

# Optimisation del ancho de banda (Networking Basics)

 Christian Benvenuti  
([christian.benvenuti@libero.it](mailto:christian.benvenuti@libero.it))

La Paz, Bolivia, 4-9/5/2009



# Do you know ...

- how to use binary operators?
- the difference between L2 and L3 addresses?
- the difference between L2 and L3 networks?
- how subnetting works?
- the difference between Routing and Switching?
- what NAT is and how it is used?

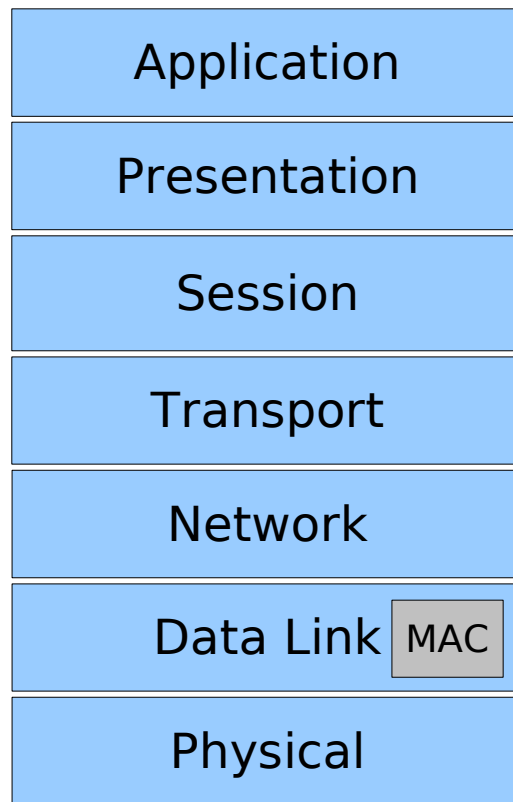
# Agenda

- Evolution of computer networks
- Protocol suites (ISO/OSI vs TCP/IP)
- Most common network device types  
(classified by layers where they operate and functionalities)
- L2 networks VS L3 networks
- Basic network design rules
- Basics on Addressing/Subnetting
- Common Network Services
- Where does Linux fit into the network?

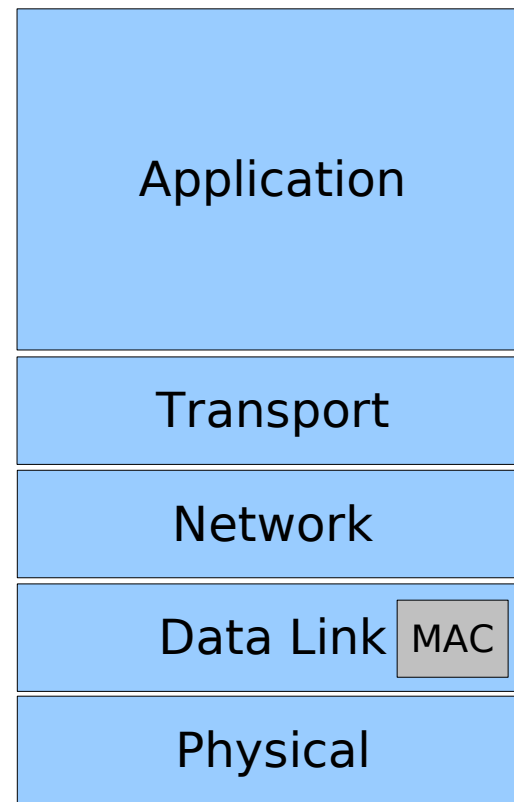
# Evolution of the role of computer networks

- 10 Years ago:
  - Can the use of “the network” come useful to this problem?
- Today:
  - Is there a valid reason not to use “the network” to address this problem?
- Future
  - I can't imagine a world without the Internet!

# The most common slide you'll find in a "Networking Basics" presentation

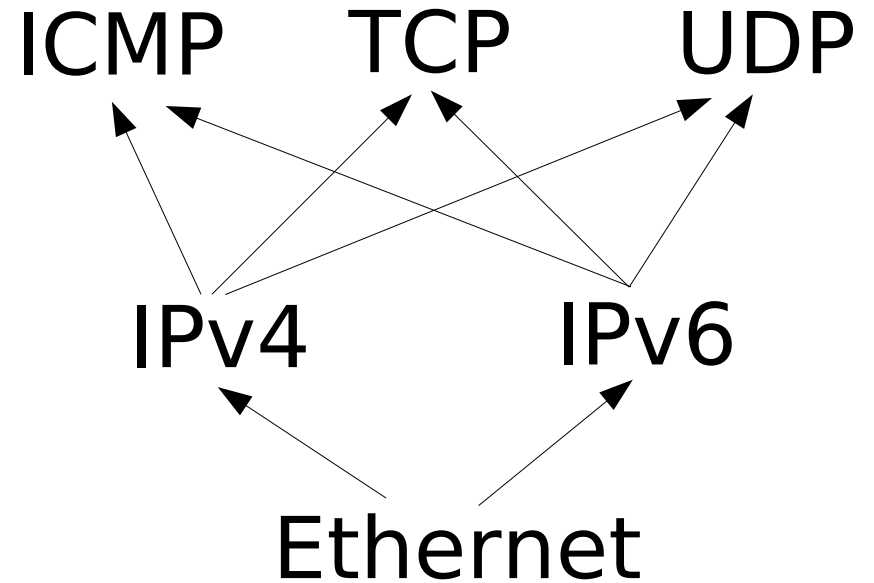
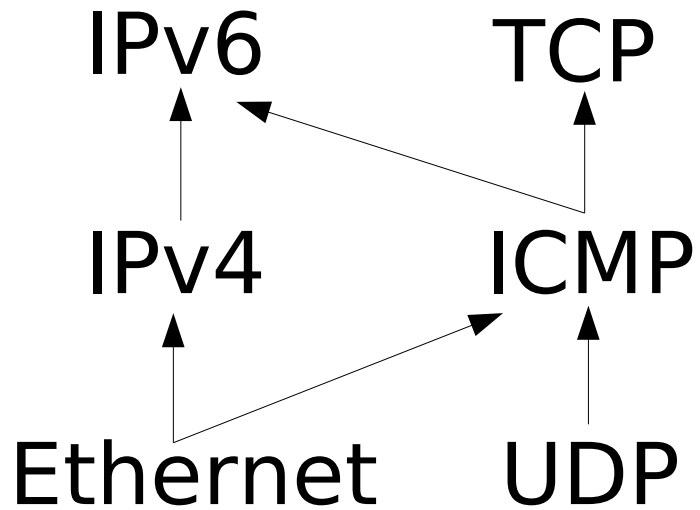


**ISO-OSI**



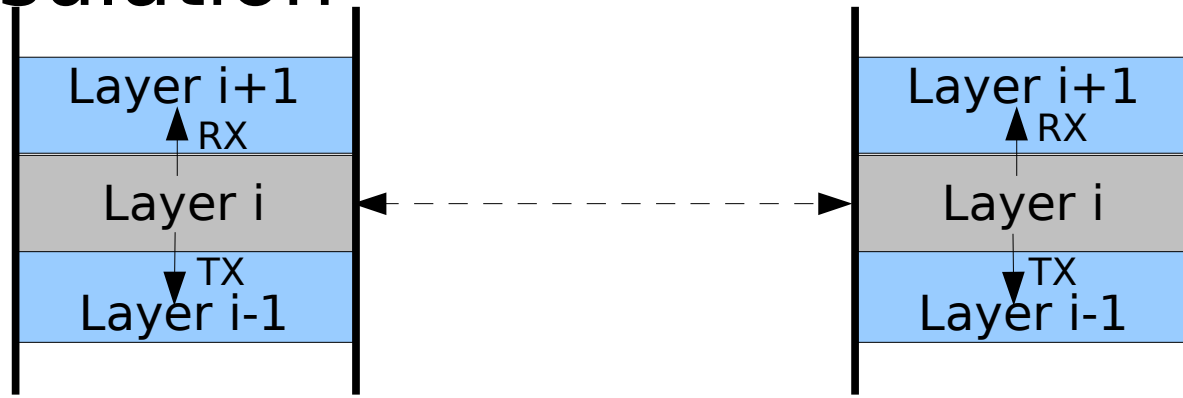
**TCP/IP**

# What's wrong here?



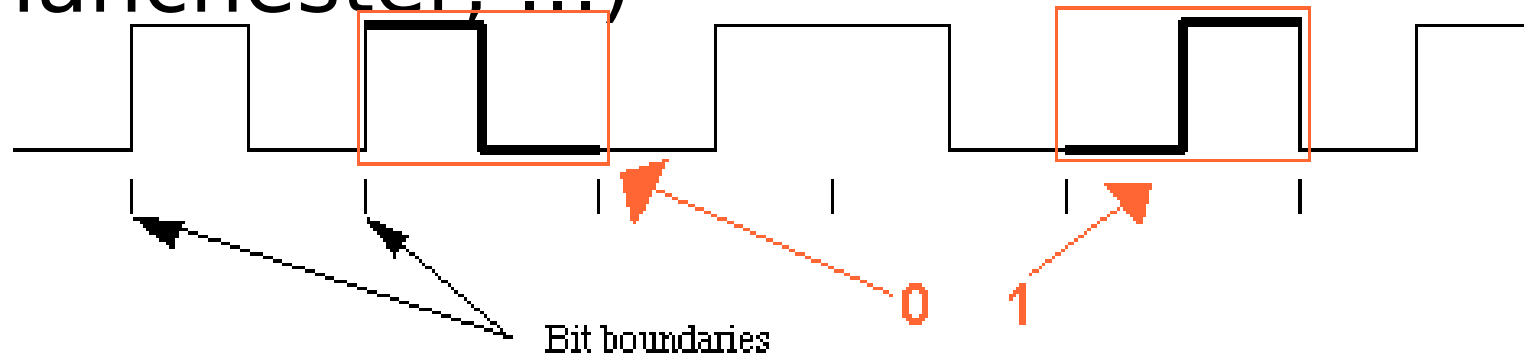
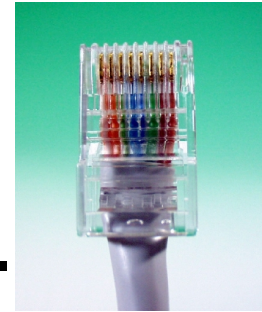
# Why a network stack?

- Need of rules
- Need to reduce the design complexity
- Distinction between the functions of the upper and lower layers
- Peer-2-Peer communication + encapsulation



# The *Physical* Layer

- Transmission media
  - Coaxial, UTP, Fiber, Wireless, ..
- Signaling of bits (RZ, NRZ, Manchester, ...)



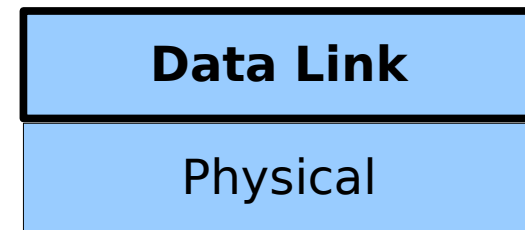
- Timing of bits (Sync vs Async)

Physical

TCP/IP

# The *Data-Link* Layer (1/2)

- Framing
- Error Detection/Correction (CRC)
- Simplex/Half/Full duplex
- ...

