of order (retransmission, timeout, different routes,...).



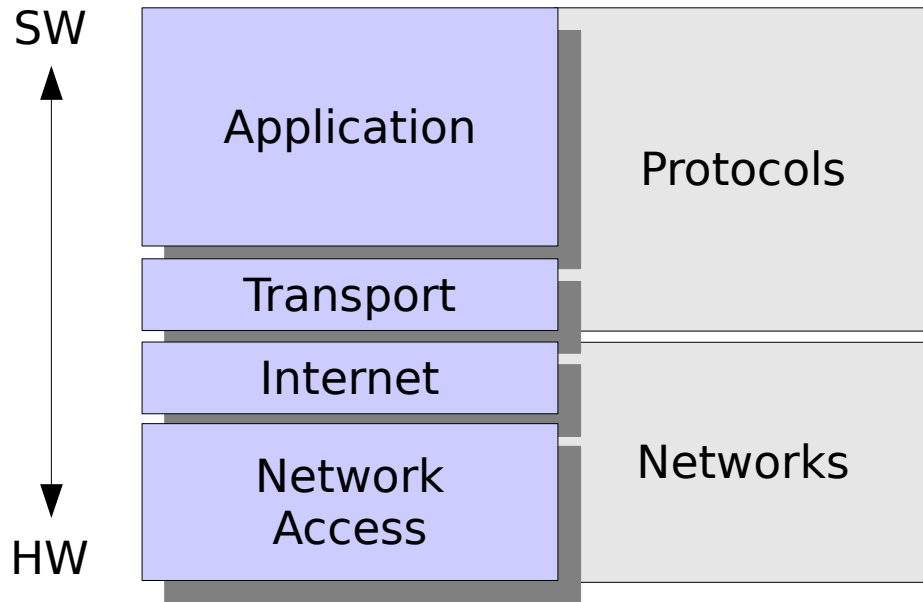
# Network Stack



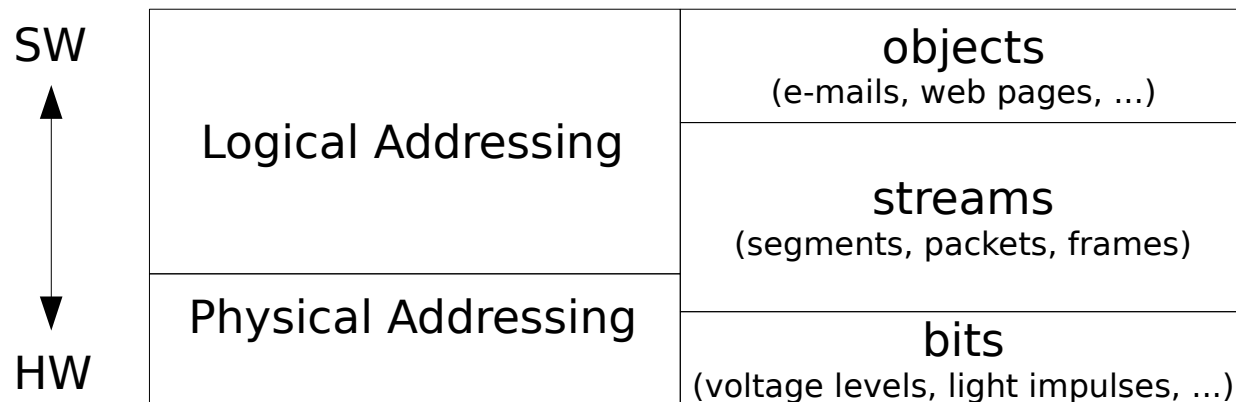
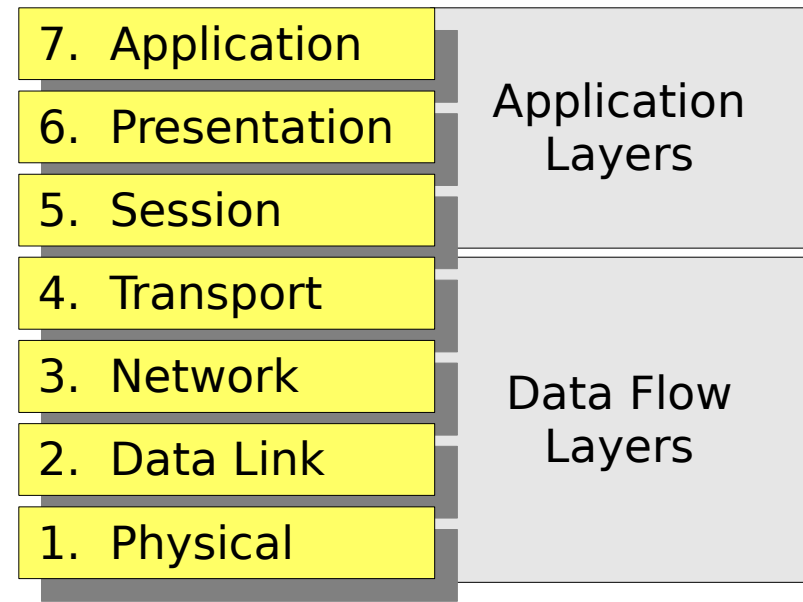