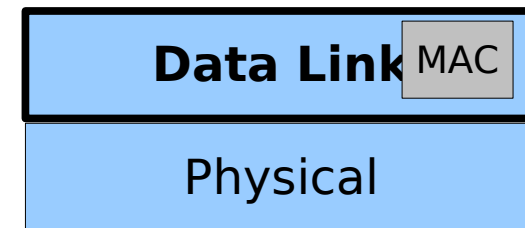


TCP/IP

# The ***Data-Link*** Layer, (2/2)

## The Medium Access Control (MAC)

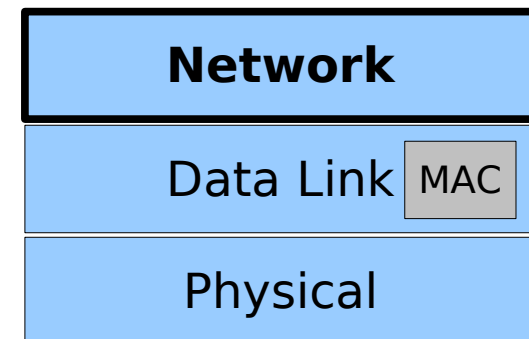
- Collision free
  - Static (TDMA, ...)
  - Dynamic (Token, Bitmap, ...)
- Not Collision Free
  - CSMA [p-persistent] /CD
- Wireless mediums use different algorithms. (ask the wireless gurus in the room ...)



**TCP/IP**

# The ***Network*** Layer

- Here is where the interesting stuff starts ...
  - Physical to logical boundary
  - End-to-end hosts communication
    - static/dynamic routing, addressing and subnetting, ...
  - Fragmentation/Defragmentation
  - ...

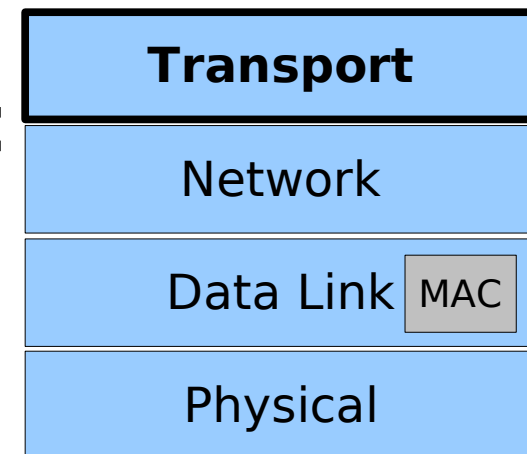


**TCP/IP**

# The *Transport* Layer

- (Logical) Connection multiplexing
- Flow control
- Error detection (data corruption)
- ...

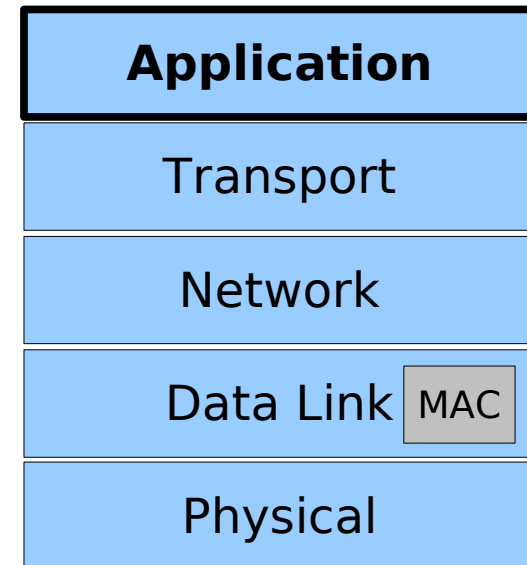
- Most common transport protocols:
  - UDP (unreliable)
  - TCP (reliable)
  - ICMP (control data)



**TCP/IP**

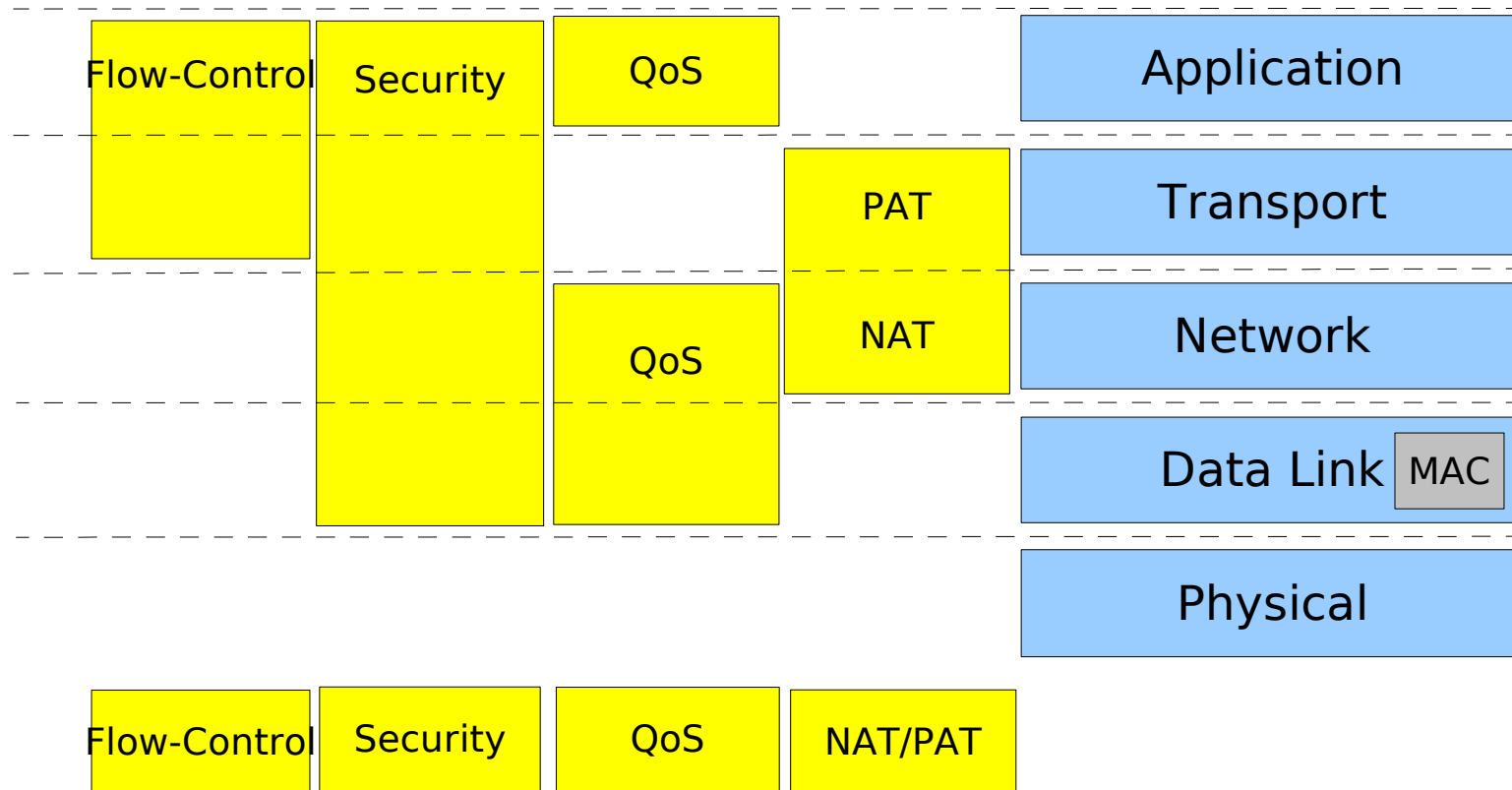
# The *Application* Layer

- This is the layer where the applications (users interface with) are located.

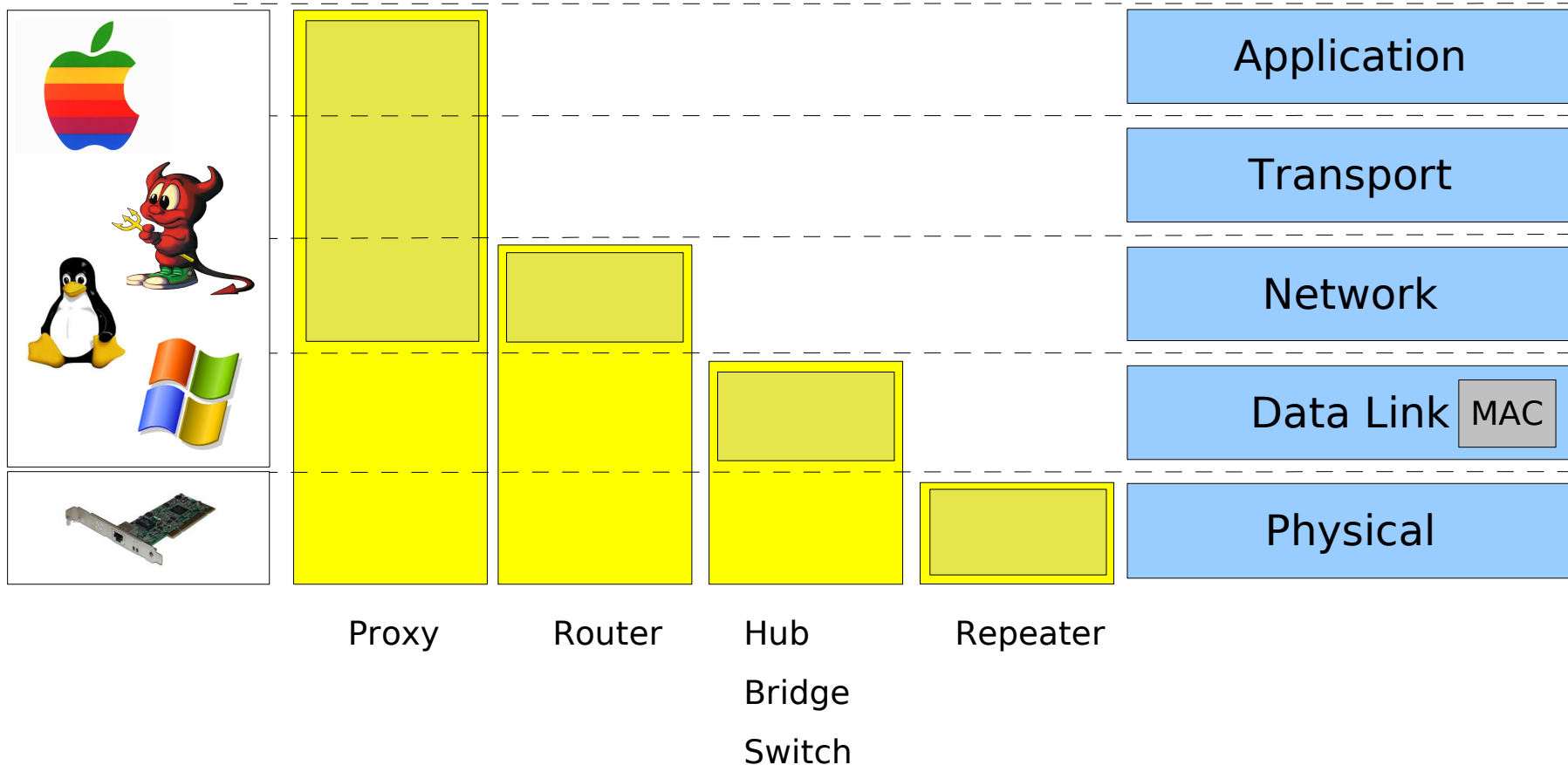


TCP/IP

# Let's assign few common features to the right layer ...



# Network stacks ... everything clear, right? Let's check ...



What about Firewalls, Intrusion Detection Systems, etc ?