# Network Stack Models

## TCP/IP Model



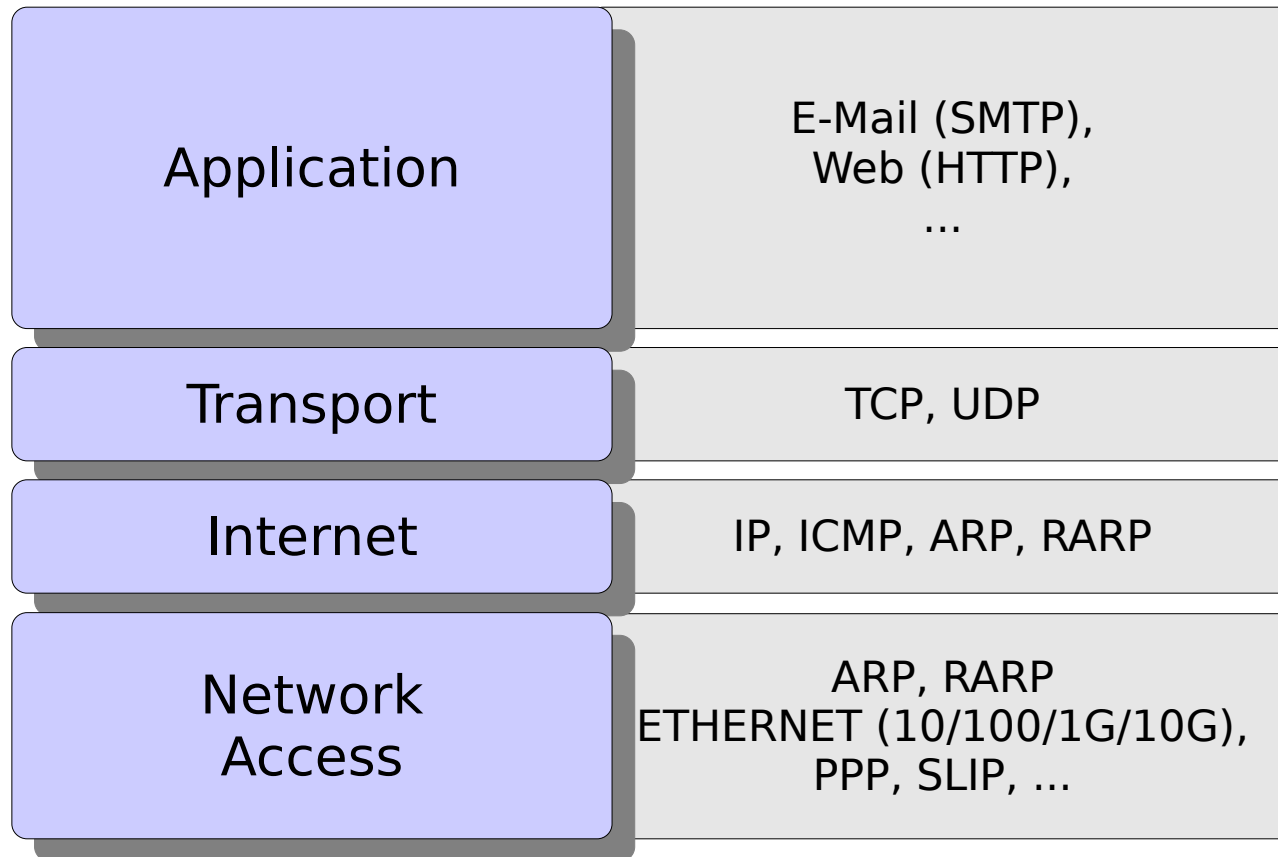
## ISO/OSI Model





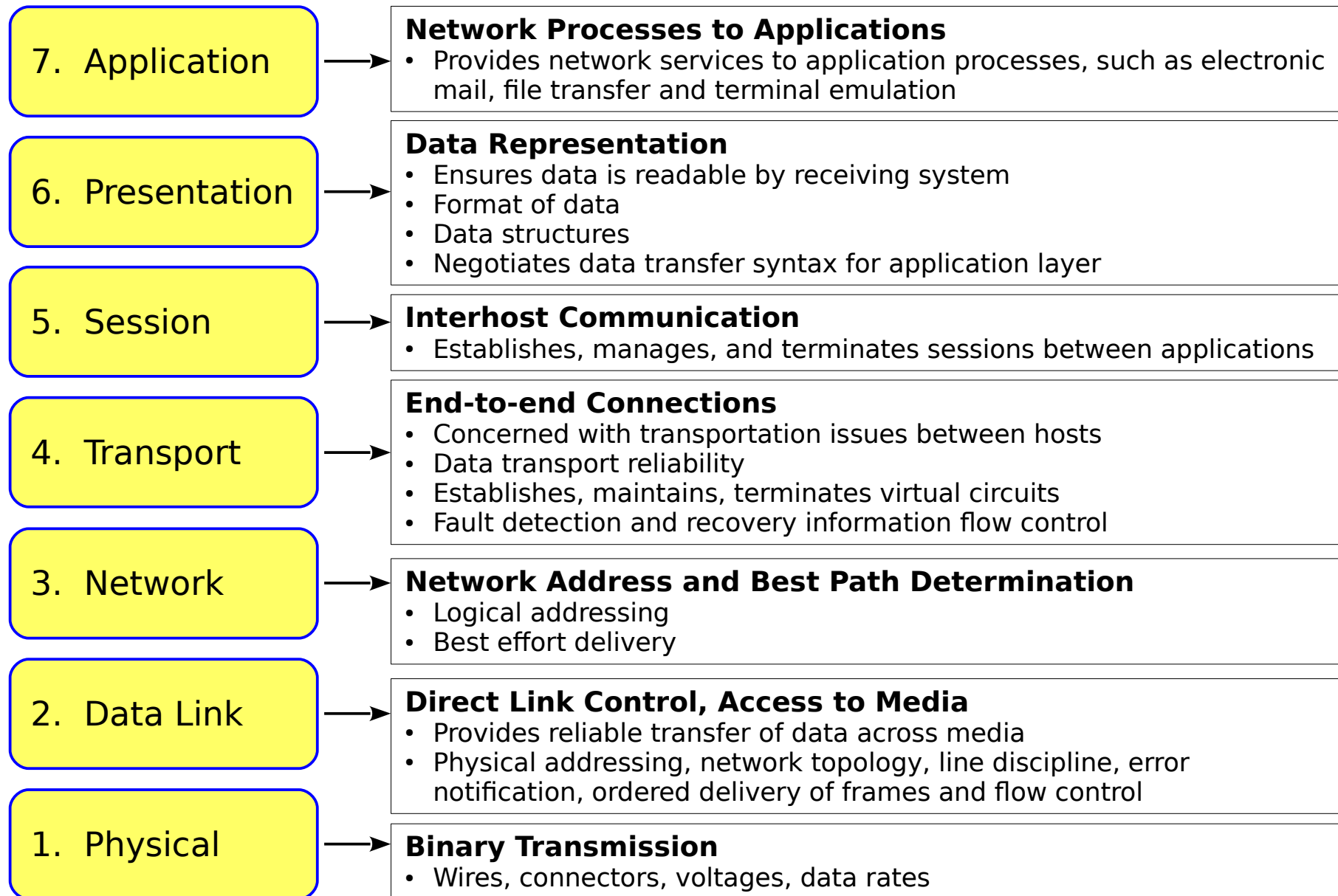
# TCP/IP Model

## Protocols



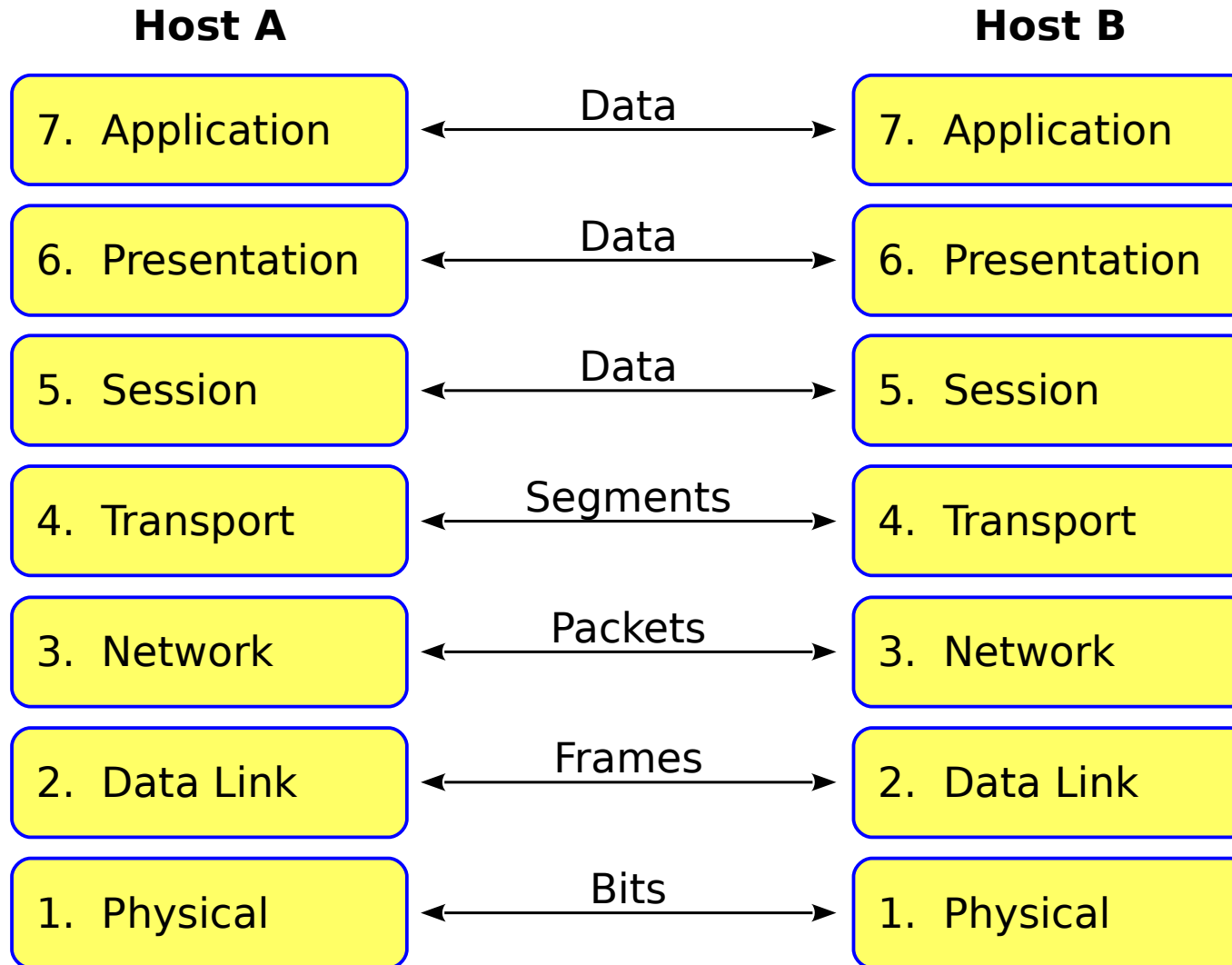


# ISO/OSI Model





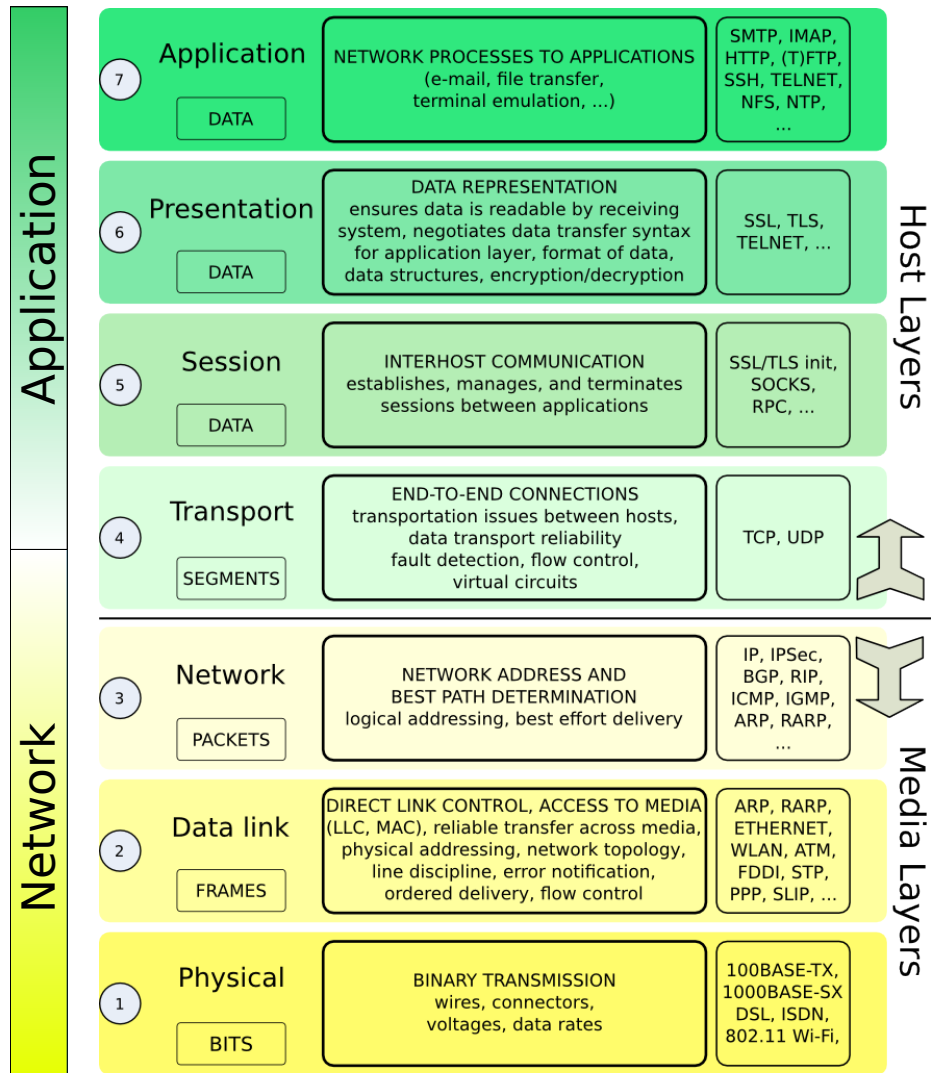
# Protocol Data Unit (PDU)



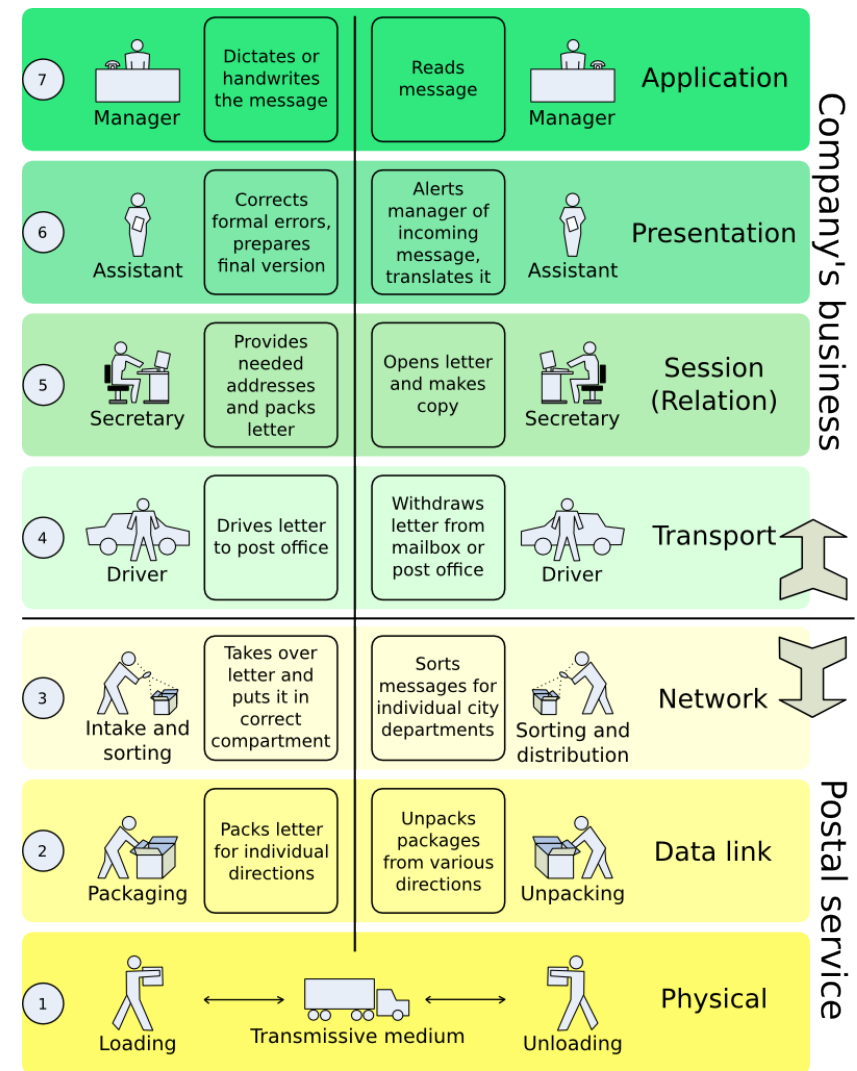
D-S-P-F-B



# OSI Model and snail-mail communication parallel



OSI 7-Layer Reference Model



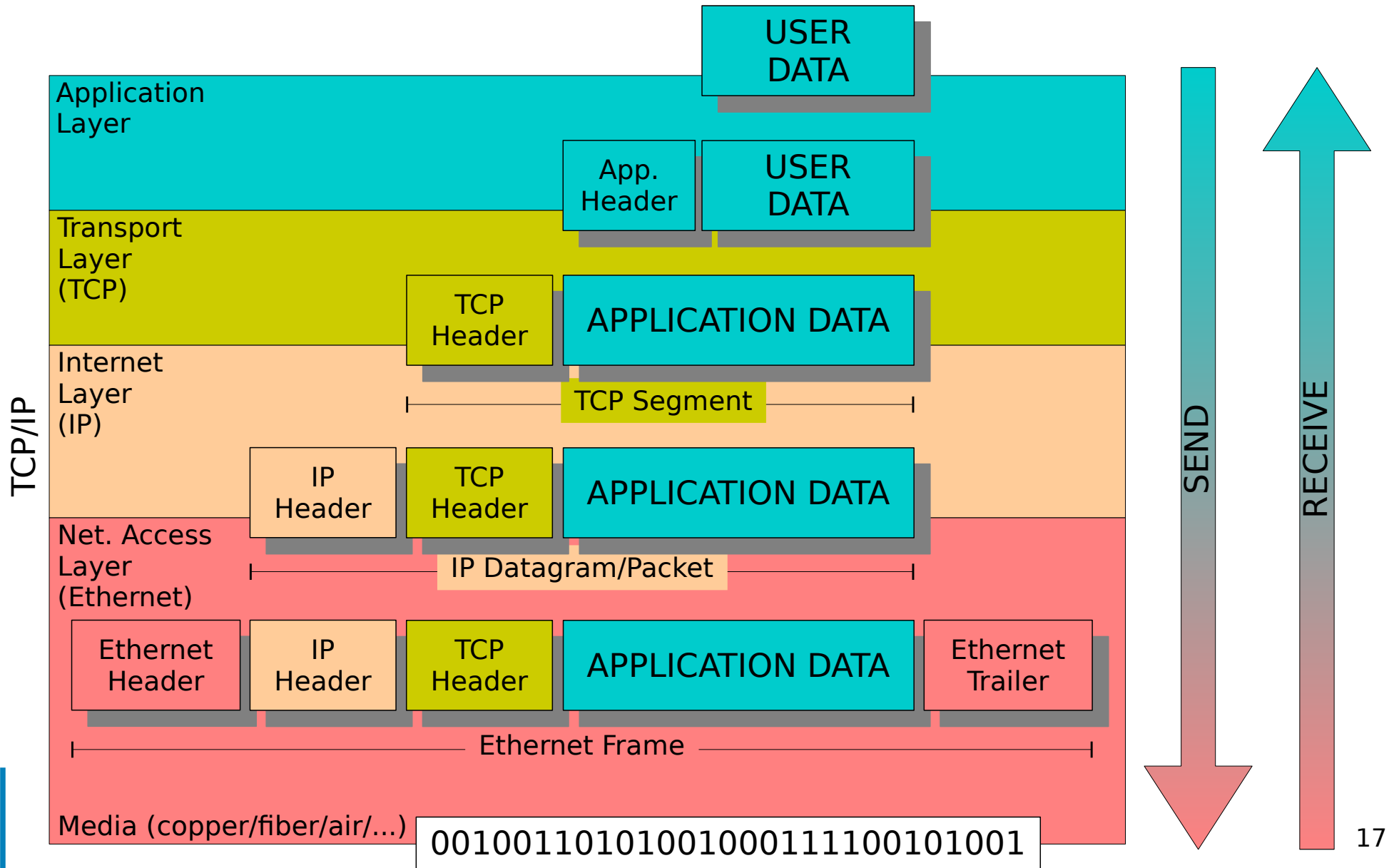
OSI and letter communication parallel

Adapted from: [http://en.wikipedia.org/wiki/File:Rm-osi\\_parallel.png](http://en.wikipedia.org/wiki/File:Rm-osi_parallel.png) by Josef Šábl



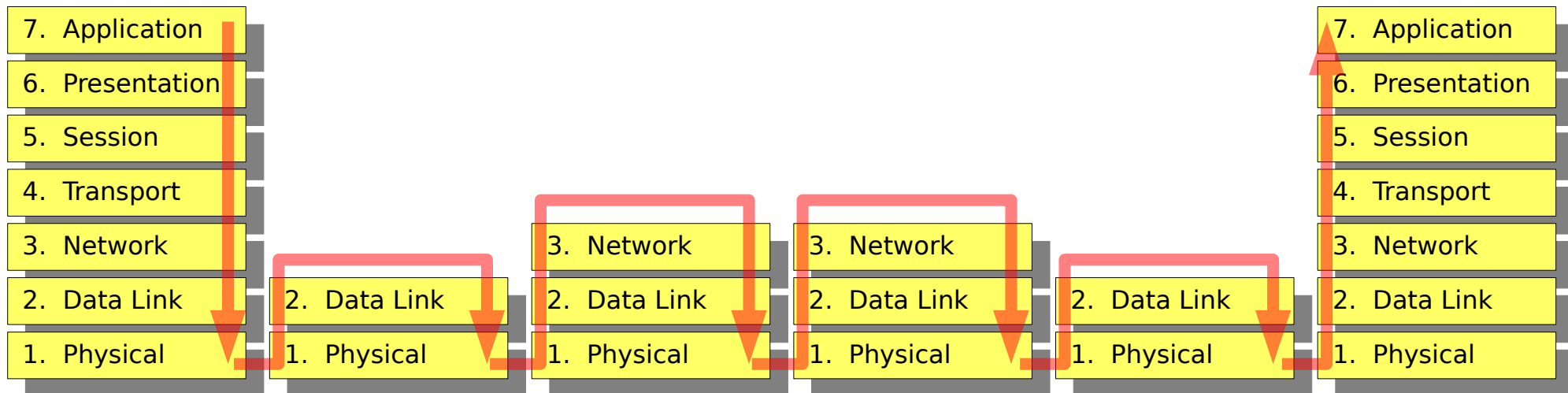
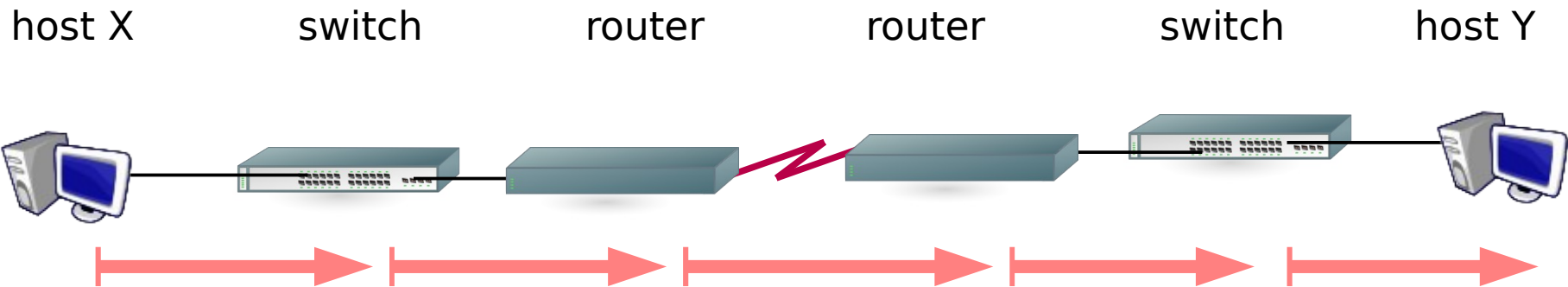


# Encapsulation/De-encapsulation





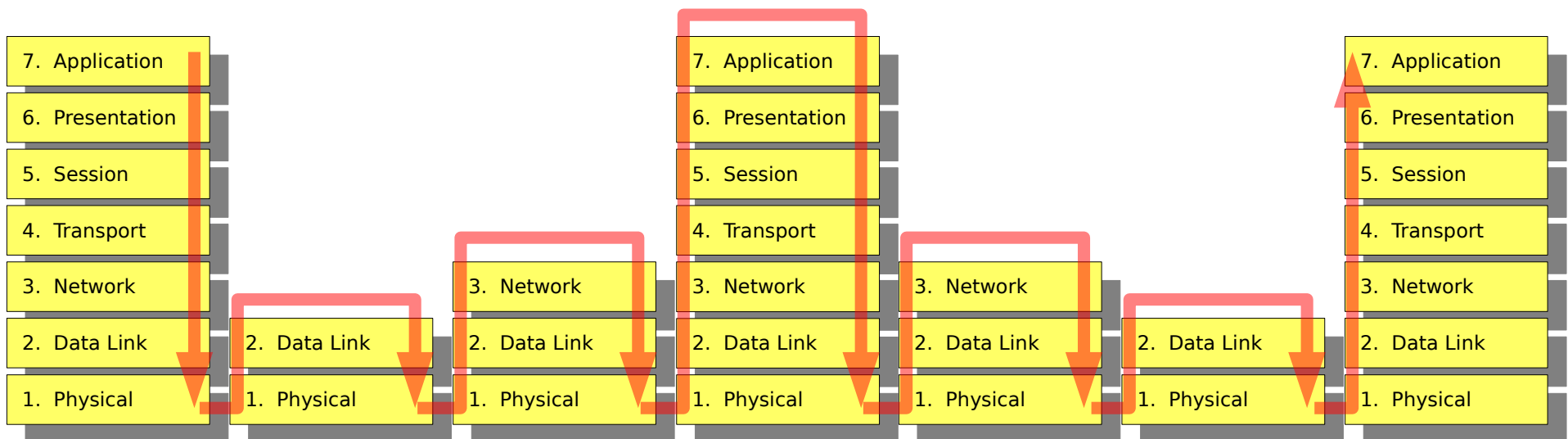
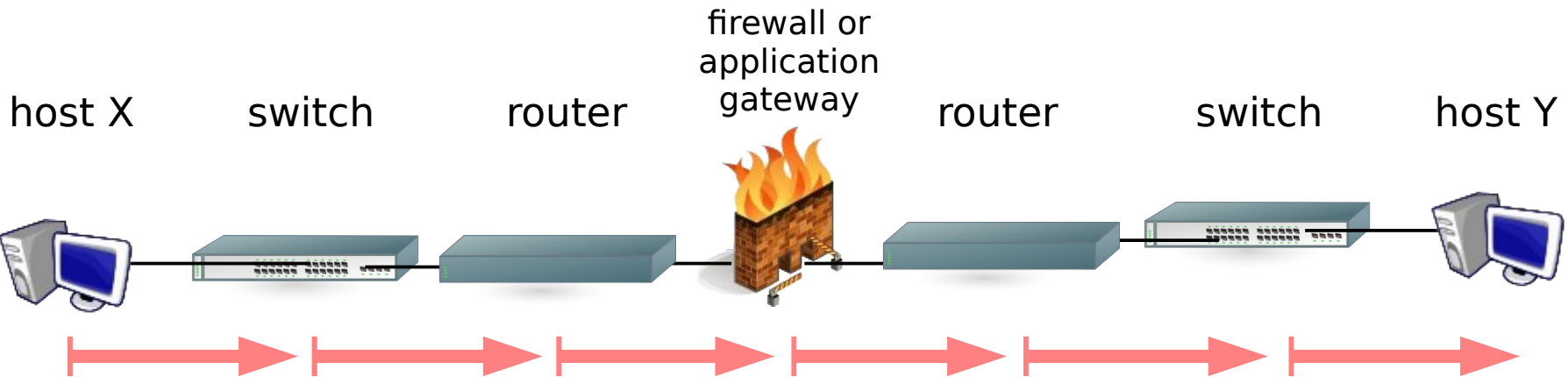
# Data flow



- Switches inspect the traffic for layer 2 info (MAC)
- Routers inspect the traffic for layer 3 info (IP)



# Data flow

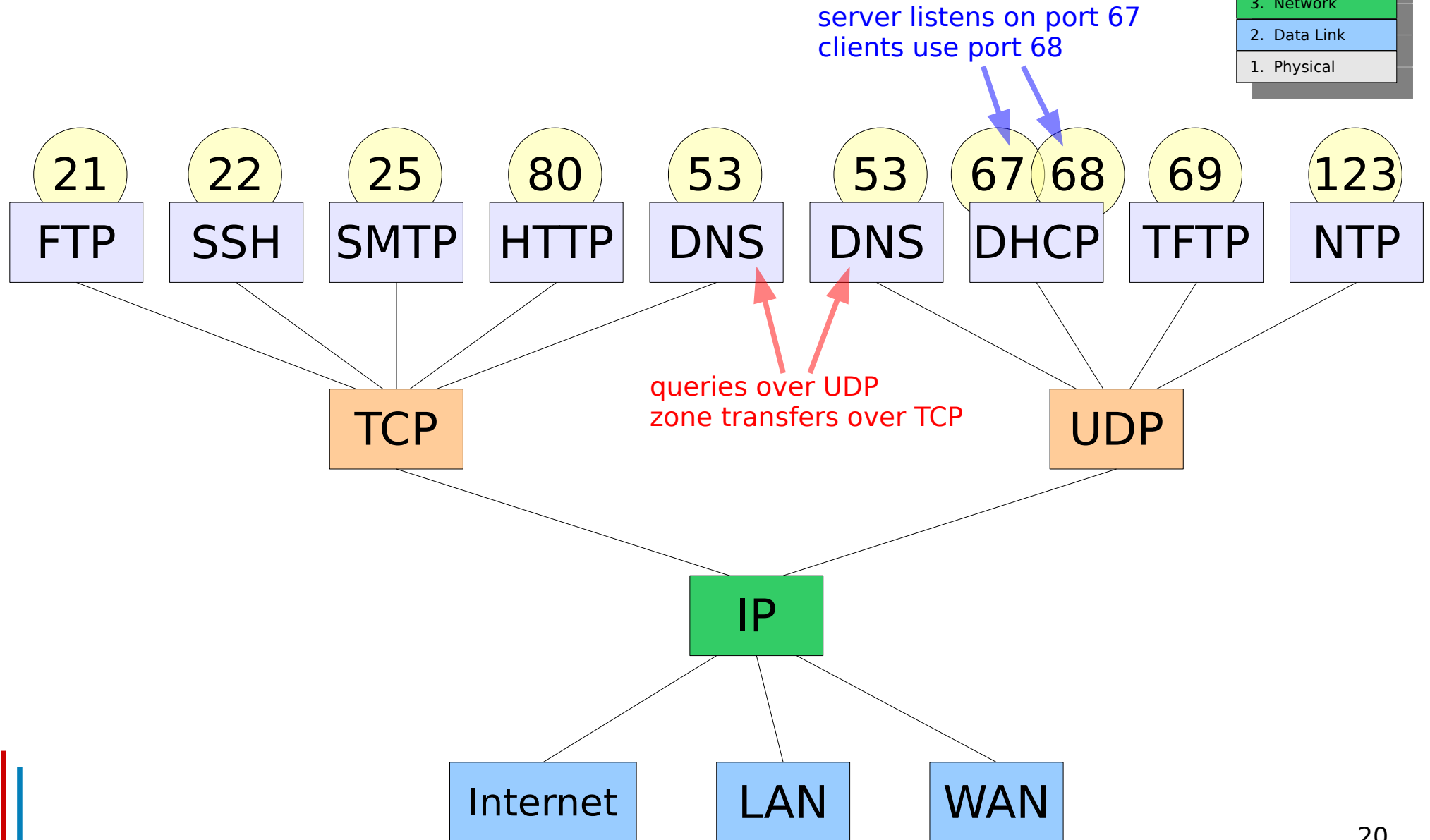


- Switches inspect the traffic for layer 2 info (MAC)
- Routers inspect the traffic for layer 3 info (IP)
- most Firewalls inspect the traffic for layers 2, 3 and 4 info
- Application Gateways (proxies) and layer-7 firewalls inspect the traffic up to layer 7



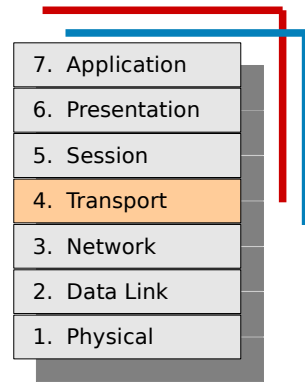
# Protocols, Ports and Services

7. Application
6. Presentation
5. Session
4. Transport
3. Network
2. Data Link
1. Physical





# Ports

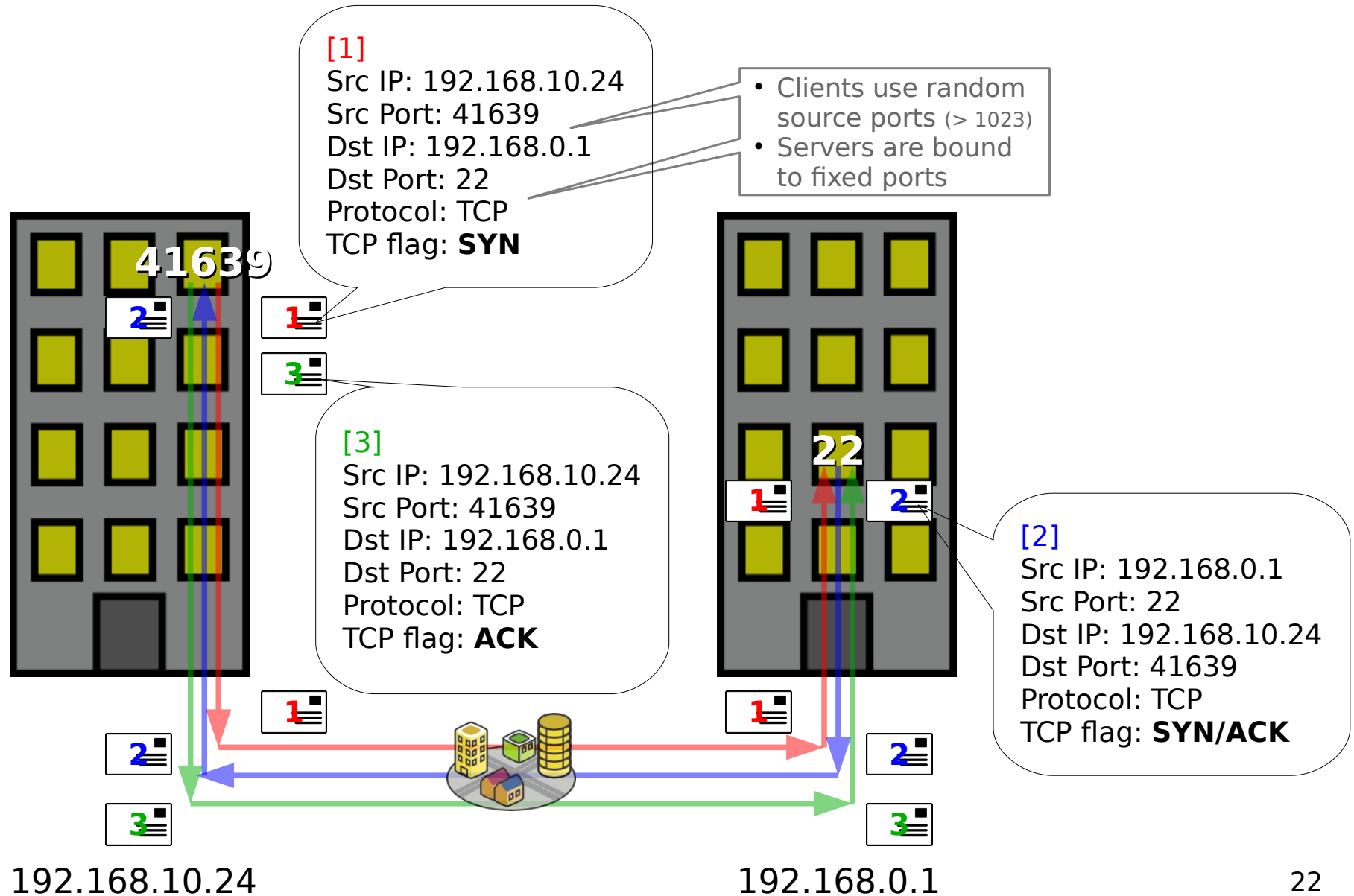


- Privileged Ports: 1-1023
  - main network services (SSH, SMTP, FTP, TFTP, DHCP, HTTP, HTTPS, ...)
  - need superuser's privileges
- Unprivileged Ports: 1024-65535
  - clients and unprivileged/no-suid services (Squid, NFS, X11, MySQL, ...)
  - any user can bind to any unprivileged port



# Opening a connection

## TCP 3-way Handshake





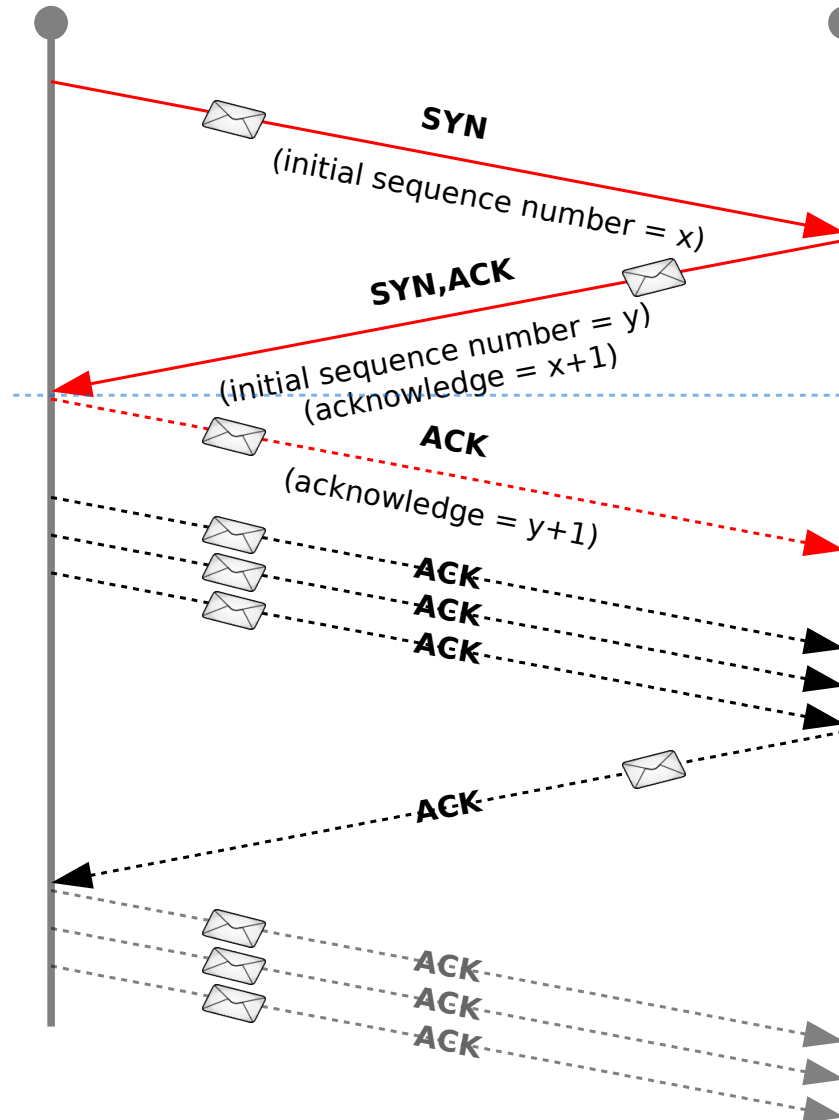
# Opening a connection

## TCP 3-way Handshake (TCP flags)



Host A

Host B



After this point, the SYN flag won't be ever set again for this connection, while the ACK will be always set.



# Opening a connection

## Protocols and (some) operations involved

### Sending

- layer 6/7: ssh someuser@host.somedomain.com
- layer 5: ssh/openssl session to host.somedomain.com, port 22, send **data**
- DNS: IP for host.somedomain.com
- layer 4: split the data into TCP **segments** and open a connection
- layer 3: obtain the route to the IP address for host.somedomain.com and send the **packets** to the next-hop
- layer 3/2: do we know the MAC address to next-hop? Check ARP table.
- layer 2: obtain MAC of next-hop, set frame header/trailer, send **frames**
- layer 1: check the link and transmit **bits** translated to a physical quantity (electric levels, light impulses, radio waves, ...)

### Receiving

- layer 1: check whether there's a transmission on the media
- layer 2: check whether the destination MAC belong to the NIC
- layer 3: check whether the destination IP address is ours
- layer 4: check whether at port 22, protocol TCP, there's a server listening
- layer 5: ssh/openssl negotiation and data transmission
- layer 6/7: check whether the connection is allowed from the source-host and that the user is hosted at this domain (authentication and spawn of a shell/command or deny connection)



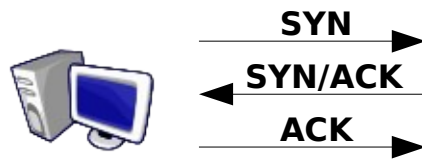


# TCP Connection

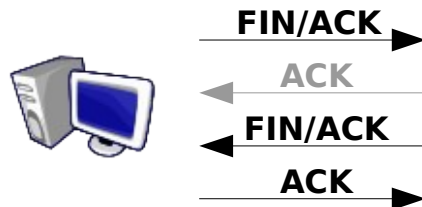
## TCP CONNECTION TOWARD AN OPEN PORT (listening)

CLIENT

SERVER



3-Way handshake  
- SYN/half-open scan  
- connect()

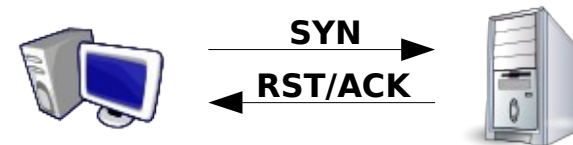


normal ending  
(if ESTABLISHED)

## TCP CONNECTION TOWARD A CLOSED PORT (non-listener)

CLIENT

SERVER





# UDP & ICMP

## UDP CONNECTION TOWARD AN OPEN PORT (listening)

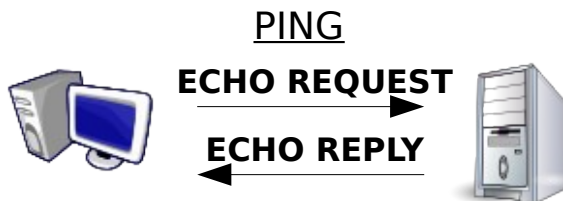
CLIENT SERVER



Depending on the application, there could be a reply or not

## ICMP REQUEST/REPLY

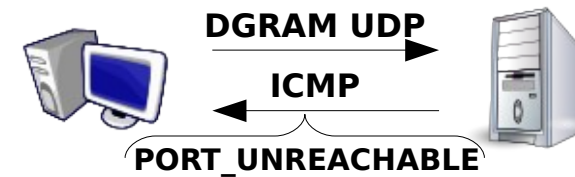
CLIENT SERVER



RFC 768  
RFC 792

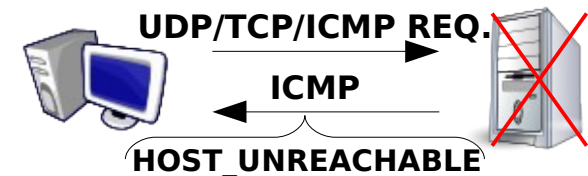
## UDP CONNECTION TOWARD A CLOSED PORT (non-listener)

CLIENT SERVER

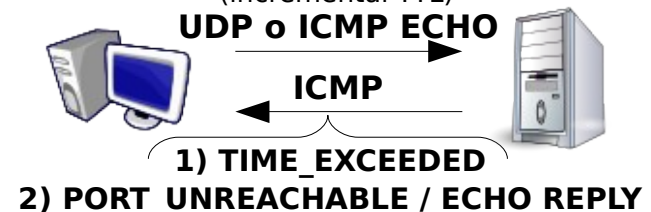


## ICMP NOTIFY

CLIENT SERVER



TRACEROUTE (\*)  
(incremental TTL)