# Network **size** and topology

- Size

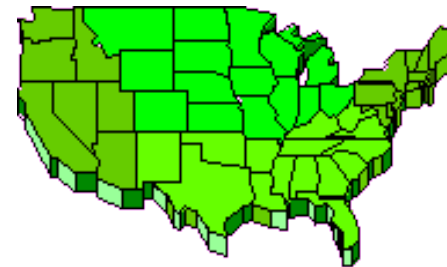
- Local Area Network (LAN)



- Metropolitan Area Network (MAN)



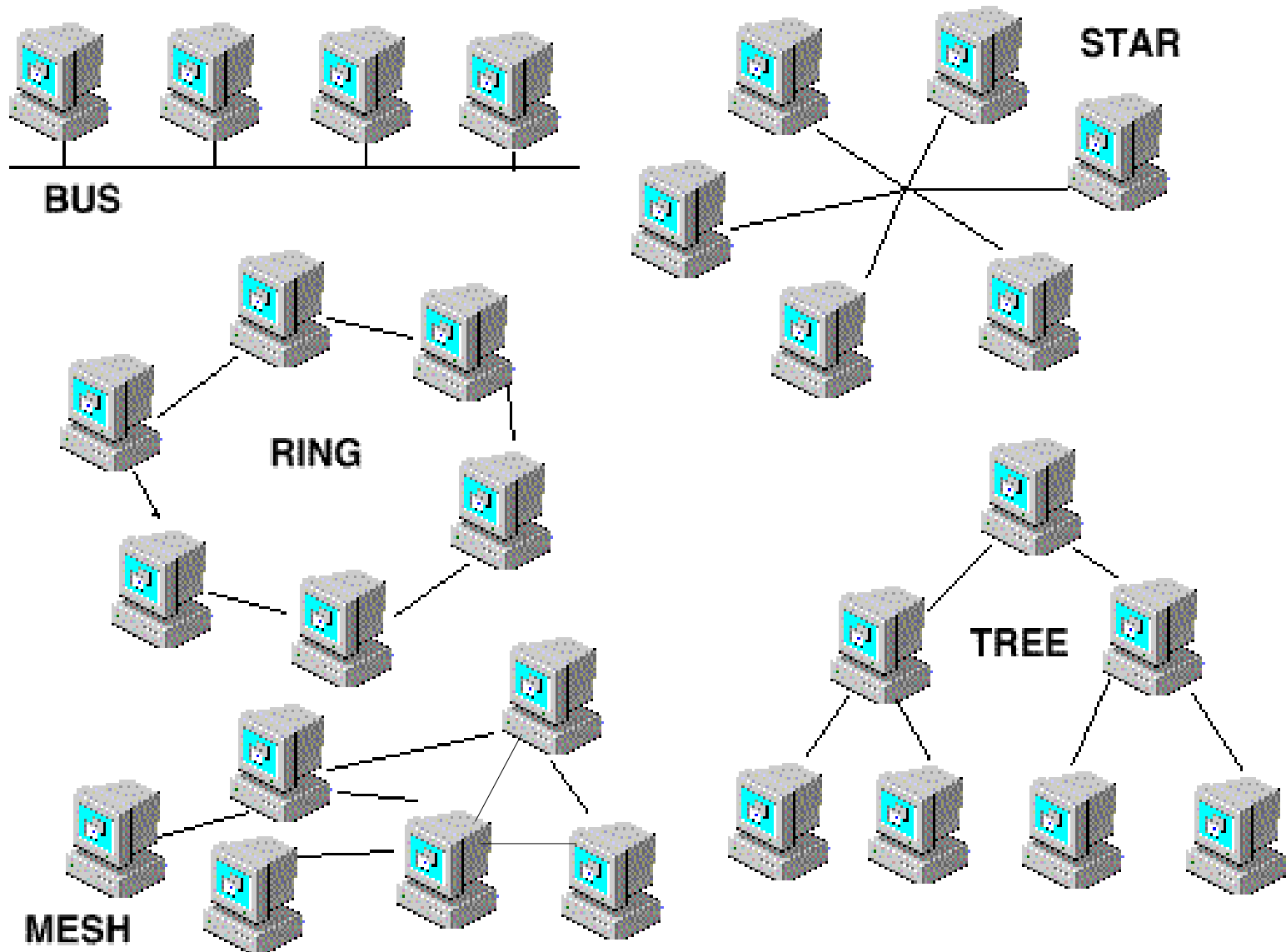
- Wide Area Network (WAN)



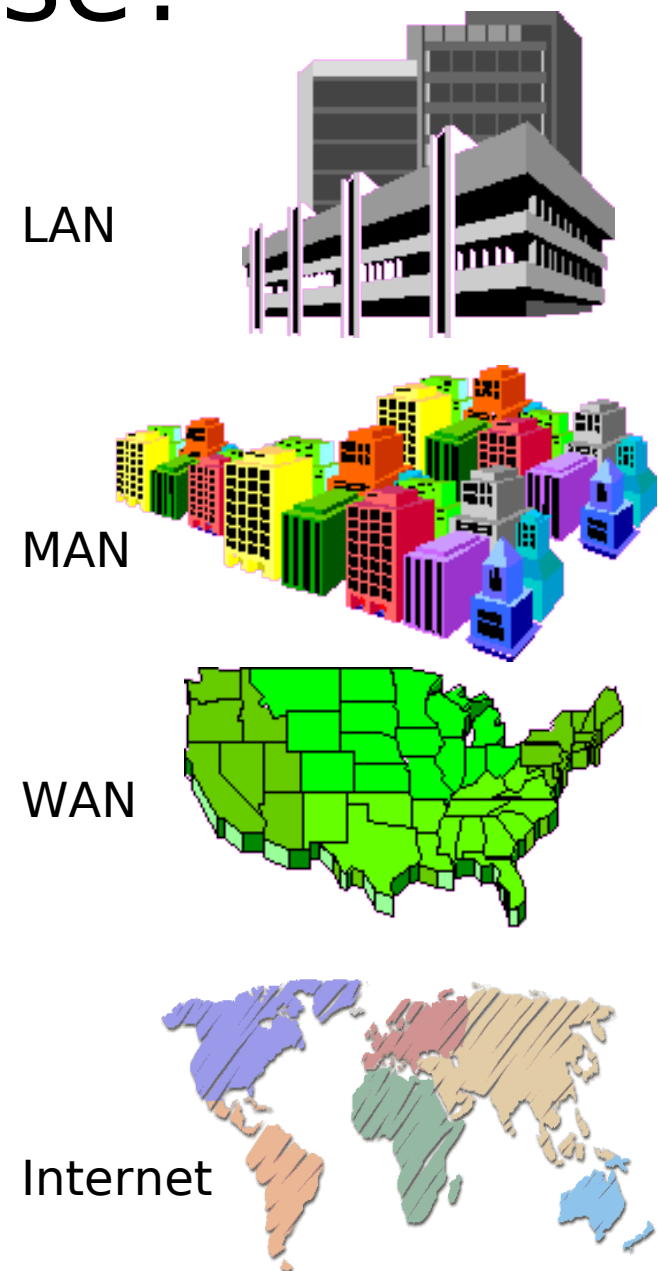
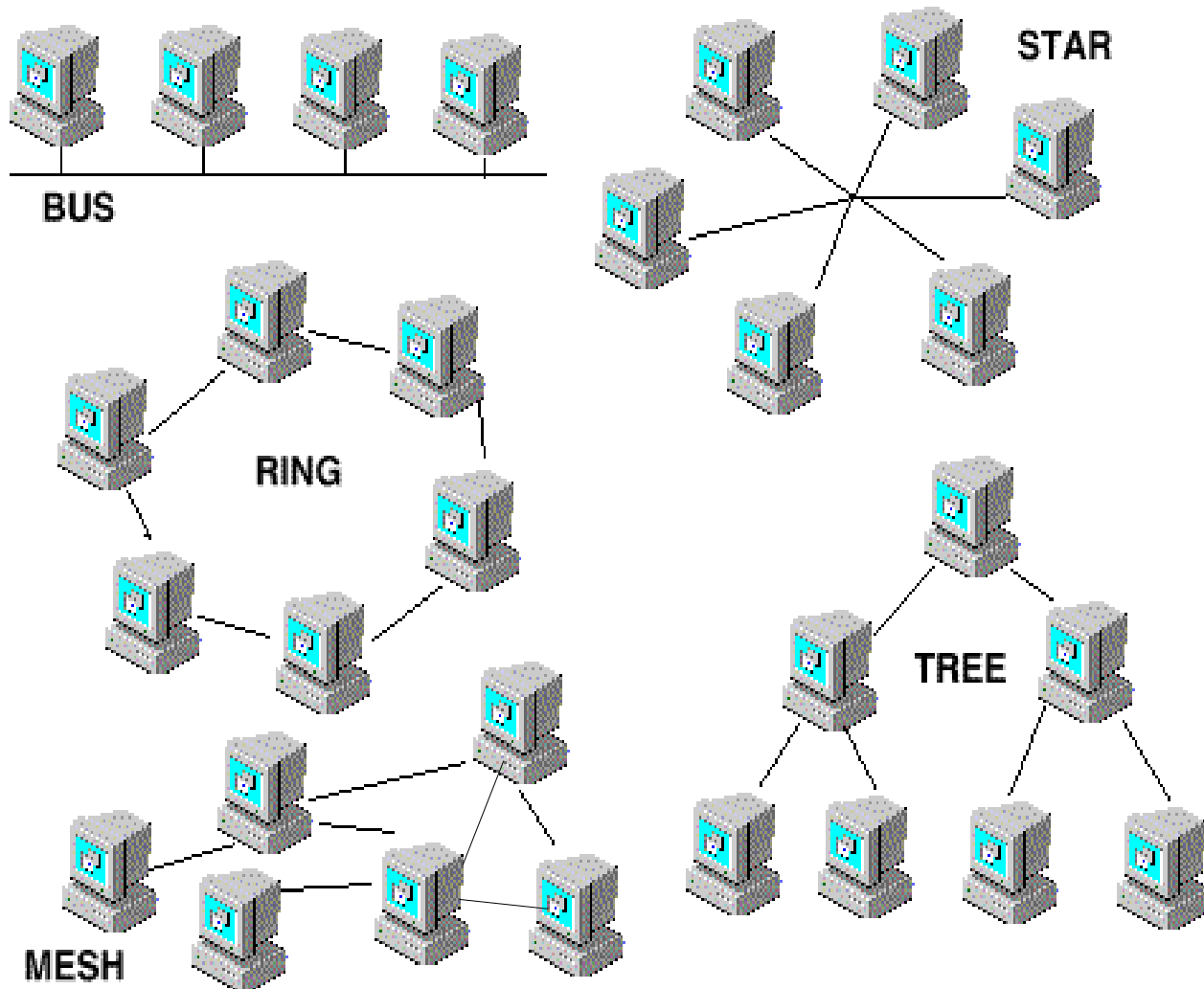
- The Internet



# Network size and **topology**



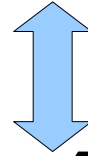
# What combinations do you think make more sense?