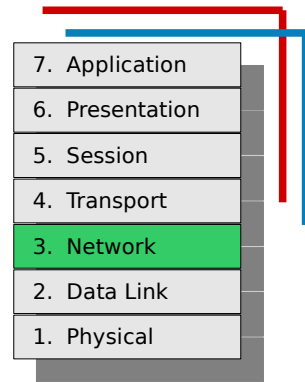
(\*) TCPTRACEOUTE does the same but uses TCP SYN, SYN/ECN/CWR or ACK and expects a SYN/ACK or RST, RST/ACK



# Internet Protocol and IP Address Space



# Internet Protocol



## The Internet Protocol (IP):

- provides network connectivity at **layer 3**
- is the main **routed protocol** of the Internet
- is the most widely used implementation of a **hierarchical network-addressing scheme**
- **addresses are used to route packets** from a source to a destination through the **best available path**
- is a **connectionless, unreliable, best-effort delivery protocol** (verification handled by upper protocols)



# IP(v4) addresses

The **IP address** is:

- a **numerical label** assigned to devices participating in a computing network, expected to **uniquely identify each host**
- obtained by the ISP (public IPs) or the system/network administrator (private IPs)
- **assigned** to a host **statically or dynamically** (BOOTP/DHCP)
- a 32 bits / 4 bytes unsigned integer number, usually **represented in a dotted-decimal notation**, as four 8bits/1byte numbers (0-255), called “octets”, separated by a dot '.' (0.0.0.0-255.255.255.255), sometimes in hexadecimal format (00000000-FFFFFFFF)
- divided in two parts, the *network* portion and the *host* portion
- something like this: 192.168.0.1 (C0A80001)



# Netmask, Network and Broadcast

The **netmask address** is:

- a **bit-mask of contiguous 1s** (starting from the MSB) that **separates the host portion from the network portion** of an IP address (1s on the network portion, 0s on the host portion)
- often represented in the “slash format” as the total number of bits used for the network and subnetwork portion of the mask (/8, /16, /24, /32, ...)
- something like this: 255.255.255.0 (FFFFFF00)

The **network address** is:

- what **identifies the network itself** and **defines the group of IP addresses that belongs to the same broadcast domain**, hosts that can communicate with each other without the need of a layer 3 device
- an IP address with the **host portion filled by 0s** (192.168.0.0, C0A80000)

The **broadcast address** is:

- a network address that **allows information to be sent to all nodes on a network**, rather than to a specific network host (unicast)
- an IP address with the **host portion filled by 1s** (192.168.0.255, C0A800FF)



# IP Address Notation

- *Dotted Quad Notation (four-octet dotted-decimal, numbers-and-dots)*
  - 10.240.27.73 / 255.255.255.0 (10.240.27.73/24)
- Hexadecimal Notation
  - 0AF01B49 / FFFFFFFF00
- Binary Notation
  - 00001010 11110000 00011011 01001001 /  
11111111 11111111 11111111 00000000

11111111 11111111 11111111	00000000	FFFFFF00	255.255.255.	0	Netmask
00001010 11110000 00011011	01001001	0AF01B49	10.240. 27.	73	IP Addr.
00001010 11110000 00011011	00000000	0AF01B00	10.240. 27.	0	Network Addr.
00001010 11110000 00011011	11111111	0AF01BFF	10.240. 27.	255	Broadcast Addr.

NETWORK PORTION

HOST PORTION



# IP Address Classes

Class A	Network	Host		
Octet	1	2	3	4

Class B	Network		Host	
Octet	1	2	3	4

Class C	Network			Host
Octet	1	2	3	4

Class D	Host			
Octet	1	2	3	4

Class	Network	Netmask	Broadcast	Host
A	x.0.0.0	255.0.0.0 (/8)	x.255.255.255	x.*.*.*
B	x.x.0.0	255.255.0.0 (/16)	x.x.255.255	x.x.*.*
C	x.x.x.0	255.255.255.0 (/24)	x.x.x.255	x.x.x.*

Class D addresses are used for *multicast* groups. There is no need to allocate octets or bits to separate network and host addresses.





# Identifying Address Classes

IP Address Class	High Order Bits	First Octet Address Range	Number of Bits in the Network Address	Number of Bits in the Host Address
Class A	0xxx	0 - 127	8	24
Class B	10xx	128 - 191	16	16
Class C	110x	192 - 223	24	8
Class D ( <i>multicast</i> )	1110	224 - 239	-	-
Class E ( <i>research</i> )	1111	240 - 255	-	-

Address Class	Number of (usable) Networks	Number of (usable) Hosts per Network
A	$2^{8-1} - 2 = 126$	$2^{24} - 2 = 16,777,214$
B	$2^{16-2} = 16,384$	$2^{16} - 2 = 65,534$
C	$2^{24-3} = 2,097,152$	$2^8 - 2 = 254$
D ( <i>multicast</i> )	N/A	N/A

The first address is reserved for the network ID, the last one is the network broadcast.

The 127.x.x.x address range is reserved as a loopback address, used for testing and diagnostic purposes, 0.x.x.x is reserved as "this"-network.



# Subnetting

- **prevented complete IP address exhaustion**
- **a method used to manage IP addresses dividing full network address classes into smaller pieces** (the network is not limited to the default Class A, B, or C network masks)
- subnetting a network means to **use the subnet mask to divide the network and break a large network up into smaller, more efficient and manageable segments, or subnets**, increasing the flexibility in the network design and providing also **broadcast containment and low-level security**
- **subnet addresses include the network portion plus a subnet field and a host field** (fields created from the original host portion of the major IP address by **re-assigning bits from the host portion to the network portion**)
- the number of bits “borrowed” depends on the number of hosts required per subnet



# Subnetting

**Class A, /8**  $2^8$  networks  
 $2^{24}$  hosts per network

11111111	00000000	00000000	00000000
00001010	11110000	00011011	01001001
00001010	00000000	00000000	00000000
00001010	11111111	11111111	11111111

255. 0. 0. 0 Netmask

10.240. 27. 73 IP Addr.

10. 0. 0. 0 Network Addr.

10.255.255.255 Broadcast Addr.

**NETWORK  
PORTION**

**HOST PORTION**

12 bits “borrowed” from hosts to networks



11111111	11111111 1111	0000	00000000
00001010	11110000 0001	1011	01001001
00001010	11110000 0001	0000	00000000
00001010	11110000 0001	1111	11111111

**NETWORK  
PORTION**

**SUBNET  
FIELD**

**HOST  
FIELD**

**Classless, /20**  $2^{20}$  networks  
 $2^{12}$  hosts per network

255.255.240. 0 Netmask

10.240. 27. 73 IP Addr.

10.240. 16. 0 Network Addr.

10.240. 31.255 Broadcast Addr.

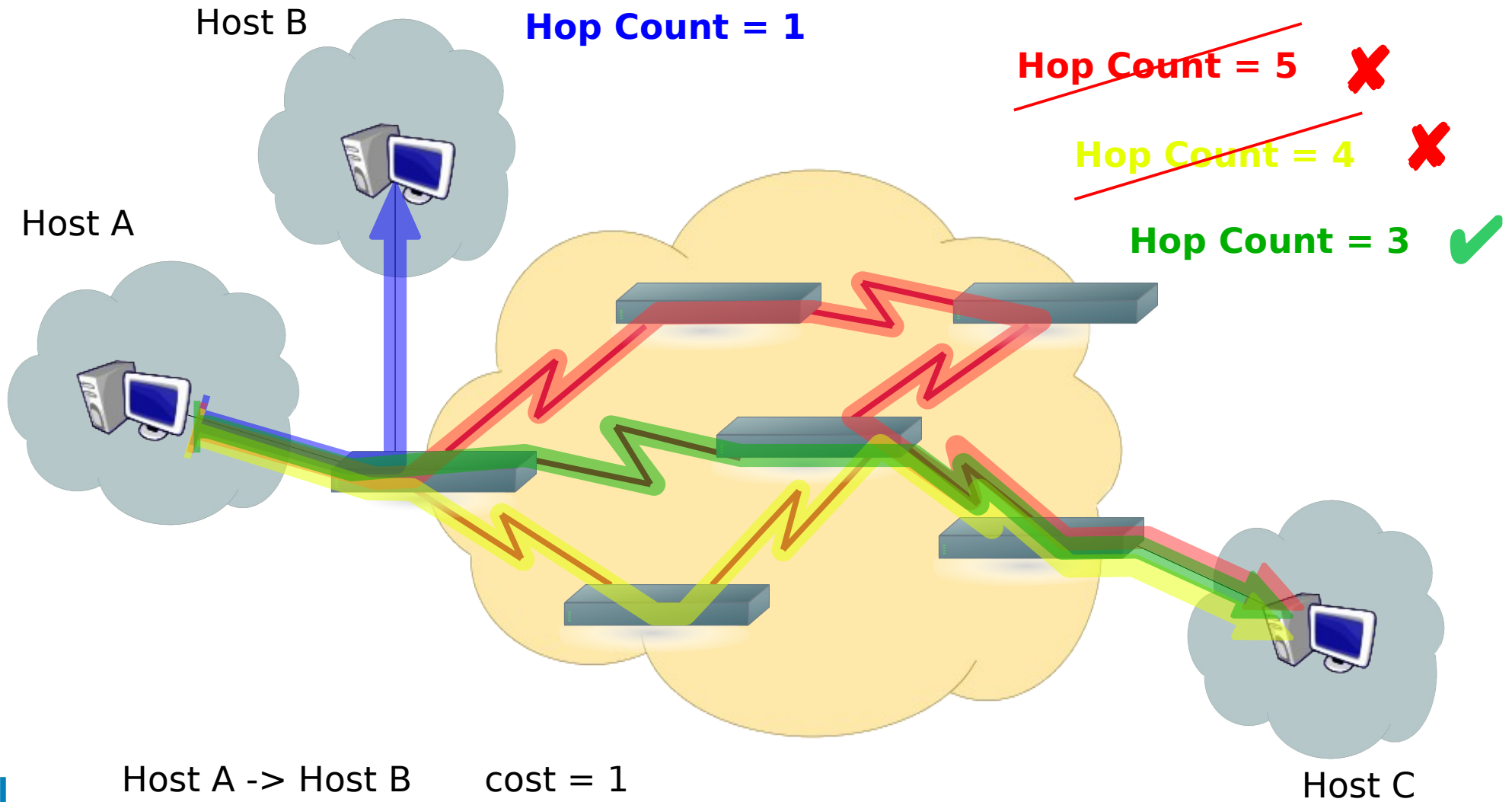


# Routing

- **routers are layer 3 devices that use the IP address to move data packets between networks** (when packets arrive at an interface, the **router uses the routing table** to determine where to send them)
- **each router that the packet encounters along the way is called a hop, the hop count is the distance traveled**
- **routing protocols use various combinations of metrics** (values that are used to determine the advantage of one route over another) **to determine the best path** along which network traffic should be forwarded (**hop count, load, bandwidth, delay, cost, and reliability of a network link**).
- **if a packet does not match any entries in the routing table, the router checks to see if a default route has been set.** If a default route has been set, the packet is forwarded to the associated port. **A default route is a route that is configured by the network administrator as the route to use if there are no matches in the routing table. If there is no default route, the packet is discarded.** A message is often sent back to the device that sent the data to indicate that the destination was unreachable.



# Best path determination





# Reserved IP Addresses

RFC 3330  
RFC 1918  
RFC 2606

- “This” network: 0.0.0.0/8
- Loopback: 127.0.0.0/8
- Private addresses:
  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16

10.0.0.0	172.16.0.0	192.168.0.0
10.255.255.255	172.31.255.255	192.168.255.255
- “TEST-NET” (example.com, org, net): 192.0.2.0/24
- 6to4 Relay: 192.88.99.0/24
- “Link local” (zeroconf): 169.254.0.0/16
- Multicast: 224.0.0.0/4



## IPv6 (in a nutshell)

- more extendible and scalable version of IP, **uses 128 bits** rather than the 32 bits
- **address assignment** can be **static** (manual configuration), **stateful** (DHCPv6) or **stateless auto-configuration** (SLAAC)
- **uses hexadecimal numbers** to represent the 128 bits, **8 groups of 4 hexadecimal digits** (16 bits) separated by ':'
  - 2001:0db8:0000:0000:00a9:0000:0000:0001
- initial 0s of each group can be removed
  - 2001:db8:0:0:a9:0:0:1
- a sequence of contiguous groups=0 can be replaced by a "::" (only once per address)
  - 2001:db8::a9:0:0:1 or 2001:db8:0:0:a9::1, NOT both (2001:db8::a9::1)
- netmask uses slash format "/N" (/64 is the default prefix)
  - 2001:db8::/32 (32 bits prefix)
    - addresses from 2001:db8:: to 2001:db8:ffff:ffff:ffff:ffff:ffff:ffff
  - 2001:db8::/64 (64 bits prefix):
    - addresses from 2001:db8:: to 2001:db8:0:0:ffff:ffff:ffff:ffff
- **::1/128 is the loopback address** ("ping6 ::1" to test IPv6 stack)



# Host names, Domain names and DNS

- **hostname**
  - **cerbero**.hpc.sissa.it
- **first level domain**
  - cerbero.hpc.sissa.**it**
- **second level domain**
  - cerbero.hpc.**sis**sa.it
- **third level domain**
  - cerbero.**hpc**.sis
- **Fully Qualified Domain Name (FQDN)**
  - **cerbero.hpc.sissa.it**
- **DNS**
  - cerbero.hpc.sissa.it      --> 147.122.17.62
  - 147.122.17.62          --> cerbero.hpc.sissa.it







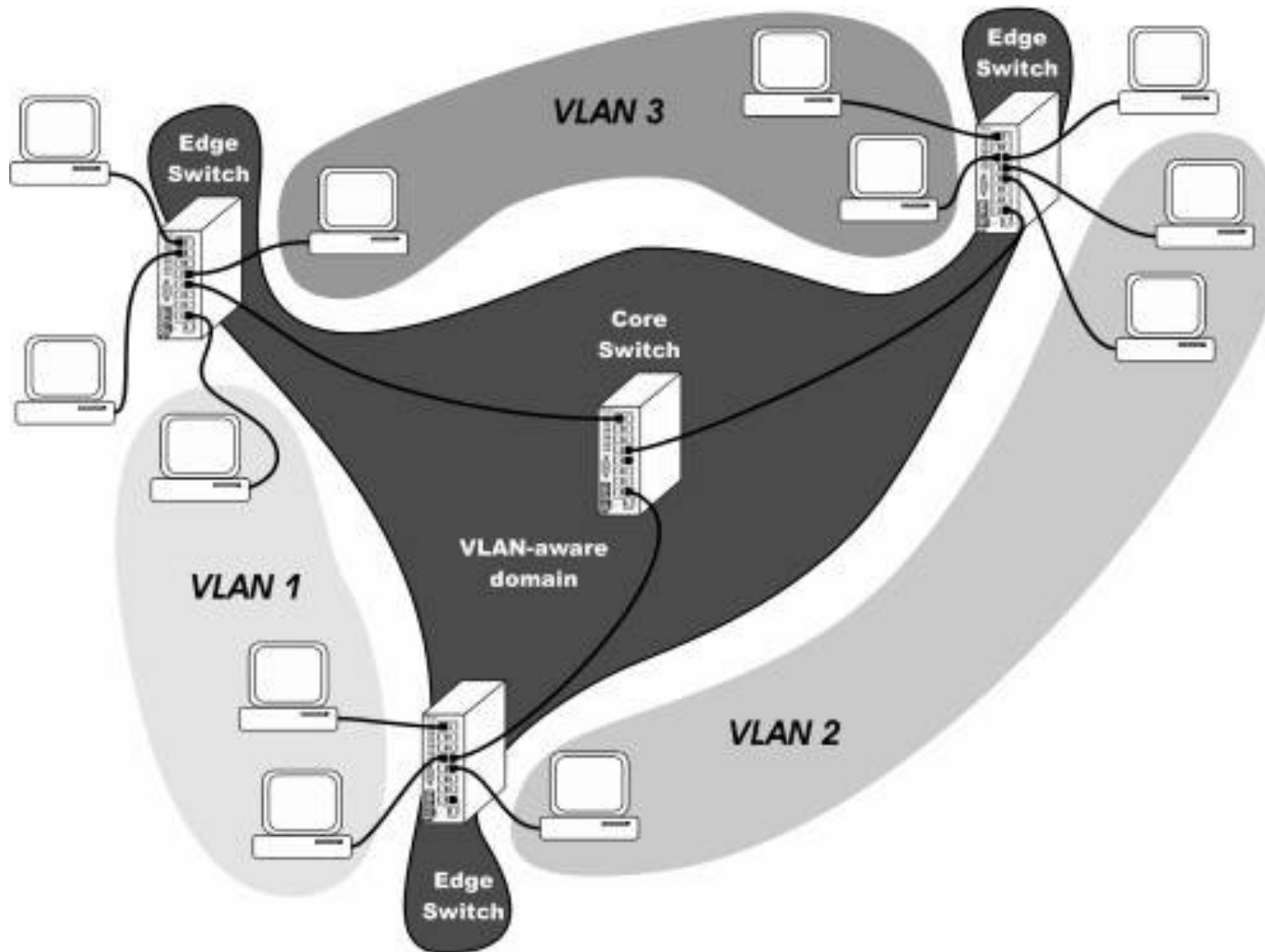
# Static vs. Dynamic IP assignment

- **static:** manual configuration (servers, network devices, workstations)
- **dynamic:** the DHCP server assigns an IP address to each DHCP client, associating the MAC address to an IP.  
The IP address can be:
  - **randomly assigned from a pool of IPs** (laptops on a wireless network or a LAN)
  - **sticky**, as above but the lease time is set to long periods (ISP)
  - **fixed** (workstations, network devices, cluster nodes, any device that must be always reachable at the same address), **requires individual profile for each device** (maps MAC-IP, providing Network Settings and, optionally, hostnames)
- **autoconfiguration (*link-local*):** communication between hosts on a single link (LAN segment) or a point-to-point connection



# Virtual LAN, network segmentation and trunking

7. Application
6. Presentation
5. Session
4. Transport
3. Network
2. Data Link
1. Physical



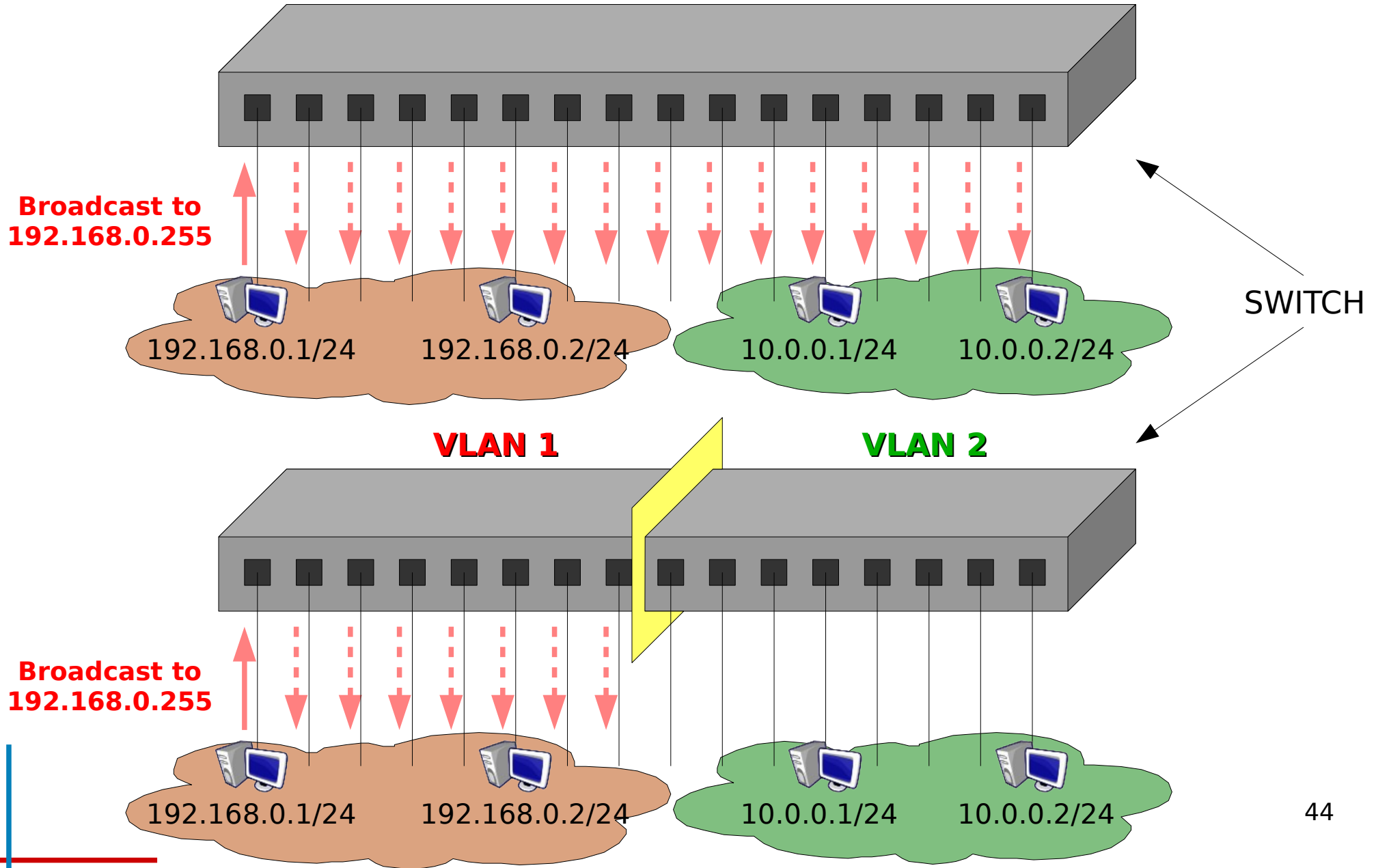


# Virtual LANs (VLAN)

- **logical group of hosts** with a common set of requirements that communicate as if they were attached to the Broadcast domain, **regardless of their physical location** (independent of physical topologies and distances – network location of users is no longer tightly coupled to their physical location)
- **provide segmentation services** traditionally provided by routers in LAN configurations, **addressing issues such as scalability, security, and network management**
- **a VLAN has the same attributes as a physical LAN**, but it allows for end stations to be grouped together even if they are not located on the same network switch (**network reconfiguration can be done through software instead of physically relocating devices**)
- **multiple Layer 3 networks on the same Layer 2 switch**
- in an environment employing VLANs, a one-to-one relationship often exists between VLANs and IP subnets (**Virtual LANs are Layer 2 constructs while IP subnets are Layer 3 constructs**)
- switches may not bridge IP traffic between VLANs as it would violate the integrity of the VLAN broadcast domain (routers in VLAN topologies provide broadcast filtering, security, address summarization, and traffic flow management)



# VLAN - Segmentation





# Port Trunking, Link Aggregation, NIC Teaming, Ethernet Channel Bonding, Etherchannel?

Different names for similar technologies.

Same purpose: provide fault tolerance and/or greater bandwidth.

**Link Aggregation:** general term that describes various methods of combining multiple network connections

**LACP (Link Aggregation Control Protocol):** IEEE 802.3ad, independent standard (became 802.1ax in 2008)

**Ethernet Channel Bonding:** LINUX main and historical software implementation (kernel-space)

**Linux Team Driver (libteam):** new LINUX project implemented in user-space (*teamd* daemon)

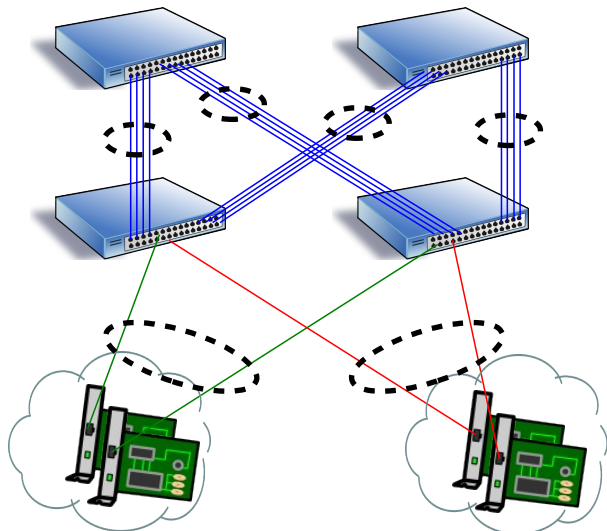
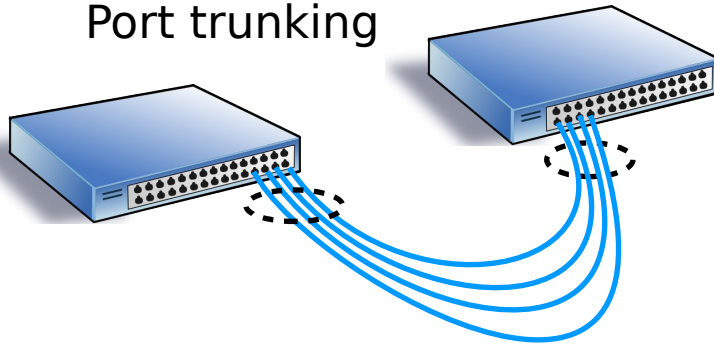
**Port Trunking:** (general term, switch configuration) method that combine more ports into a single virtual channel. Various protocols may define the (auto)configuration of the channel.