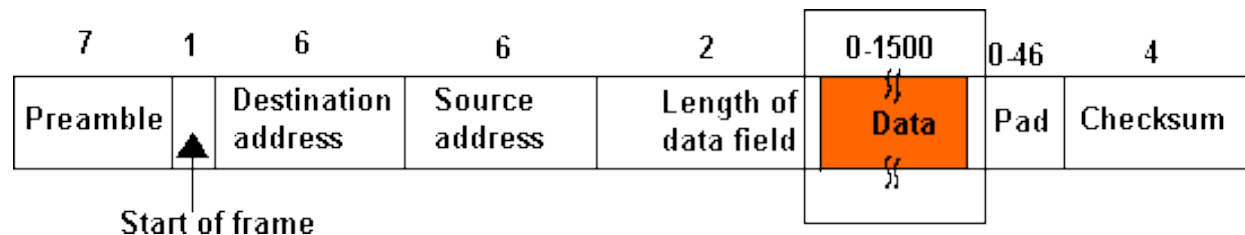
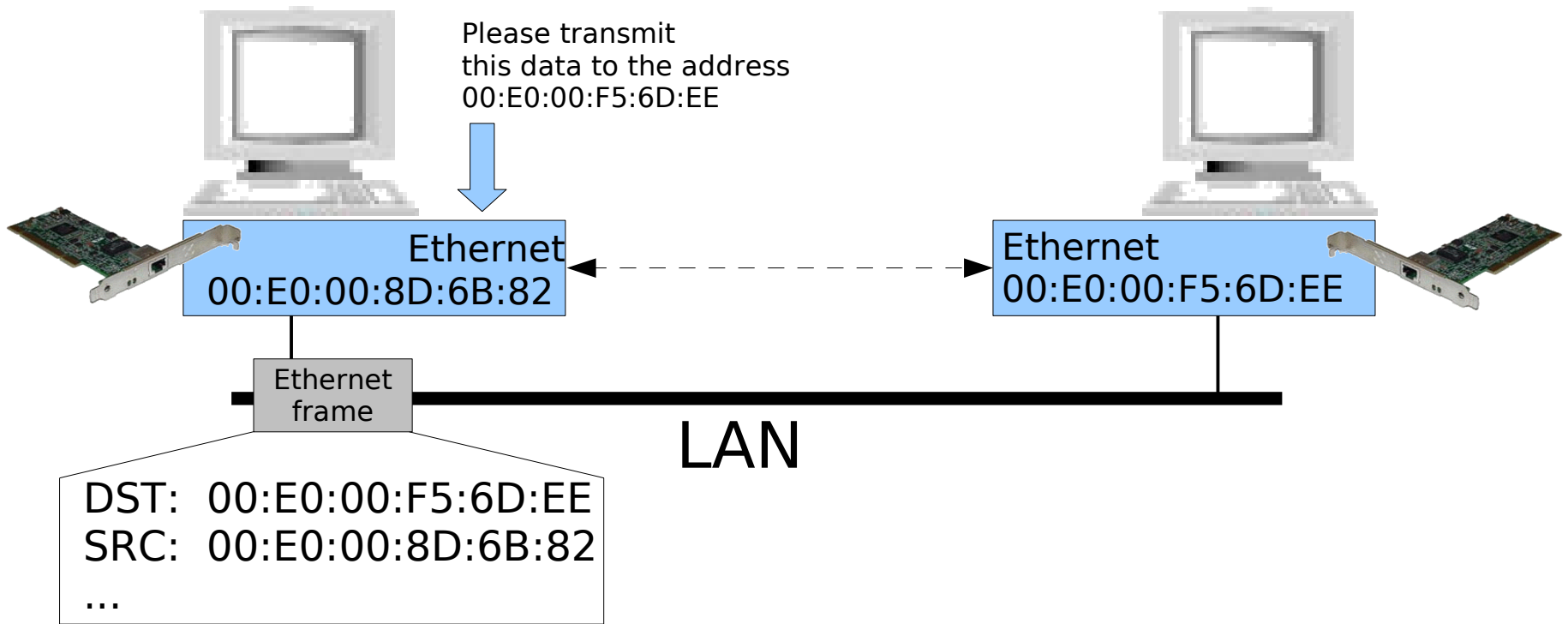
# The L2/L3 de-facto standards

- Wired LANs
  - Ethernet
    - Fast/Giga/10-Giga
- Wireless LANs
  - 802.11b, ... *<this and much more (ie., not only LAN) will be the focus of the next three weeks>*
- Internet Protocol Version 4 (IPv4)
  - Its younger brother IPv6 is not yet as widely used as IPv4.

# Network (IPv4)



# Data-Link (Ethernet)

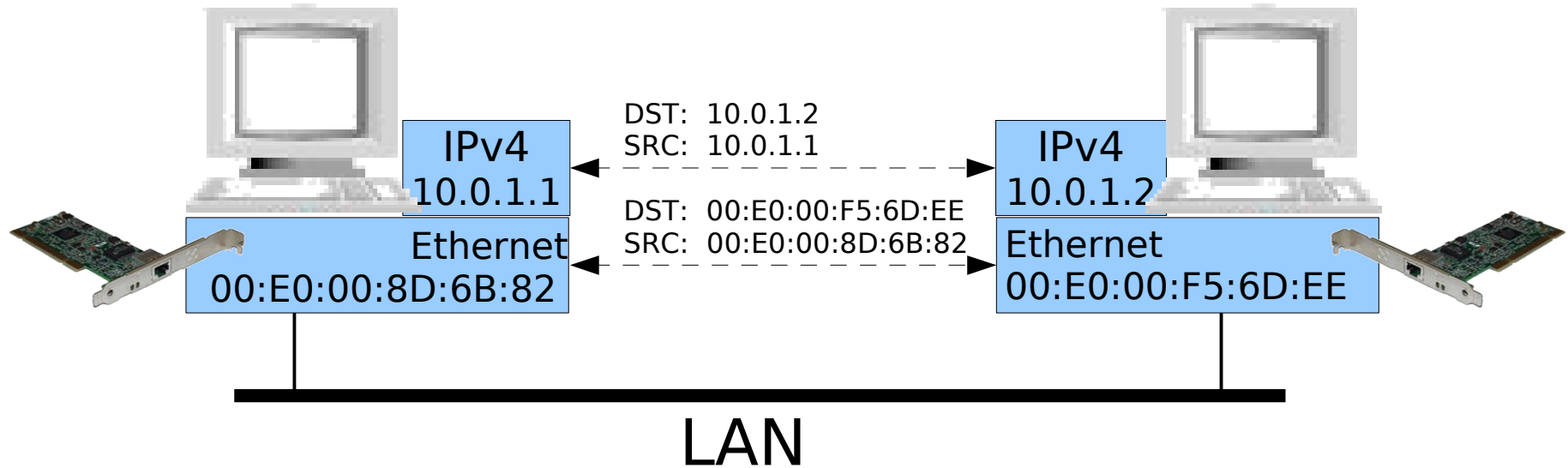


Structure of an Ethernet frame

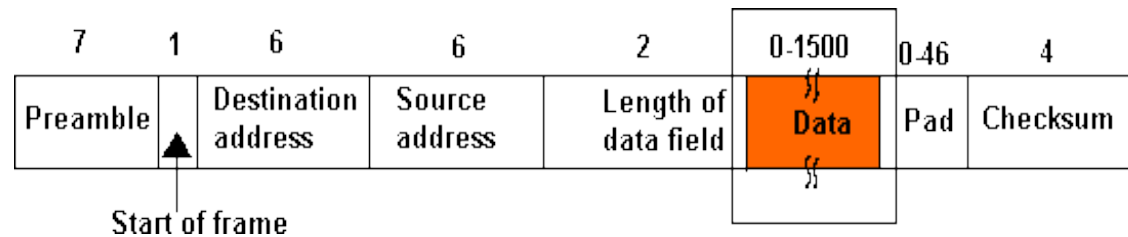
# Network (IPv4)



# Data-Link (Ethernet)

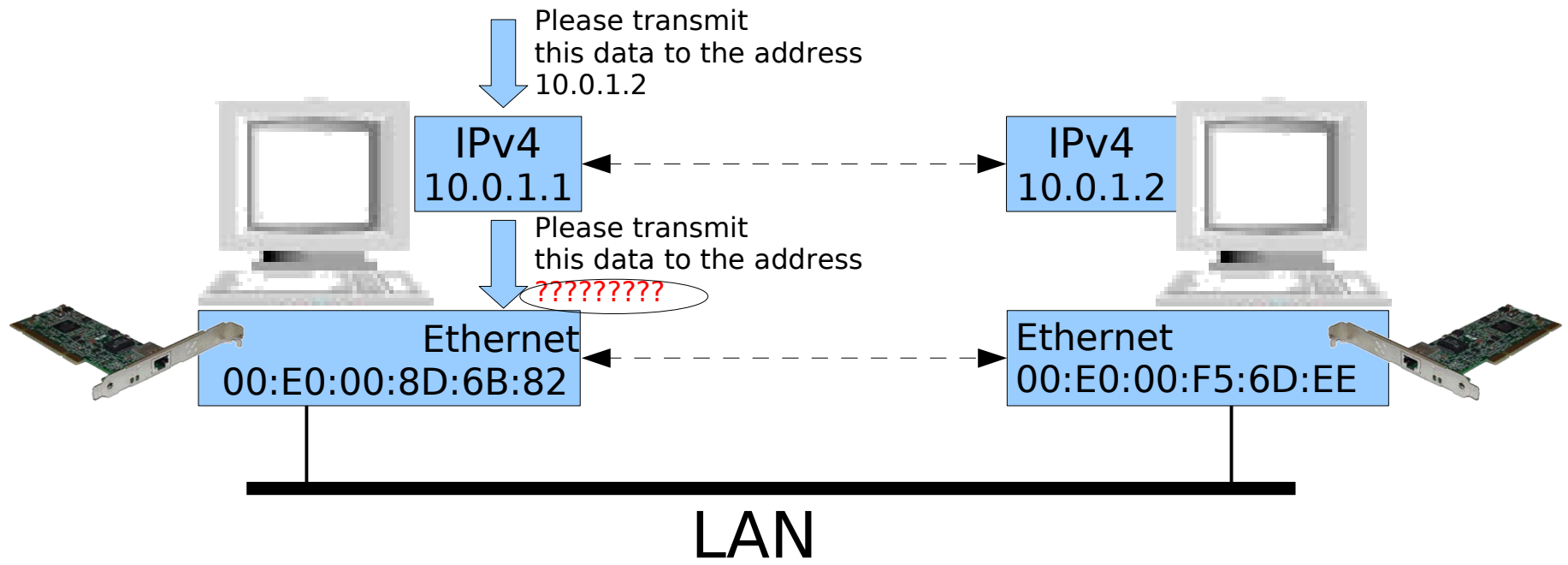


0	4	8	16	19	31
VERS	LEN	Type of Service	Total Length		
Identification			Flags	Fragment Offset	
TTL	Protocol		Header checksum		
source IP address					
destination IP address					
Options		.. ..	padding		
data					
....					
....					