**EtherChannel:** as above, for Cisco technologies

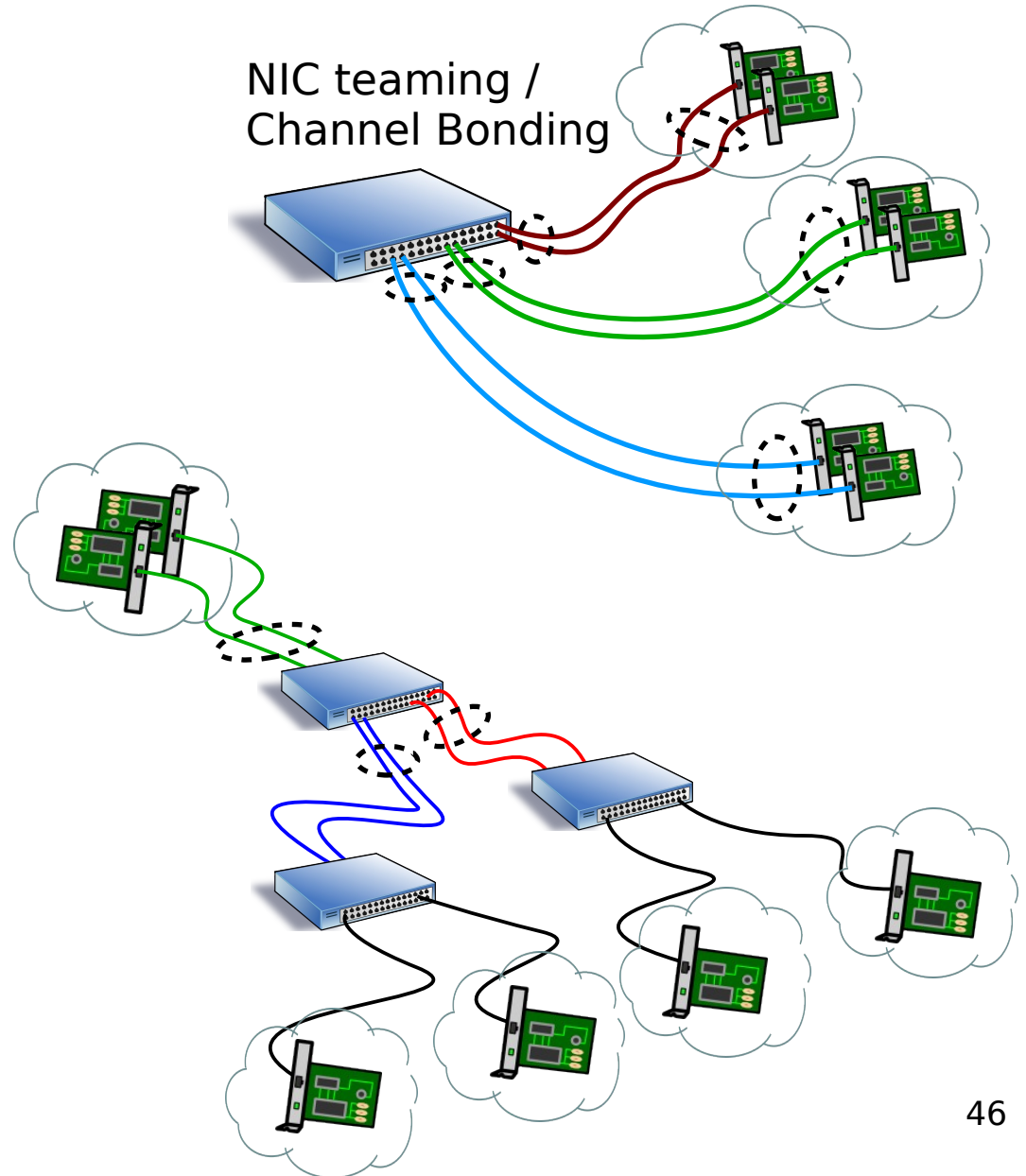


# Port trunking and NIC teaming

Port trunking

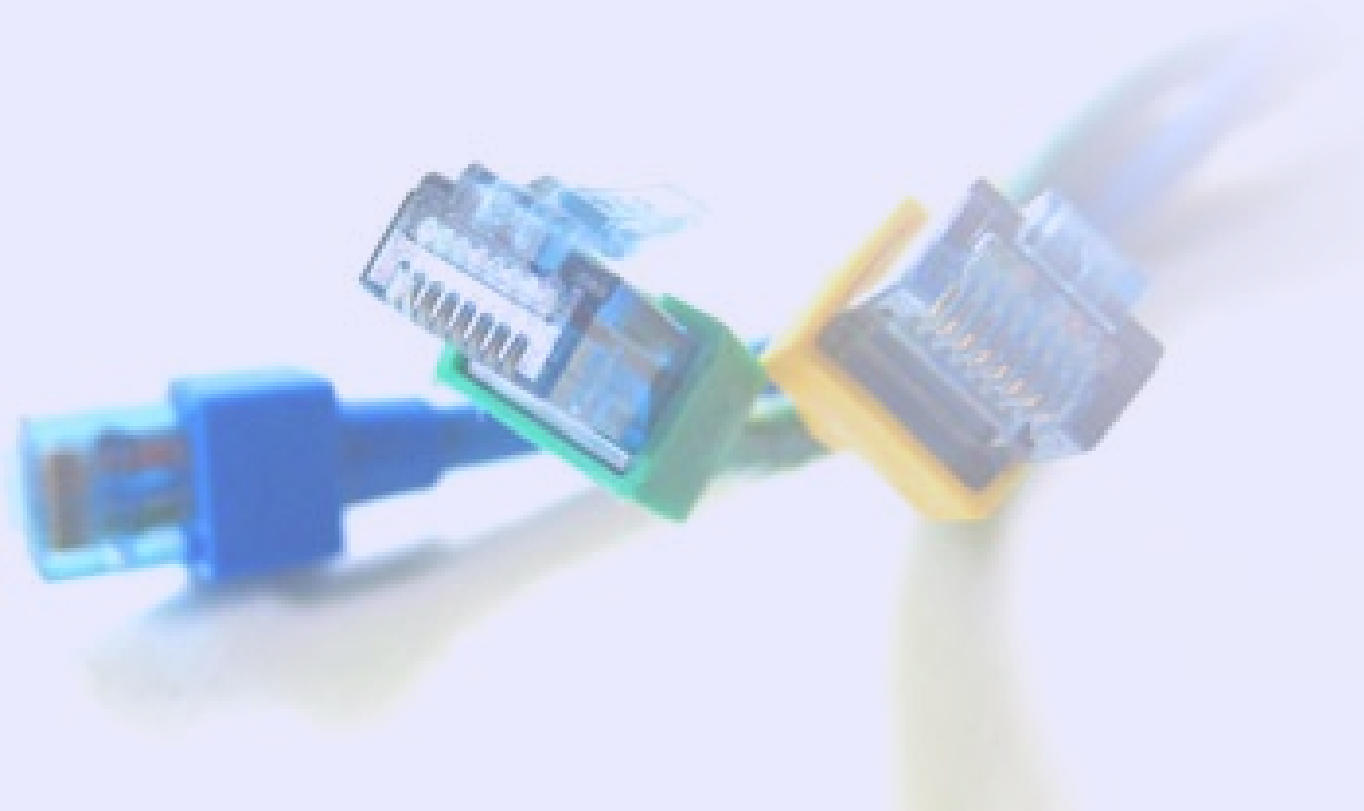


NIC teaming /  
Channel Bonding



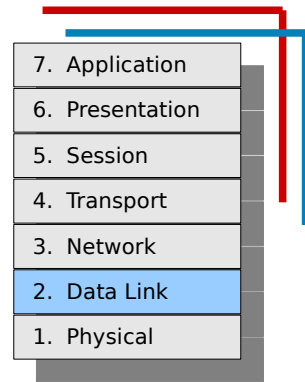


# Ethernet and Physical Address





# MAC Address



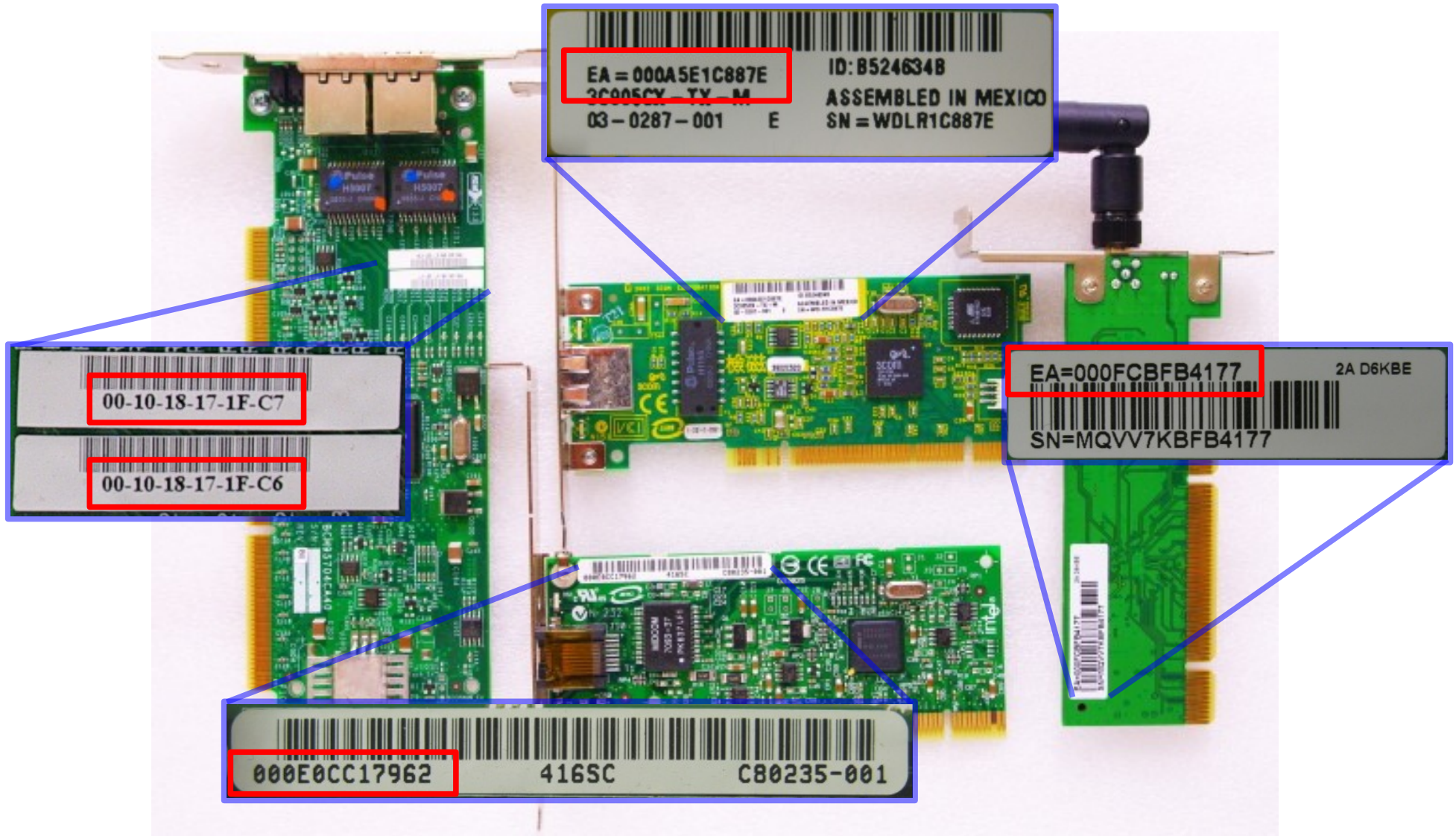
The **Media Access Control Address** is:

- a **physical address, globally unique**
- **assigned by the manufacturer** of the NIC and **burned-in into the PROM of the NIC** (in some cases, can be administratively assigned)
- part of the Ethernet protocol and **operates at Layer 2**
- sometimes called **Ethernet Hardware Address** (EHA)
- **used by DHCP to dynamically assign IP Addresses**
- a 48bits number represented as a 6 groups of two hexadecimal digits (6 bytes) separated by ':' (00:1d:09:d7:3b:25)
- made of two parts, 3 bytes each:
  - the OUI (Organizationally Unique Identifier)
  - the production number
- replaced in the frame header at each layer 3 device (router/gateway) during transmission





# MAC Address

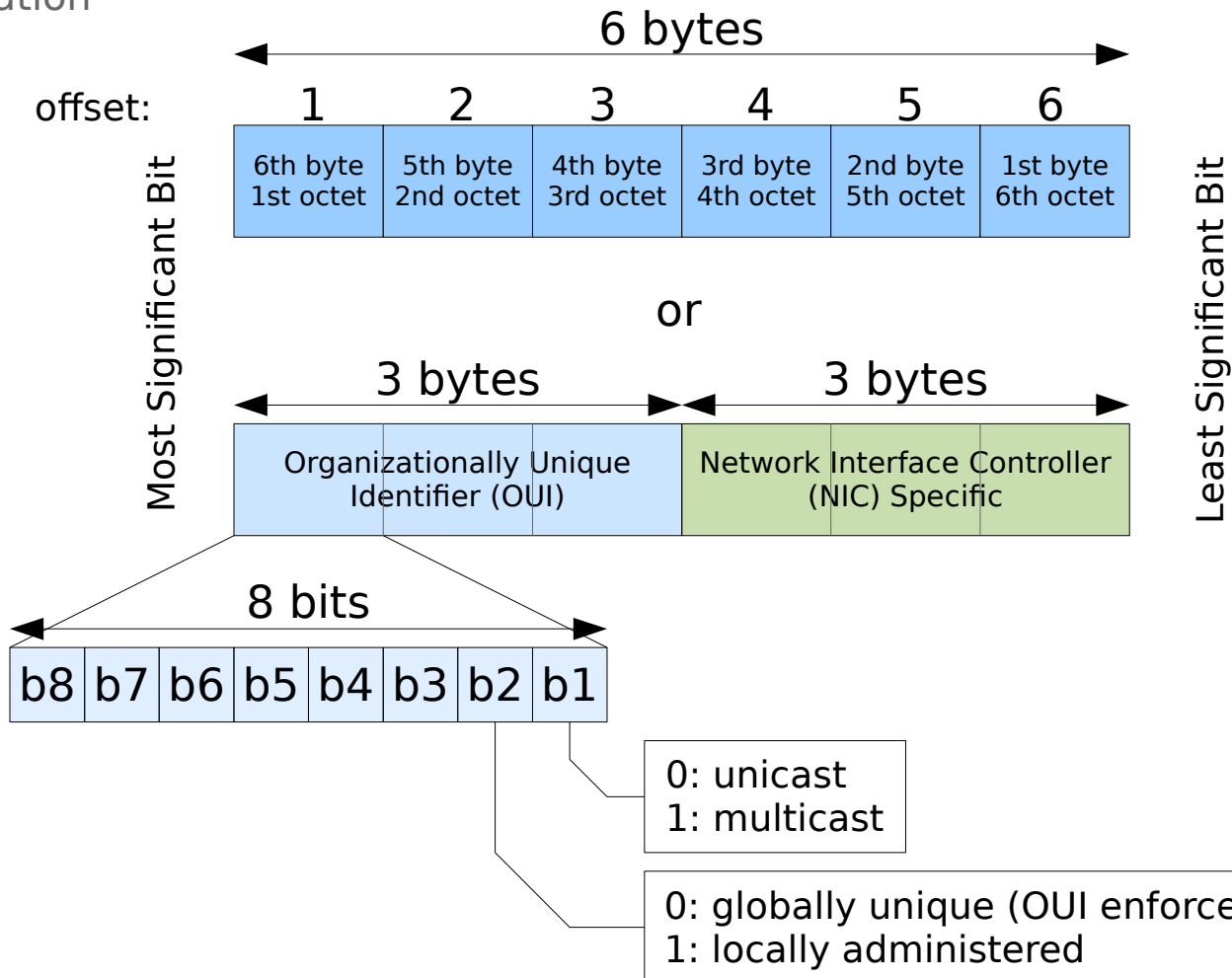




# MAC Address Notation

the OUI 00-0e-0c  
belongs to the  
Intel Corporation

00:0e:0c:d7:3b:25





# ARP: IP to MAC mapping

## ***Address Resolution Protocol:***

- layer 3 - layer 2 conversion, **maps IP addresses to physical hardware addresses (MAC)**
- used to **identify the next layer-2 hop/device** (NIC or switch) in order to reach the destination IP
  - ARP Who has 192.168.0.101? Tell 192.168.1.1
  - ARP 192.168.0.101 is at 00:04:76:9b:ec:46
- the OS keeps IP/MAC couples in the **ARP table** for a limited amount of time



# Cables and connectors

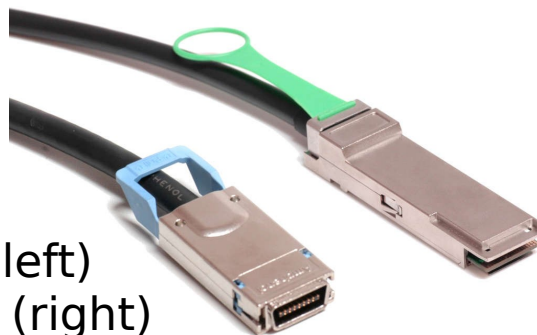
7. Application
6. Presentation
5. Session
4. Transport
3. Network
2. Data Link
1. Physical

- **bandwidth varies depending upon the type of media as well as the technologies used**, the physics of the media account for some of the difference
- signals travel through twisted-pair copper wire, coaxial cable, optical fiber, and air
- **the physical differences in the ways signals travel result in fundamental limitations on the information-carrying capacity of a given medium**
- **actual bandwidth of a network is determined by a combination of the physical media and the technologies chosen for signaling and detecting network signals.**

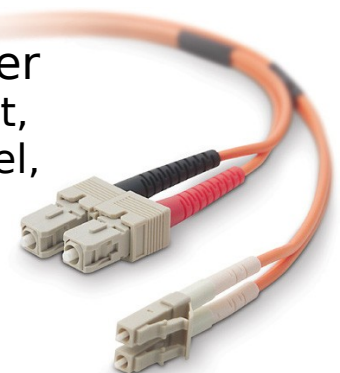
Ethernet RJ45  
(10/100/1000)



CX4 (left)  
QSFP (right)  
(Infiniband & 10GB Ethernet)

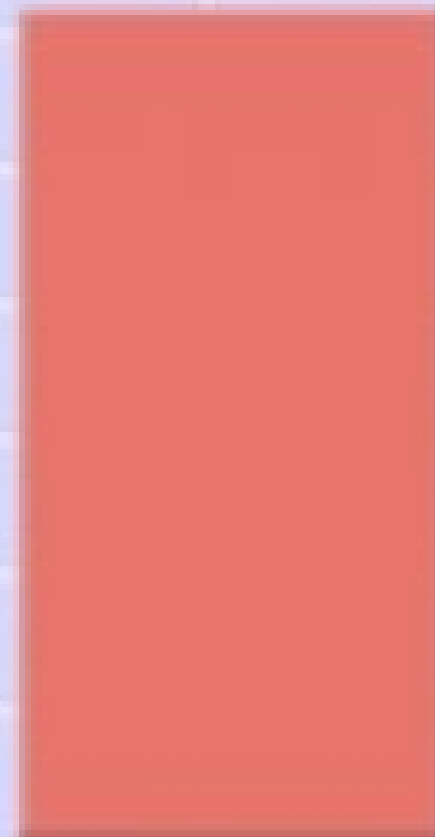
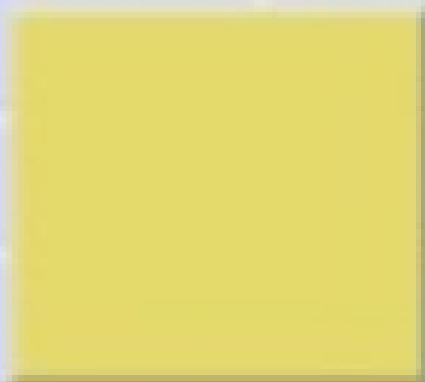


SC / LC Fiber  
(\*G Ethernet,  
Fiber Channel,  
Myrinet  
& more)





# About Latency, Bandwidth, Speed and Throughput





# Latency in Networking

**Latency** is the **delay** between the time a frame begins to leave the source device and when the first part of the frame reaches its destination. A variety of conditions can cause delays:

- Media delays may be caused by the **finite speed** that signals can travel through the physical media.
- Circuit delays may be caused by the **electronics that process the signal** along the path.
- Software delays may be caused by the **decisions that software must make** to implement switching and protocols.



# Latency in HPC

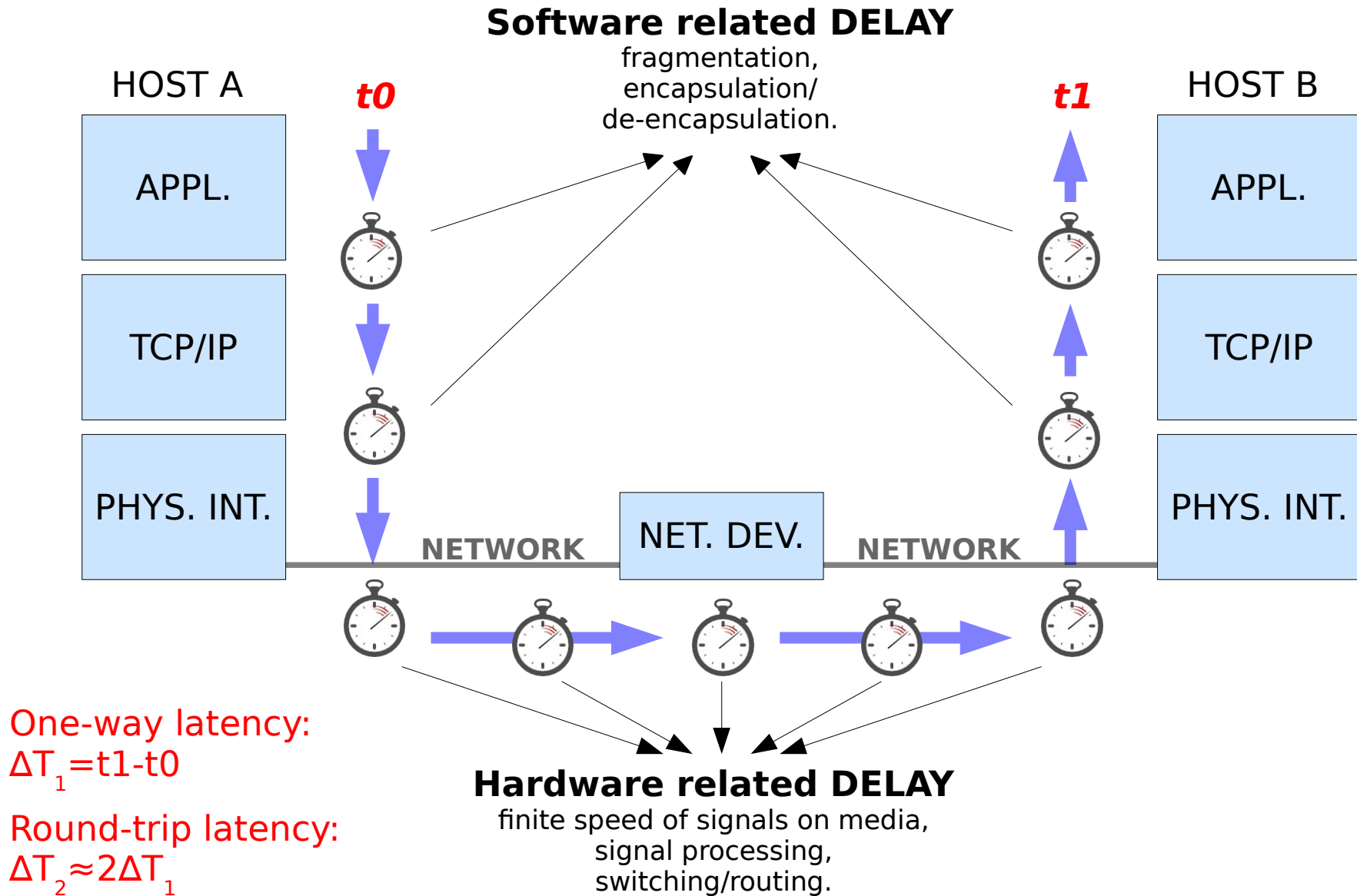
The **one-way latency** may be also meant as **the period of time that a 0-sized message spends traveling from its source to its destination**, which involves the time necessary to encode, send the packet, receive the packet, and decode it to be made available to the higher level software stack. The **round-trip latency** includes also the travel back to the source of an **acknowledge message**.

- Protocol delays (TCP/IP fragmentation, TCP windowing and acknowledgments, encapsulation and de-encapsulation)
- Network delays (media, circuitry and signal processing, switching/routing)
- Application delays (message creation, acknowledgment)





# Latency







# Bandwidth and Speed

**Bandwidth** is the measure of the **amount of information that can move through the network in a given period of time**. A typical LAN might be built to provide 100 Mbps to every desktop workstation, but this does not mean that each user is actually able to move 100 megabits of data through the network for every second of use. This would be true only under the most ideal circumstances.

**Speed** is often used interchangeably with bandwidth, but a large-bandwidth device will carry data at roughly the same speed of a small-bandwidth device if only a small amount of their data-carrying capacity is being used.



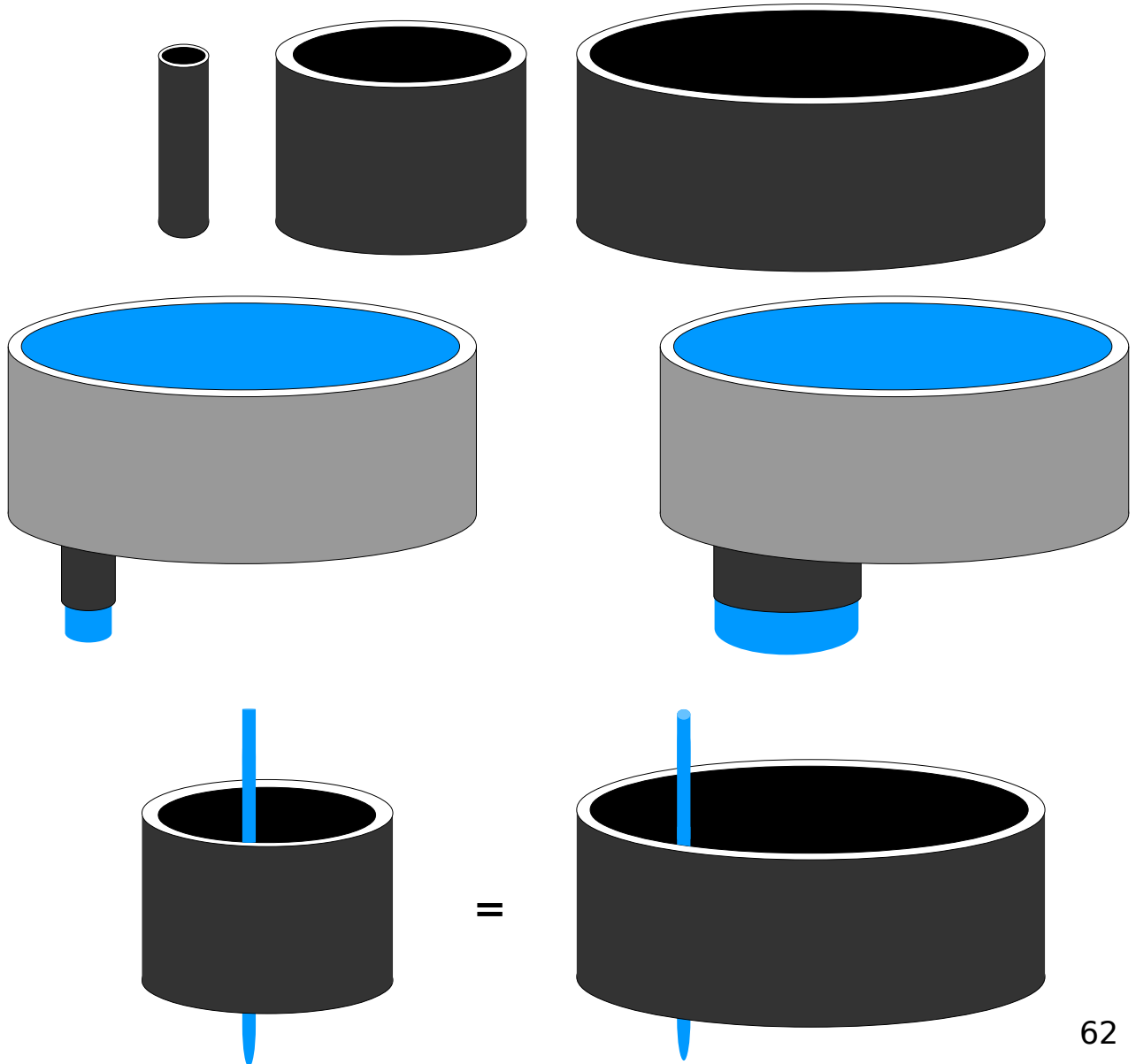
# Bandwidth and Speed

The larger the bandwidth  
the larger the amount of  
data that can pass through

**BUT**

for amount of data  
significantly smaller  
than the actual capacity

**BANDWIDTH  $\neq$  SPEED**





# Throughput

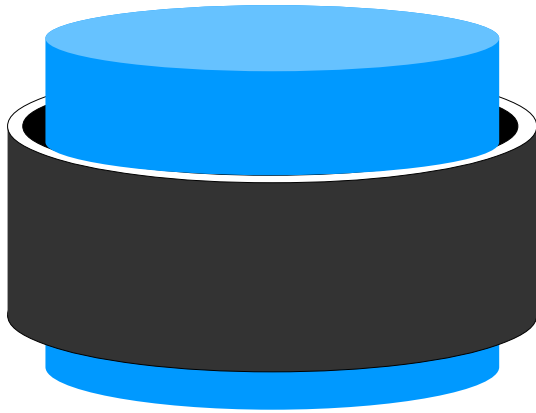
**Throughput** refers to **actual measured bandwidth**, at a specific time of day, using specific Internet routes, and while a specific set of data is transmitted on the network. Unfortunately, for many reasons, throughput **is often far less than the maximum possible digital bandwidth** of the medium that is being used. The following are some of the factors that determine throughput:

- **Internetworking devices**
- **Type of data being transferred**
- Network topology
- Number of users on the network
- User computer
- Server computer
- Power conditions

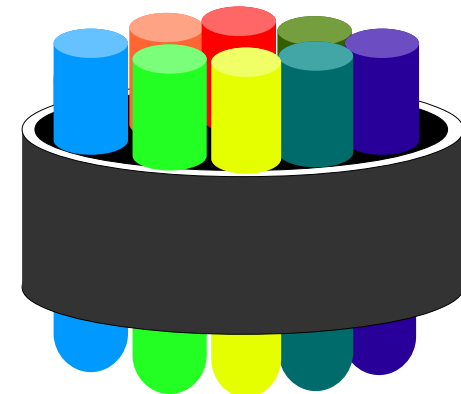
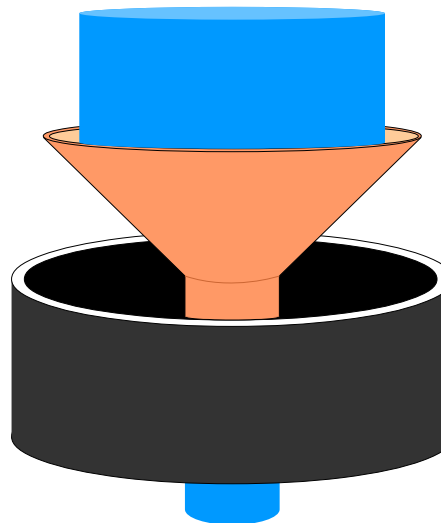
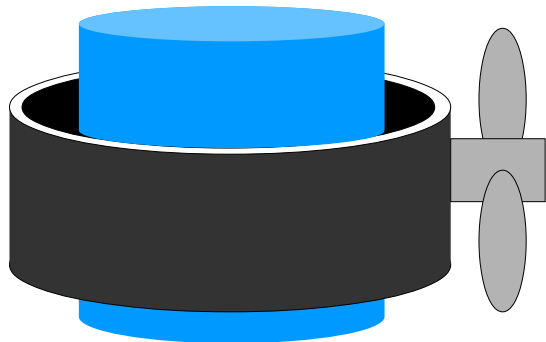
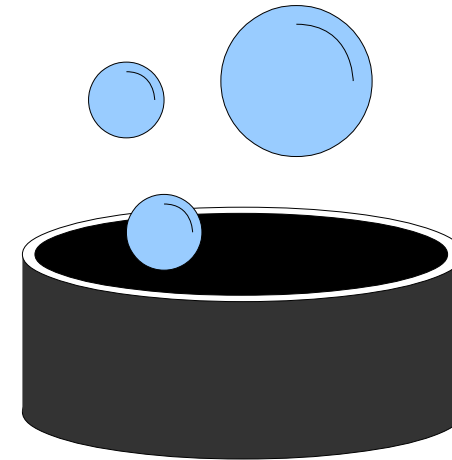


# Throughput

Best-case:  
Throughput = Bandwidth

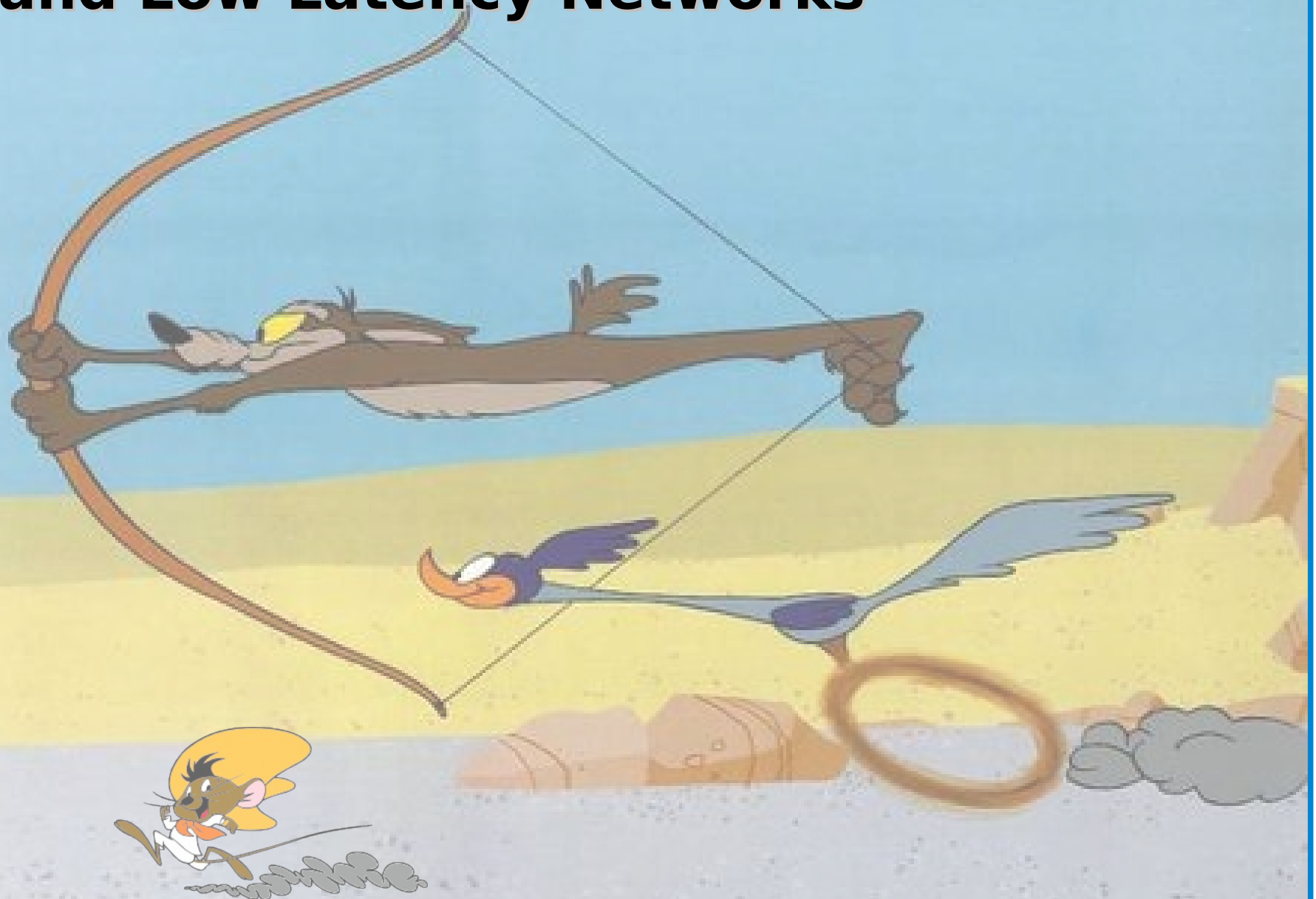


Real-case is often:  
Throughput  $\ll$  Bandwidth





# Network Performance Benchmarking and Low Latency Networks





# Network performance

If the typical file size for a given application is known, dividing the file size by the network bandwidth yields an estimate of the fastest time that the file can be transferred:

$$T = S / BW$$

Two important points should be considered when doing this calculation:

- **the result is an estimate only, because the file size does not include any overhead added by encapsulation**
- **the result is likely to be a best-case transfer time, because available bandwidth is almost never at the theoretical maximum for the network type** (a more accurate estimate can be attained if throughput is substituted for bandwidth in the equation)

Best download  
Typical Download

$$T = S / BW$$
$$T = S / P$$

BW = Maximum theoretical bandwidth of the "slowest link" between the source host and the destination host (measured in bits per second)

P = Actual throughput at the moment of transfer (measured in bits per second)

T = Time for file transfer to occur (measured in seconds)

S = File size in bits



# Benchmarking

- **specifications** often “lie”, or at least **report best-case transfer rates and theoretical bandwidth** or aspects of benchmarks that show their products in the best light (*bench-marketing*)
- **direct comparison of different architectures becomes important**, especially in HPC
- **benchmarks are designed to mimic a particular type of workload**
  - ***synthetic benchmarks*** use specially created programs that impose the workload on the component
  - ***application benchmarks*** run real-world programs on the system
  - whilst *application benchmarks* usually give a much better measure of real-world performance on a given system, *synthetic benchmarks* are useful for testing individual components, like a networking device
- some **open source network benchmarking tools** are **IPERF, NETPERF, NETPIPE**



# Why is (low) latency so important in HPC?

1/2

Bandwidth is (often) not the biggest technical challenge anymore. Network-induced latency/jitter/round trip delay (RTD) are becoming the biggest problem. In HPC, latency is a very important issue.

According to **Amdahl's law**:

- **a high-performance parallel system tends to be bottlenecked by its slowest sequential process**
- in all but the most embarrassingly parallel supercomputer workloads, **the slowest sequential “process” is often the message transmission due to the latency involved**
- **the impact of latency is larger for large amount of small messages**, but can be mitigated by increasing the bandwidth of the network (when possible), reducing the hardware-related latency (infiniband), the software-related delays (offload, RDMA, ...) and increasing the message size (grouping small messages in larger chunks at application level), thus increasing the efficiency of data transmission across the network





# Why is (low) latency so important in HPC?

2/2

The time taken to send a message of length  $n$  is roughly:

$$Time = latency + n * cost\_per\_byte$$

assuming that the topology is irrelevant it becomes simply:

$$Time = latency + n / bandwidth$$

Often called "a-b model" (alpha/beta model) and written:

$$Time = a + n * b$$

Since, usually,  $a \gg b \gg \text{time-per-flop}$ , sending one single large message is cheaper than sending many short ones:

$$1*(a + n*b) \ll n*(a + 1*b)$$

Sending a message can cost hundreds or thousands of flops.

**Lesson: Need large computation-to-communication ratio to be efficient.**

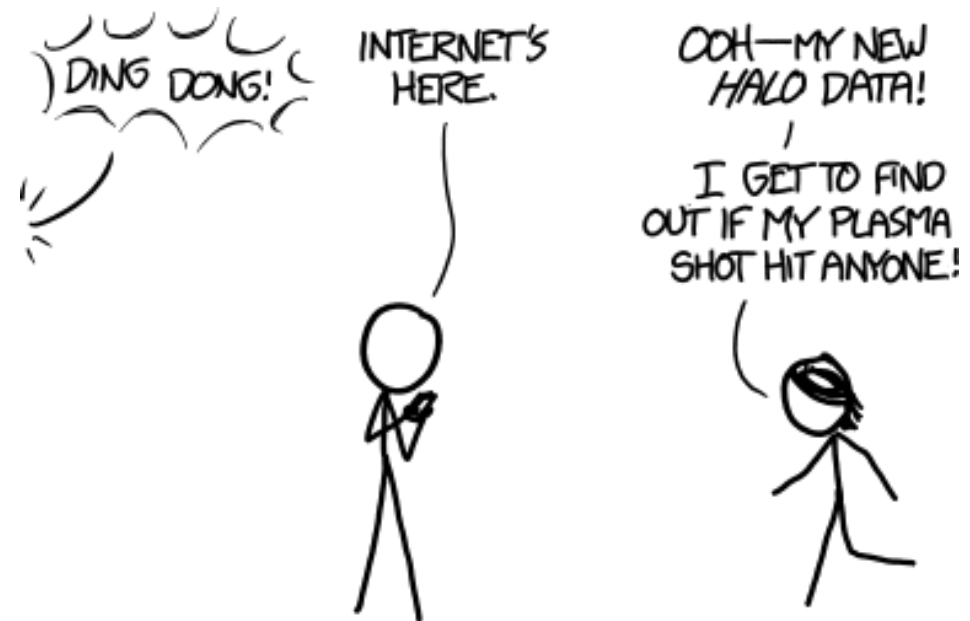


# Why is (low) latency so important in general?

*Never underestimate the bandwidth of a station wagon full of tapes hurtling down the highway.*  
- Andrew Tanenbaum, 1981

What about the latency of such medium? Is larger bandwidth and/or larger message size always good for all applications?