# Network (IPv4) (Ethernet)

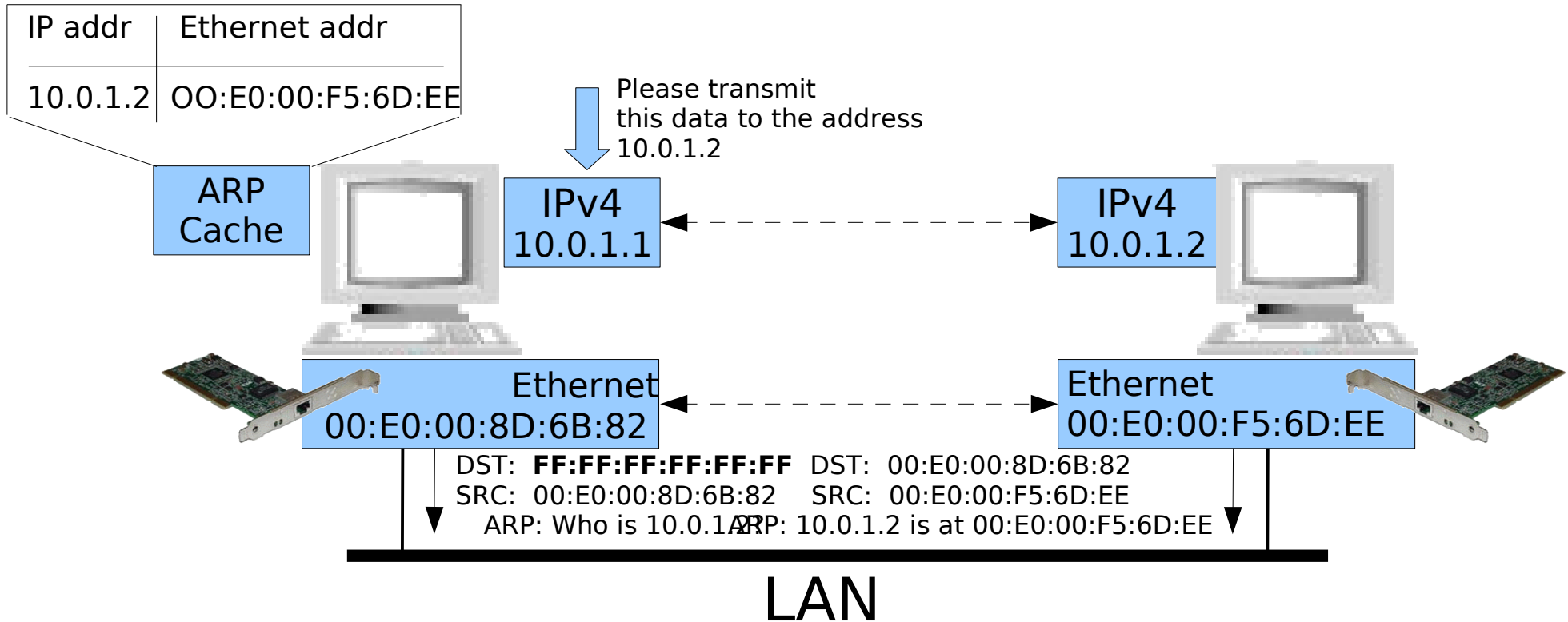
# Data-Link



# Network (IPv4)



# Data-Link (Ethernet)



# More on **Network** <--> **Data-Link**: Fragmentation (1/2)

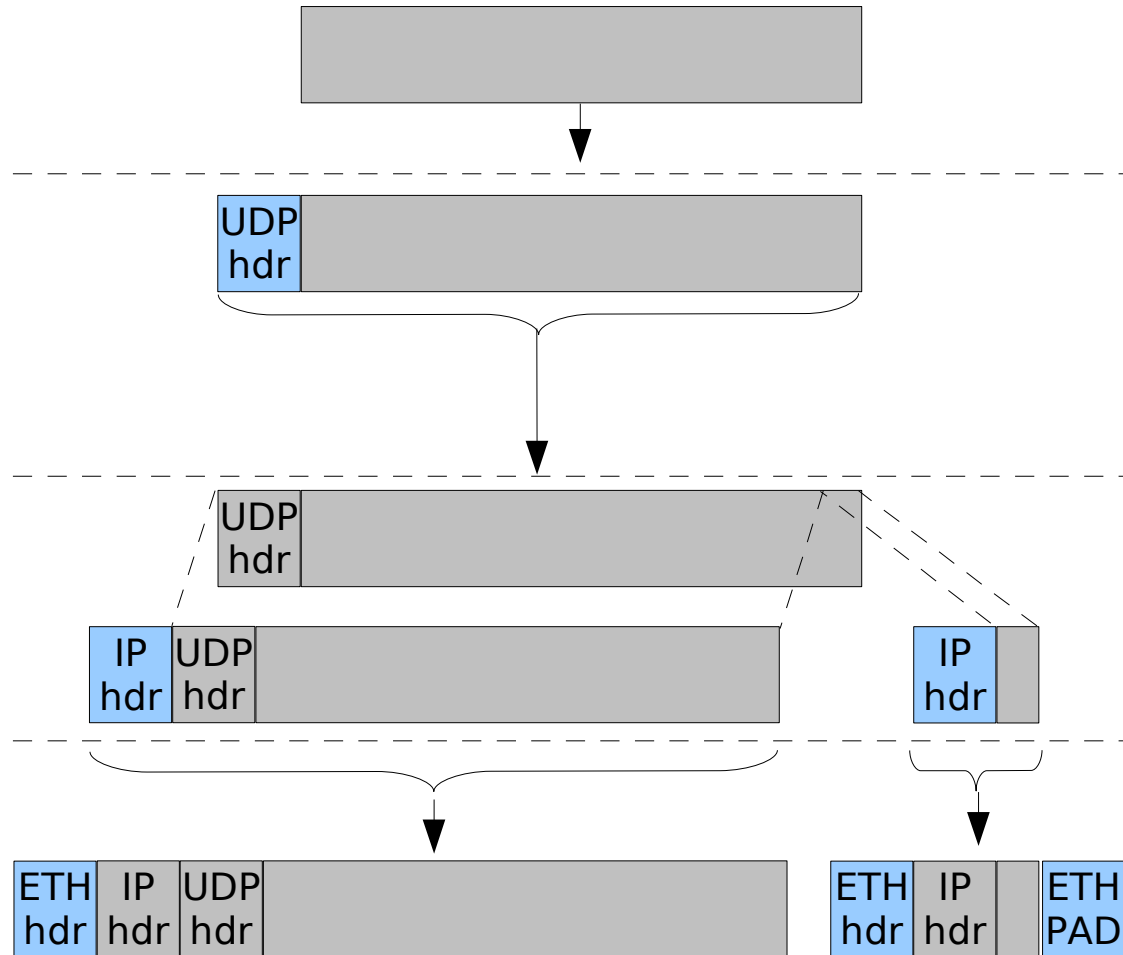
- Fragmentation is needed each time the size of the block of data (PDU) a layer needs to transmit exceeds the maximum size handled by the lower layer.
- There are good reasons to avoid fragmentation whenever possible
  - PMTU, IPv6, ...
- Each layer uses a different name for its PDU (and the associated maximum/minimum sizes)
  - L1:*Bit*, L2:*Frame*, L3:*Datagram* L4:*Segment*, L5:*Message*
  - Generic: *Packet*

# More on **Network** <--> **Data-Link**: Fragmentation (2/2)

UDP	TCP
Basic hdr: 8 Bytes	Basic hdr: 20 Bytes
Min: /	Min: /
Max: 64 KBytes	Max: /

IPv4	
Basic hdr: 20 Bytes	
Min: /	Max: 64 KBytes [PMTU ...]

Ethernet	
(hdr size = 14 bytes)	
Min: 64 Bytes	Max: 1514/18 Bytes (configurable)



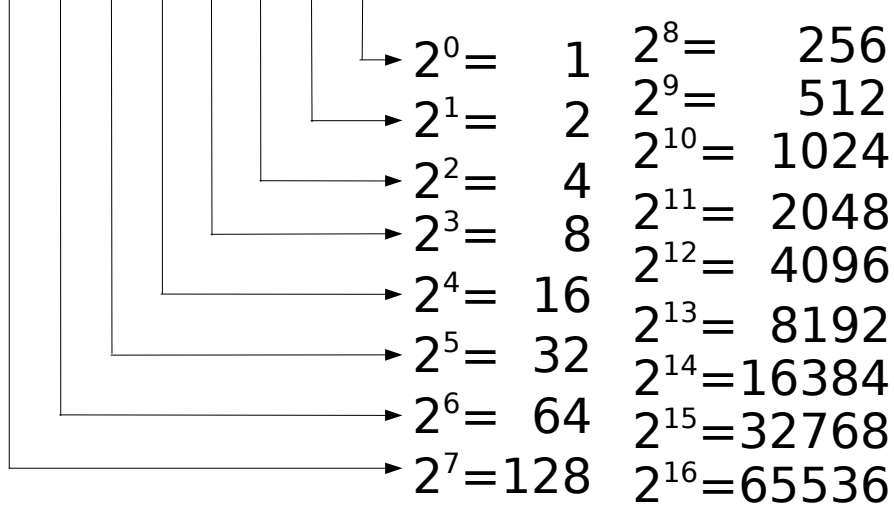
# More on the **Network** Layer

- Quick review of binary numbers/operators
- Addressing
- Subnetting
- Routing

# Binary numbers

(MSB) 8<sup>th</sup> 7<sup>th</sup> 6<sup>th</sup> 5<sup>th</sup> 4<sup>th</sup> 3<sup>rd</sup> 2<sup>nd</sup> 1<sup>st</sup> (LSB)

X X X X X X X X



- A** Commonly used with broadcasts
- B** Commonly used with netmasks

Binary	Decimal	Hex
00000000	0	0x00
0000000 <b>1</b>	1	0x01
000000 <b>10</b>	2	0x02
00000 <b>100</b>	4	0x04
0000 <b>1000</b>	8	0x08
000 <b>10000</b>	16	0x10
00 <b>100000</b>	32	0x20
0 <b>1000000</b>	64	0x40
<b>10000000</b>	128	0x80

**A**

Binary	Decimal	Hex
00000000	0	0x00
0000000 <b>1</b>	1	0x01
000000 <b>11</b>	3	0x03
00000 <b>111</b>	7	0x07
0000 <b>1111</b>	15	0x0F
000 <b>11111</b>	31	0x1F
00 <b>111111</b>	63	0x3F
0 <b>1111111</b>	127	0x7F
<b>11111111</b>	255	0xFF

**B**

Binary	Decimal	Hex
<b>11111111</b>	255	0xFF
<b>11111110</b>	254	0xFE
<b>11111100</b>	252	0xFC
<b>11111000</b>	248	0xF8
<b>11110000</b>	240	0xF0
<b>11100000</b>	224	0xE0
<b>11000000</b>	192	0xC0
<b>10000000</b>	128	0x80
00000000	0	0x00

# Binary operators

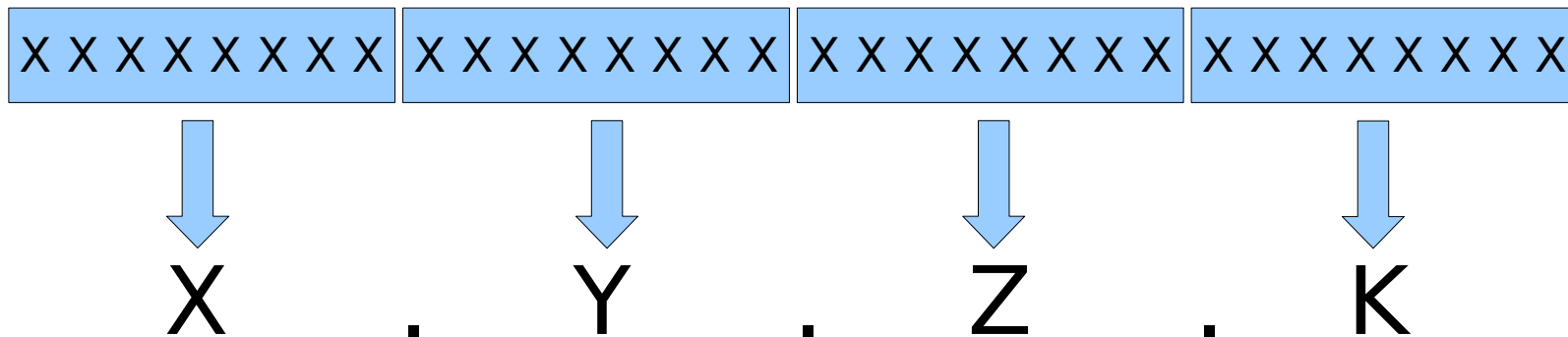
AND	0	1
0	0	0
1	0	1

OR	0	1
0	0	1
1	1	1

There are many more but we need only those two in this class

# Structure of an IPv4 address

- 32 bits



- They range from 0.0.0.0 to 255.255.255.255  
(more than 4 billions addresses)



# Structure of an IPv4 address

- Given an IP address, the netmask (and the broadcast) address is derived from its class (but you can change both)
- Public VS Private addresses
- Unicast, Multicast, Broadcast addresses

# Structure of an IPv4 address



	From	To	#Networks	#Hosts per network
Class A	0.0.0.0	127.255.255.255	127	16.777.216
Class B	128.0.0.0	192.255.255.255	16.129	65.536
Class C	192.0.0.0	223.255.255.255	2.097.152	256
Class D	224.0.0.0	239.255.255.255		
Class E	240.0.0.0	255.255.255.255		

IP addr: 140.105.16.50 →   
 { Class B  
 Default Netmask is 255.255.0.0 (or /16)  
 Default Broadcast is 140.105.255.255

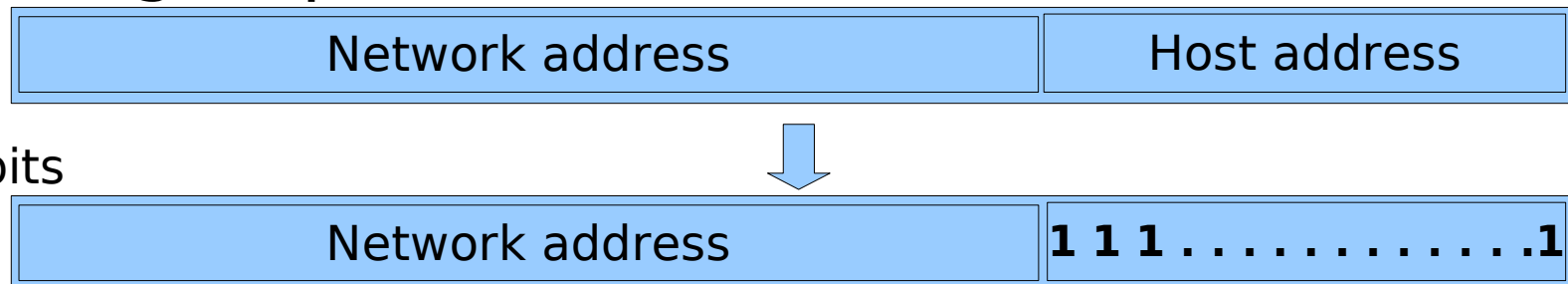
# Private Addresses

- **10.0.0.0/8**                      1 x Class A      10 = **0 0 0 0 1 0 1 0**
- **172.16.0.0/16**                  16 x Class B      172 = **1 0 1 0 1 1 0 0**
- **192.168.0.0/16**                256 x Class C      192 = **1 1 0 0 0 0 0 0**
  
- **127.0.0.0/8 loopback** (valid only on the local host)

# Unicast, Broadcast, Multicast

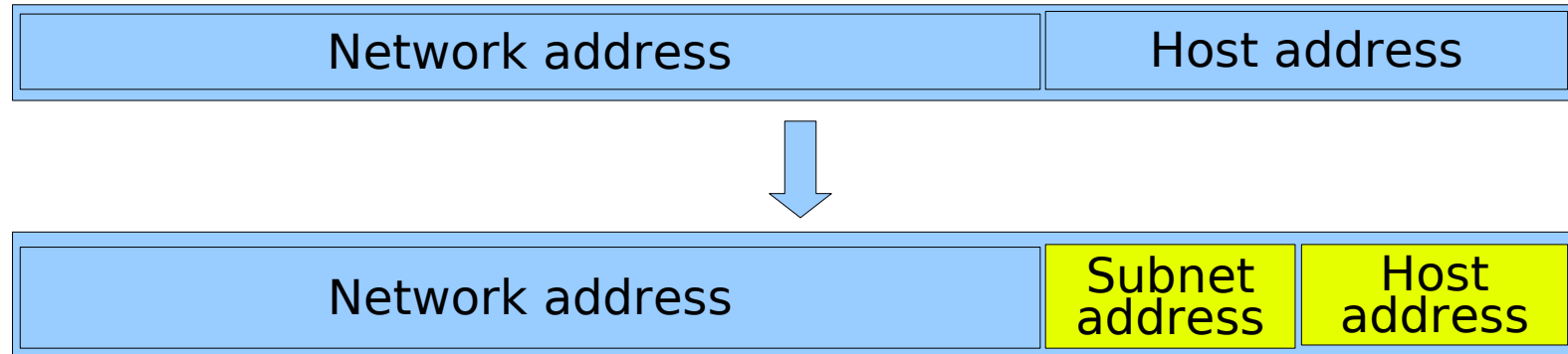
- The use of broadcast and multicast addresses is a convenient way for sending a packet to multiple recipients (link layer protocols use them too).
  - A broadcast message is addressed at all the hosts of a given network (or subnet).
    - Local VS Directed broadcasts
  - A multicast message is addressed at those hosts that subscribed to the associated multicast group.

The broadcast is built (by default) by setting to 1 all the bits of the host address



# Subnetting

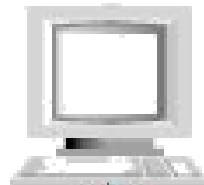
- The host address is split into two parts:
  - Subnet number
  - Host address



- It is no longer needed to use the default netmask derived from the class (i.e., /24, /16, /8)

# Example of subnetting (1/2)

- The IP address and the associated netmask together tell you what other (neighbor) IP addresses are directly connected (i.e., only 1 hop away).



192.168.1.100/24

LAN



192.168.1.100/24



Subnet address:	192.168.1.0/24
Network address:	192.168.1.0
Default broadcast:	192.168.1.255
<b>Range of addresses:</b>	<b>192.168.1.1 ... 192.168.1.255</b>
<b>#Hosts:</b>	<b>256-2 = 254</b>

# Example of subnetting (2/2)

Network address

Host address

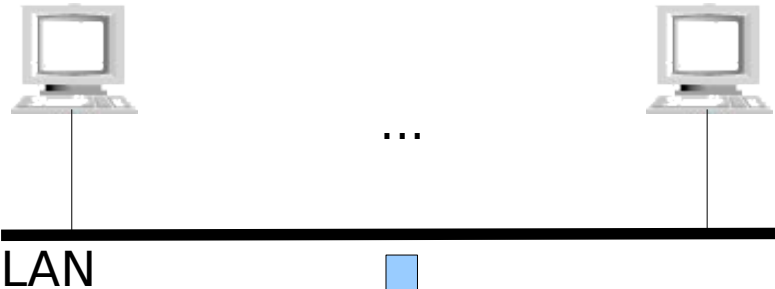


Network address

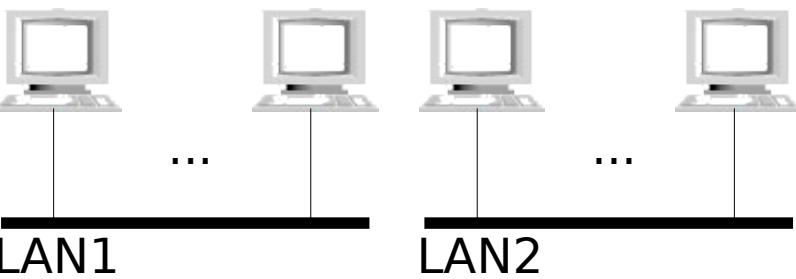
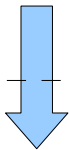
Subnet address

Host address

Let's suppose we wanted to create two subnets of the same size



Subnet address: 192.168.1.0/24  
Network address: 192.168.1.0  
Default broadcast: 192.168.1.255  
**Range of addresses: 192.168.1.1 ... 192.168.1.255**  
**#Hosts: 256-2 = 254**



Subnet address: 192.168.1.0/25  
Network address: 192.168.1.0  
Default broadcast: 192.168.1.127  
**Range of addresses: 192.168.1.1 ... 192.168.1.123**  
**#Hosts: 128-2 = 126**



Subnet address: 192.168.1.128/25  
Network address: 192.168.1.128  
Default broadcast: 192.168.1.255  
**Range of addresses: 192.168.1.129 ... 192.168.1.255**  
**#Hosts: 128-2 = 126**



# Router, Routing table --> ROUTING

- Routing is the action needed to make it possible for hosts located in **different subnets** to communicate
- A Router is a network device that routes traffic.
- A routing table is a collection of routes that define how to reach a given network/subnet.
- A basic router routes traffic based on the destination address
  - Other factors can be considered too (policy routing)