VoIP, interactive video conferencing, network gaming, interactive sessions, cloud computing, and especially automatic algorithmic trading, have strict requirements on latency, but are not necessarily concerned about bandwidth (large amount of small transactions).

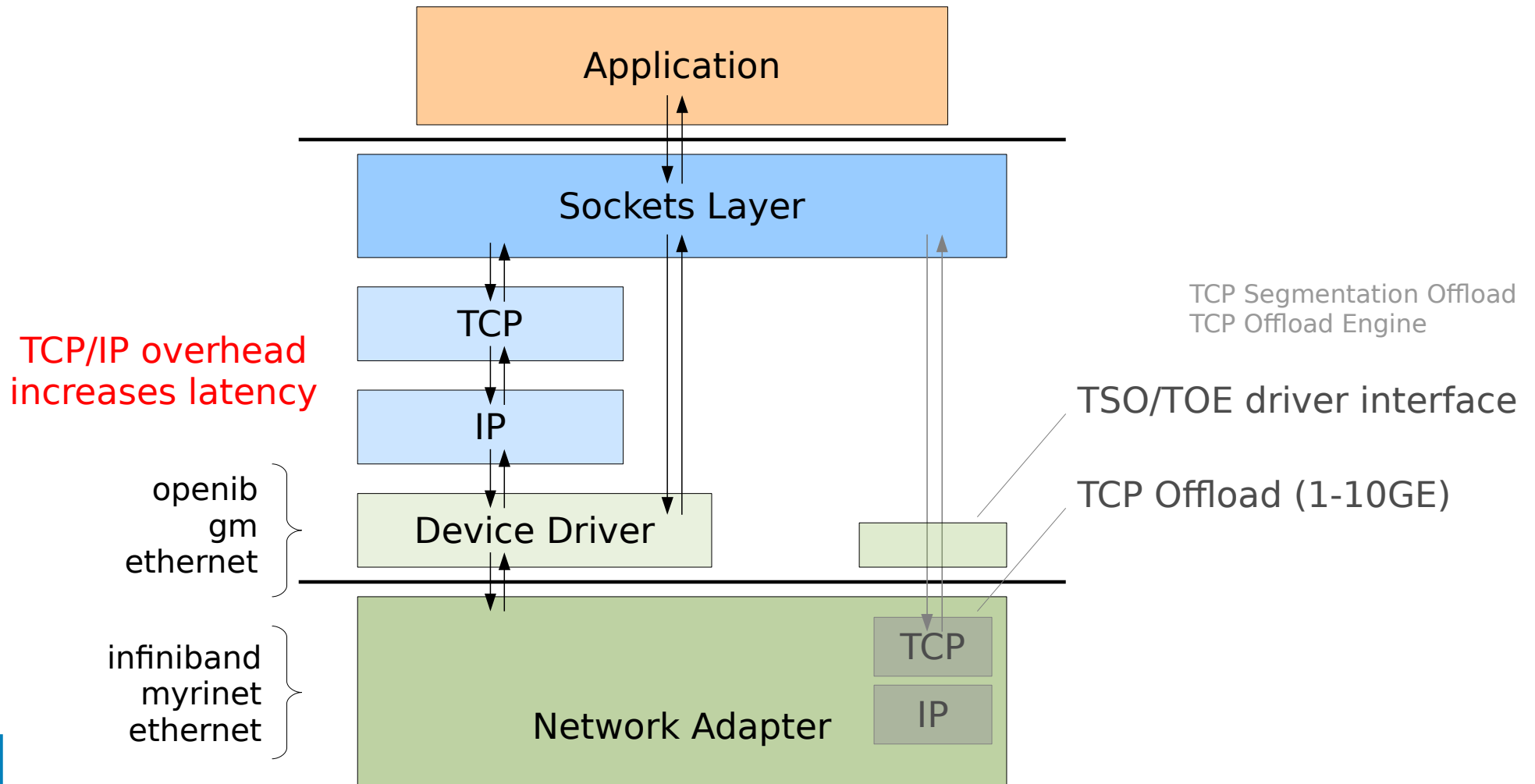


- <https://what-if.xkcd.com/31/>



# Low Latency Networks

## TCP/IP vs Native Protocols

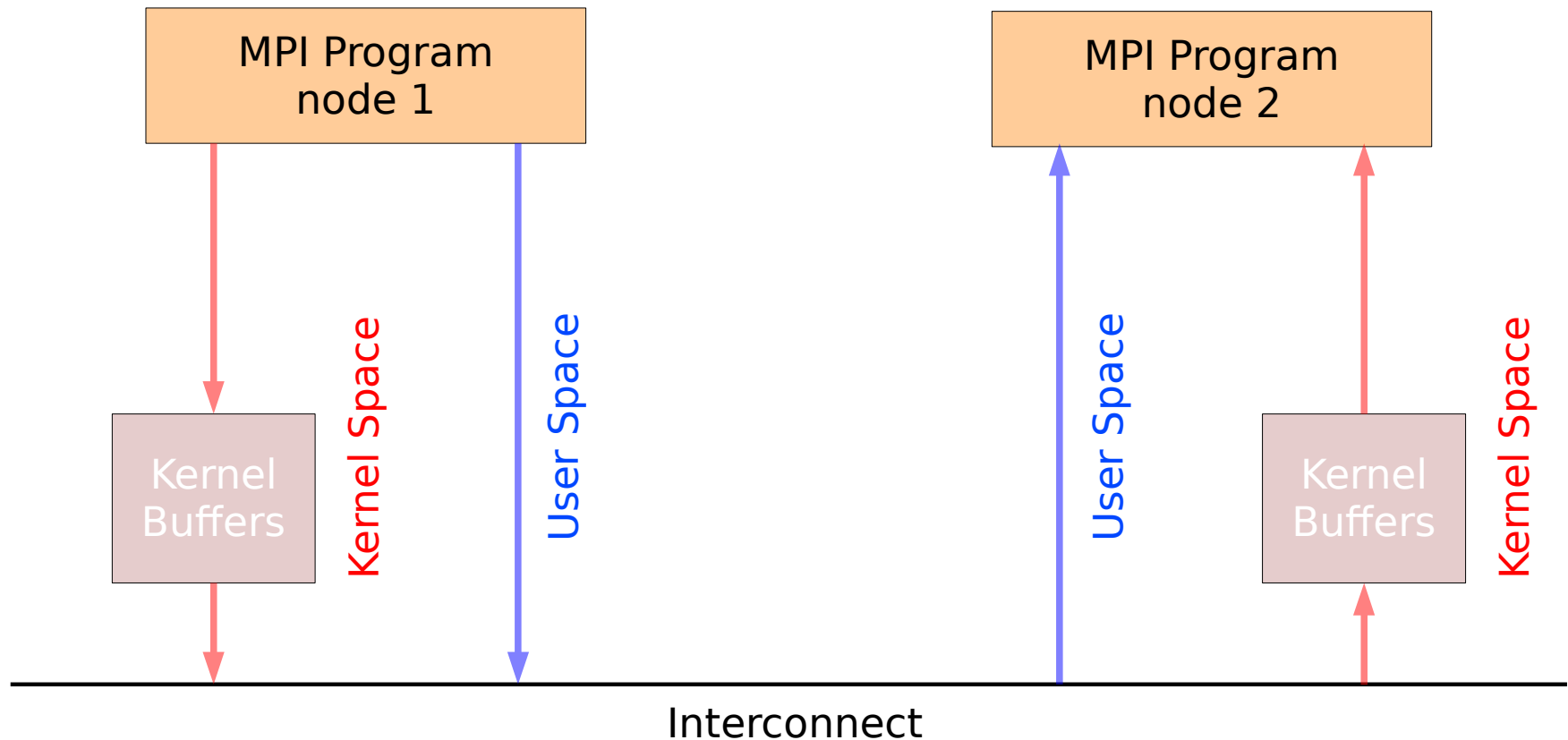




# Low Latency Networks

## Kernel Space vs User Space

Sending a message  
(block of memory)  
from node1 to node2





# High-Speed Network Devices

Technology	Blt Rate	
	Gbit/s	Mbyte/s
Gigabit Ethernet	1	125
Myrinet 2000	2	250
Infiniband SDR 1x	2	250
RapidIO Gen1 1x	2.5	312.5
Infiniband DDR 1x	4	500
RapidIO Gen2 1x	5	625
Infiniband SDR 4x	8	1000
Infiniband QDR 1x	8	1000
RapidIO Gen2 2x	10	1250
RapidIO Gen1 4x	10	1250
Myri-10G	10	1250
10 Gigabit Ethernet	10	1250
Infiniband FDR-10 1x	10.31	1290
Infiniband FDR 1x	13.64	1700
Infiniband DDR 4x	16	2000
RapidIO Gen2 4x	20	2500

Technology	Blt Rate	
	Gbit/s	Mbyte/s
Infiniband SDR 12x	24	3000
Infiniband EDR 1x	25	3125
Infiniband QDR 4x	32	4000
RapidIO Gen2 8x	40	5000
40 Gigabit Ethernet	40	5000
Infiniband FDR-10 4x	41.25	5160
Infiniband DDR 12x	48	6000
Infiniband FDR 4x	54.54	6820
RapidIO Gen2 16x	80	10000
Infiniband QDR 12x	96	12000
Infiniband EDR 4x	100	12500
100 Gigabit Ethernet	100	12500
Infiniband FDR-10 12x	123.75	15470
Infiniband FDR 12x	163.64	20450
Infiniband EDR 12x	300	37500

(as of Jan 2015)

[http://en.wikipedia.org/wiki/List\\_of\\_device\\_bandwidths](http://en.wikipedia.org/wiki/List_of_device_bandwidths)



# Final remarks

- High speed networks should have:
  - high bandwidth
  - high throughput
  - low latency
- Things to keep in mind:
  - choose the right topology for your needs (both physical and logical)
  - figure out what will be your typical data patterns (small/large chunks, frequent access, ...)
  - bet on reliable hardware
  - consider the cost



# That's All Folks!

Copyright 2006 by Randy Glasbergen.  
www.glasbergen.com



"Network is down."

```
( questions ; comments ) | mail -s uheilaaa baro@democritos.it
```

```
( complaints ; insults ) &>/dev/null
```



# REFERENCES AND USEFUL LINKS

## SOFTWARE:

- Linux Kernel <http://www.kernel.org>
- Netfilter <http://www.netfilter.org>
- nmap <http://www.insecure.org/nmap/>
- hping <http://www.hping.org/>
- netcat <http://netcat.sourceforge.net/>
- iptstate <http://www.phildev.net/iptables/>
- ss <http://linux-net.osdl.org/index.php/lproute2>
- lsof <ftp://lsof.itap.purdue.edu/pub/tools/unix/lsof/>
- netstat <http://www.tazenda.demon.co.uk/phil/net-tools/>
- tcpdump <http://www.tcpdump.org>
- Wireshark <http://www.wireshark.org>
- ethereal <http://www.ethereal.com> (see Wireshark)
- iptraf <http://iptraf.seul.org/>
- Ettercap <http://ettercap.sourceforge.net>
- Dsniff <http://www.monkey.org/~dugsong/dsniff/>
- tcptraceroute <http://michael.toren.net/code/tcptraceroute/>
- (telnet, traceroute, ping, ...)

## DOC:

- IPTables HOWTO <http://www.netfilter.org/documentation/HOWTO/>
- IPTables tutorial <http://iptables-tutorial.frozentux.net/>
- Having fun with IPTables <http://www.ex-parrot.com/~pete/upside-down-ternet.html>
- Denial of Service [http://www.cert.org/tech\\_tips/denial\\_of\\_service.html](http://www.cert.org/tech_tips/denial_of_service.html)
- IPv4 Address space
  - <http://www.cymru.com/Documents/bogon-bn.html>
  - <http://www.iana.org/assignments/ipv4-address-space>
  - <http://www.oav.net/mirrors/cidr.html>
  - <http://en.wikipedia.org/wiki/IPv4>
  - IANA <http://www.iana.org>
  - RIPE <http://www.ripe.net>
  - RFC 3330 <http://www.rfc.net/rfc3330.html>
- SANS: [http://www.sans.org/reading\\_room/whitepapers/firewalls/](http://www.sans.org/reading_room/whitepapers/firewalls/)  
[http://www.sans.org/reading\\_room/](http://www.sans.org/reading_room/)

## RFC: (<http://www.rfc.net>)

- RFC 791 – Internet Protocol (IPv4)  
<http://www.rfc.net/rfc791.html>
- RFC 793 – Transmission Control Protocol (TCP)  
<http://www.rfc.net/rfc793.html>
- RFC 768 – User Datagram Protocol (UDP)  
<http://www.rfc.net/rfc768.html>
- RFC 792 – Internet Control Message Protocol (ICMP)  
<http://www.rfc.net/rfc792.html>
- RFC 1180 – A TCP/IP Tutorial  
<http://www.rfc.net/rfc1180.html>
- RFC 1700 / IANA db – Assigned Numbers  
<http://www.rfc.net/rfc1700.html>  
<http://www.iana.org/numbers.html>
- RFC 3330 – Special-Use IPv4 Addresses  
<http://www.rfc.net/rfc3330.html>
- RFC 1918 – Address Allocation for Private Internets  
<http://www.rfc.net/rfc1918.html>
- RFC 2196 – Site Security Handbook  
<http://www.rfc.net/rfc2196.html>
- RFC 2827 – Network Ingress Filtering  
<http://www.rfc.net/rfc2827.html>
- RFC 2828 – Internet Security Glossary  
<http://www.rfc.net/rfc2828.html>
- RFC 1149 – Transmission of IP Datagrams on Avian Carriers  
<http://www.rfc.net/rfc1149.html>
- Unofficial CIPW WG  
<http://www.blug.linux.no/rfc1149/>
- RFC 2549 – IP over Avian Carriers with Quality of Service  
<http://www.rfc.net/rfc2549.html>
- Firewalling the CIPW  
<http://www.tibonia.net/>  
<http://www.hotink.com/wacky/dastrdly/>





# Some acronyms...

**IP** – Internet Protocol  
**TCP** – Transmission Control Protocol  
**UDP** – User Datagram Protocol  
**ICMP** – Internet Control Message Protocol  
**ARP** – Address Resolution Protocol  
**MAC** – Media Access Control

**OS** – Operating System  
**NOS** – Network Operating System  
**LINUX** – LINUX is not UNIX

**PING** – Packet Internet Groper

**FTP** – File Transfer Protocol – (TCP/21,20)  
**SSH** – Secure SHell – (TCP/22)  
**TELNET** – Telnet – (TCP/23)  
**SMTP** – Simple Mail Transfer Protocol – (TCP/25)  
**DNS** – Domain Name System – (UDP/53)  
**NTP** – Network Time Protocol – (UDP/123)  
**BOOTPS** – Bootstrap Protocol Server (**DHCP**) – (UDP/67)  
**BOOTPC** – Bootstrap Protocol Server (**DHCP**) – (UDP/68)  
**TFTP** – Trivial File Transfer Protocol – (UDP/69)  
**HTTP** – HyperText Transfer Protocol – (TCP/80)  
**NTP** – Network Time Protocol – (UDP/123)  
**SNMP** – Simple Network Management Protocol – (UDP/161)  
**HTTPS** – HyperText Transfer Protocol over TLS/SSL – (TCP/443)  
**RSH** – Remote Shell – (TCP/514,544)

**ISO** – International Organization for Standardization  
**OSI** – Open System Interconnection

**TLS** – Transport Layer Security  
**SSL** – Secure Sockets Layer

**RFC** – Request For Comments

**ACL** – Access Control List

**PDU** – Protocol Data Unit

## **TCP flags:**

- **URG**: Urgent Pointer field significant
- **ACK**: Acknowledgment field significant
- **PSH**: Push Function
- **RST**: Reset the connection
- **SYN**: Synchronize sequence numbers
- **FIN**: No more data from sender

## **RFC 3168 TCP flags:**

- **ECN**: Explicit Congestion Notification
- (**ECE**: ECN Echo)
- **CWR**: Congestion Window Reduced

**ISN** – Initial Sequence Number