# Example

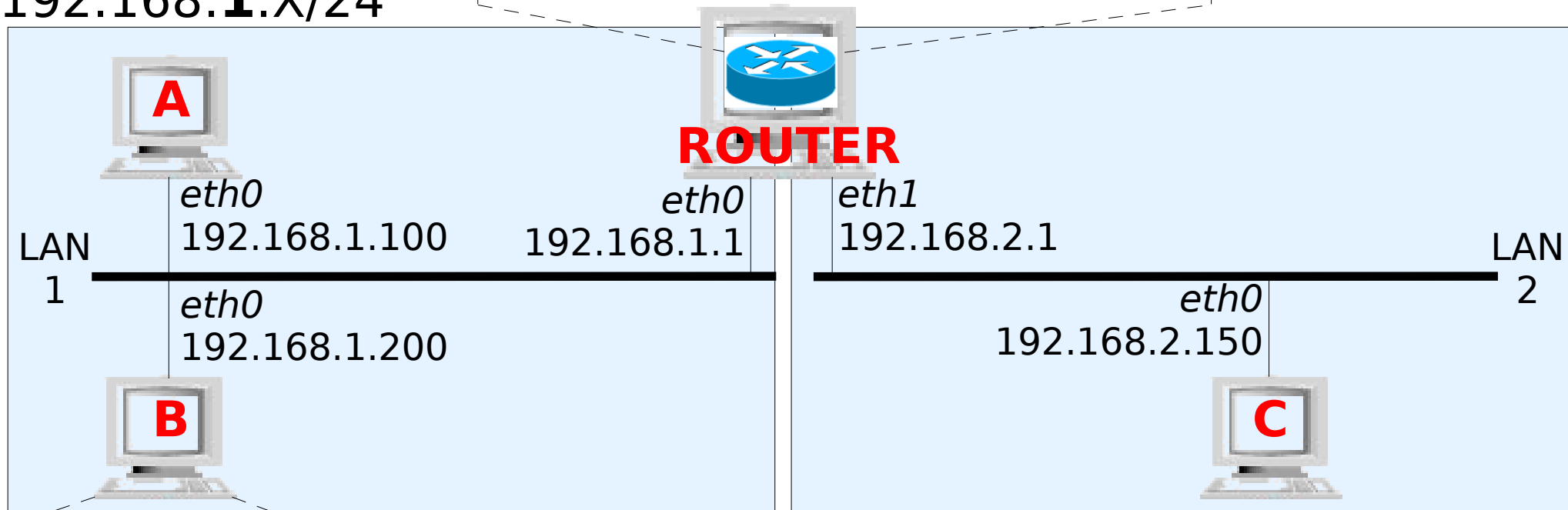
Routing table

Destination	Next-hop	Interface
192.168.1.0/24	<direct>	<i>eth0</i>
192.168.2.0/24	<direct>	<i>eth1</i>
192.168.1.1/32	<local>	</lo>
192.168.2.1/32	<local>	</lo>

**NOTE:**

Depending on the router type, forwarding may need to be explicitly enabled

**192.168.1.X/24**



**192.168.2.X/24**

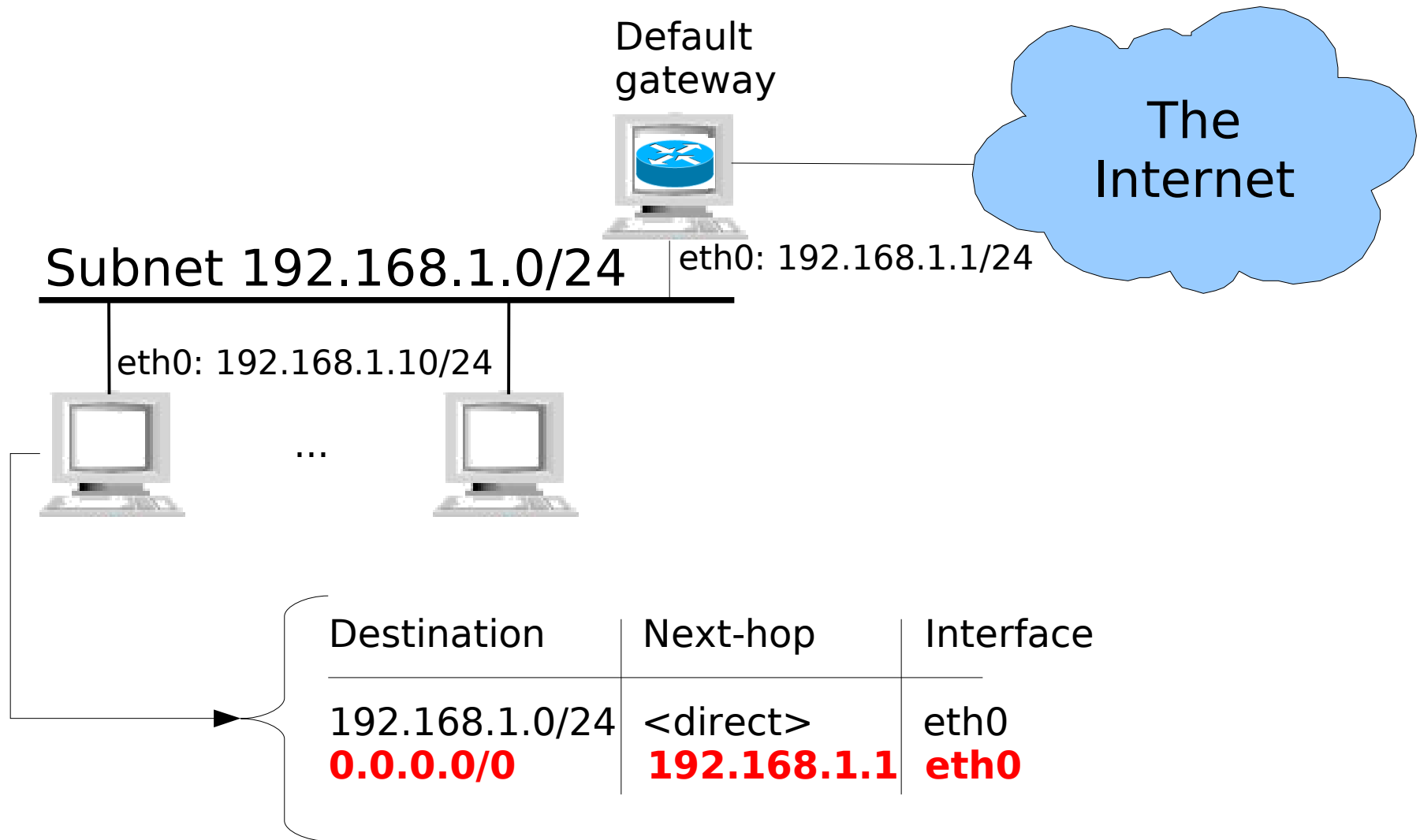
Destination	Next-hop	Interface
192.168.1.0/24	<direct>	<i>eth0</i>
192.168.1.200/32	<local>	</lo>

Routing table

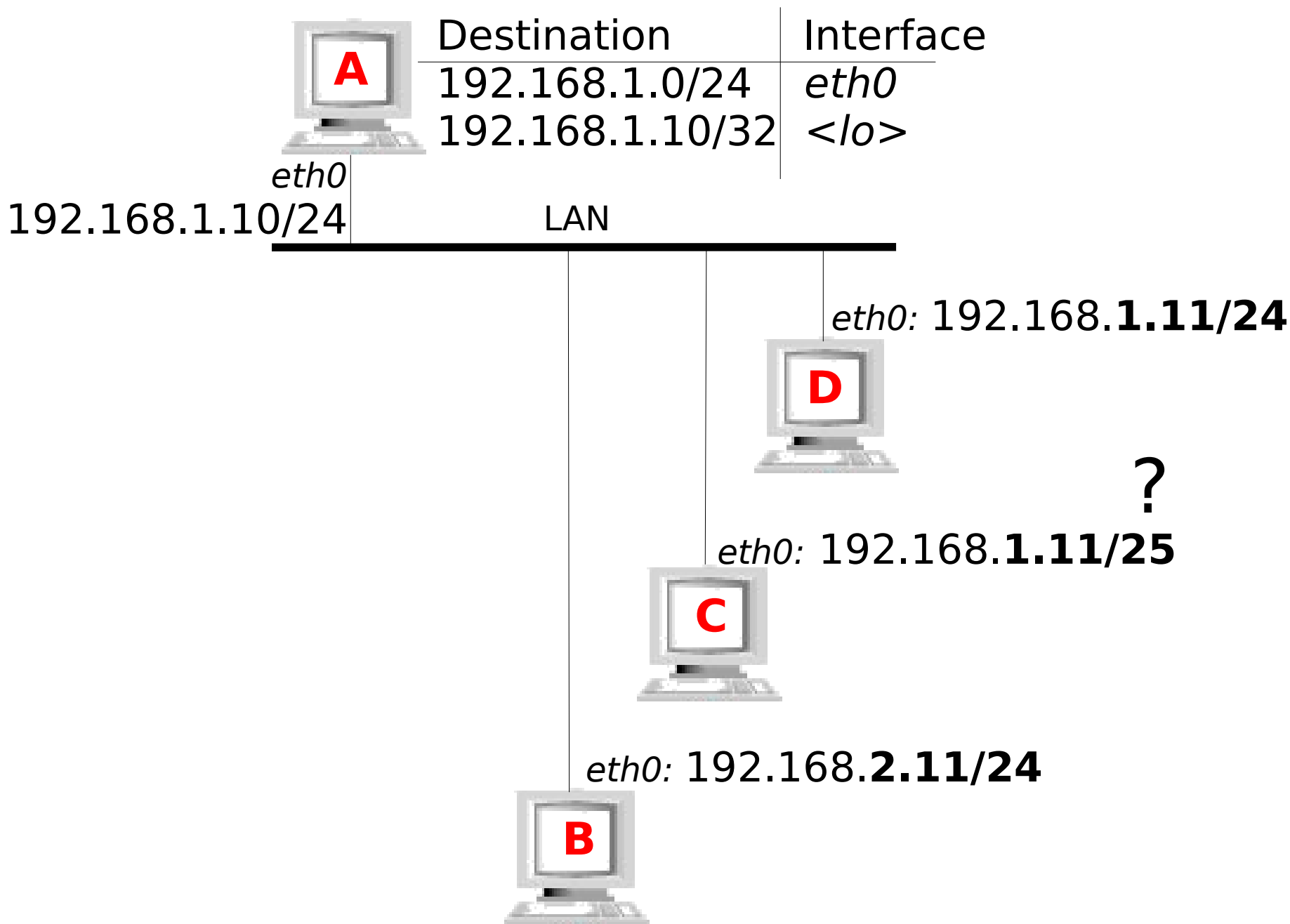
# Default routes

- A default route is used when there is no explicit route toward a given destination address.
- You can configure more than one default route (however it is not a common scenario)
- While hosts always use default routes, routers do not always need one.

# Example of default route

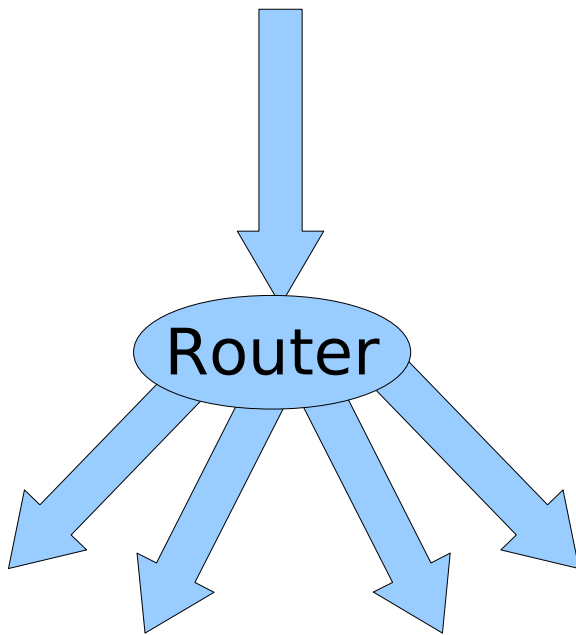


# Exercise

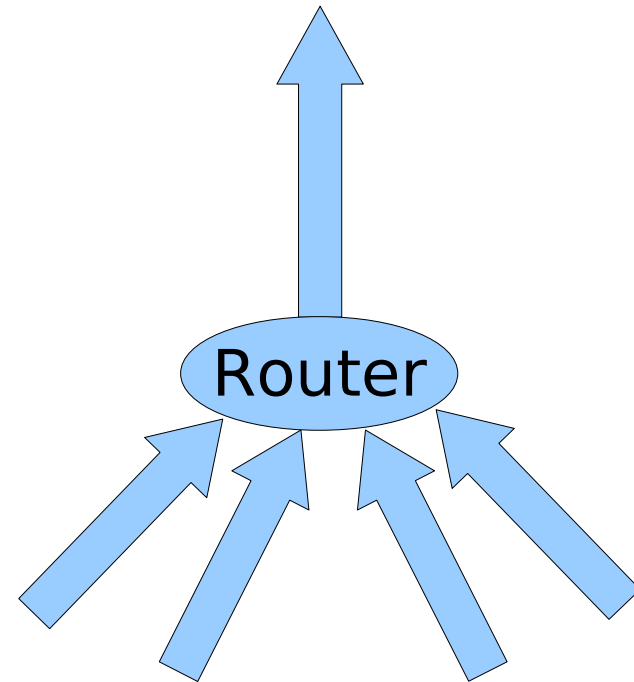


# Summarization

- Simplifies routing tables ...
  - ... which allows routers to route faster

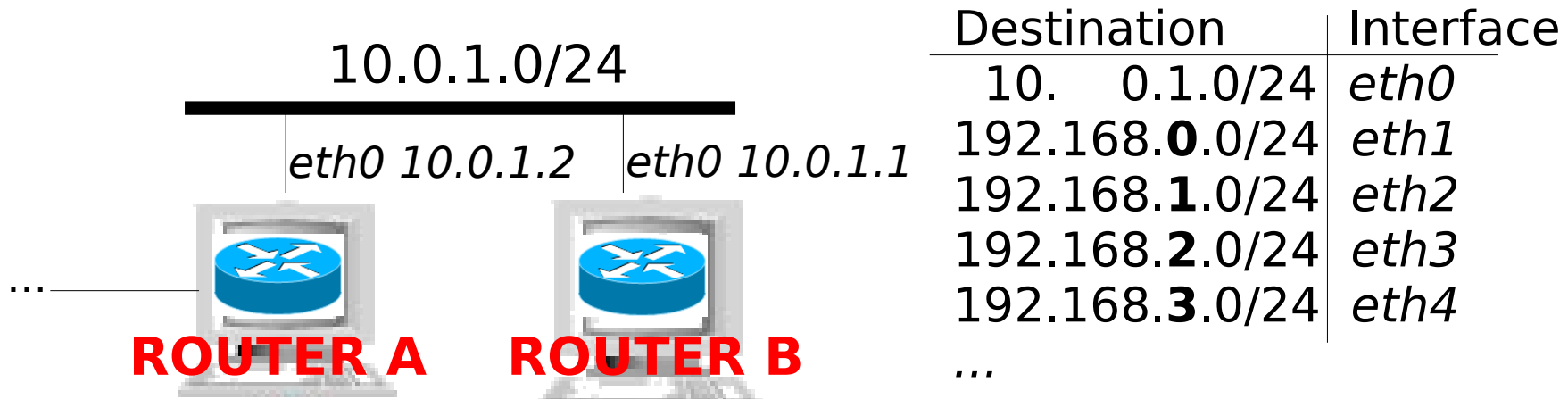


Subnetting

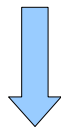


Summarization

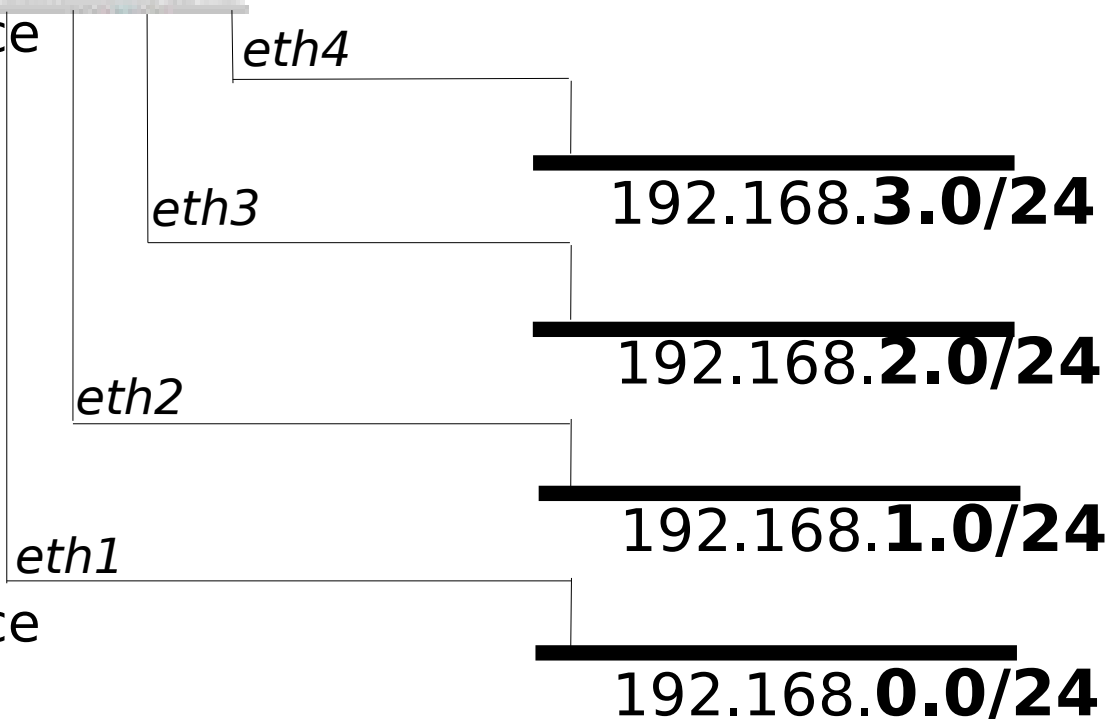
# Example of summarization



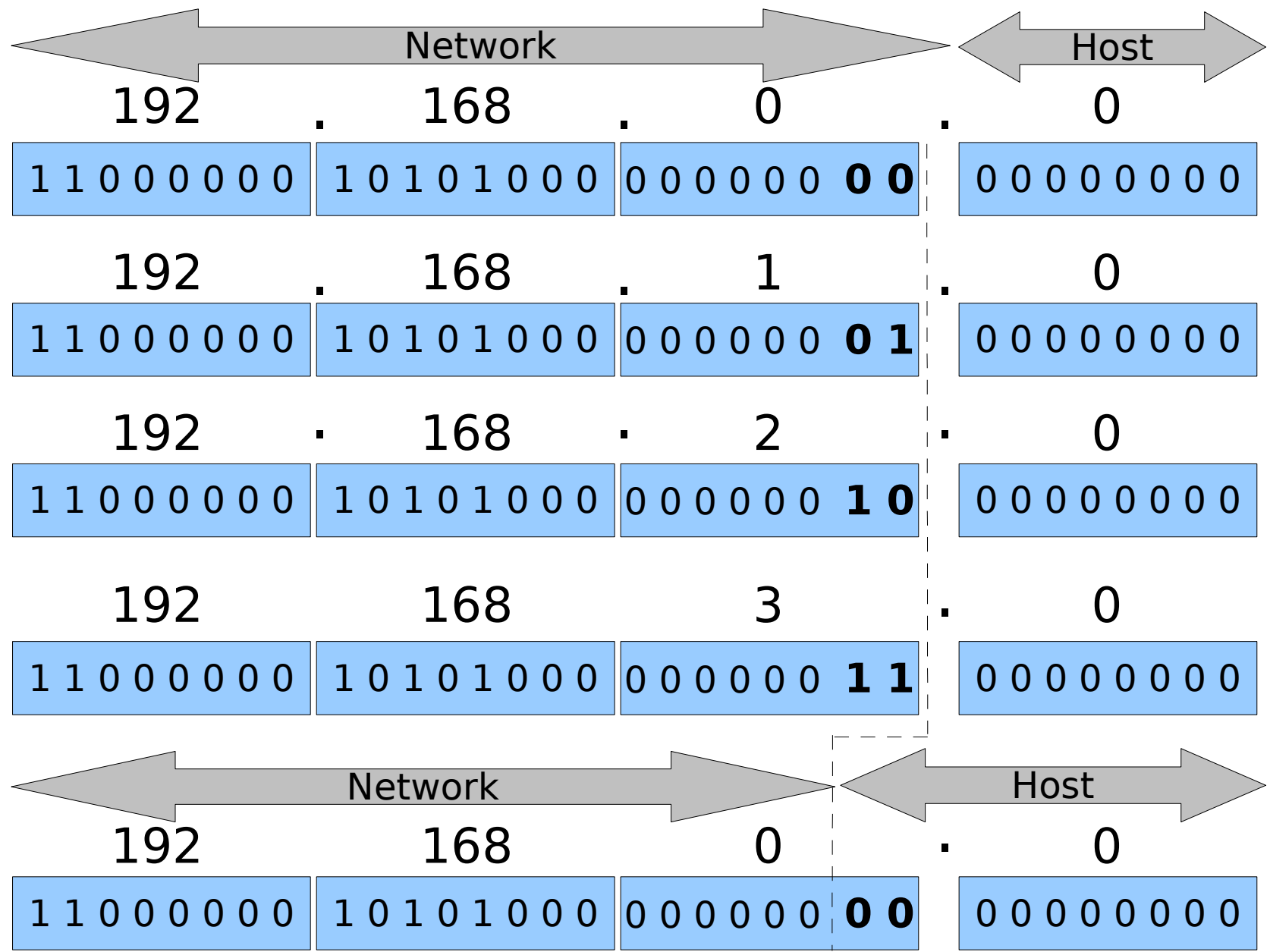
Destination	Next-hop	Interface
10. 0.1.0/24	<direct>	eth0
192.168.0.0/24	10.0.1.1	eth0
192.168.1.0/24	10.0.1.1	eth0
192.168.2.0/24	10.0.1.1	eth0
192.168.3.0/24	10.0.1.1	eth0



Destination	Next-hop	Interface
10. 0.1.0/24	<direct>	eth0
<b>192.168.0.0/22</b>	10.0.1.1	eth0



# Summarization: playing with the bits



# Routing

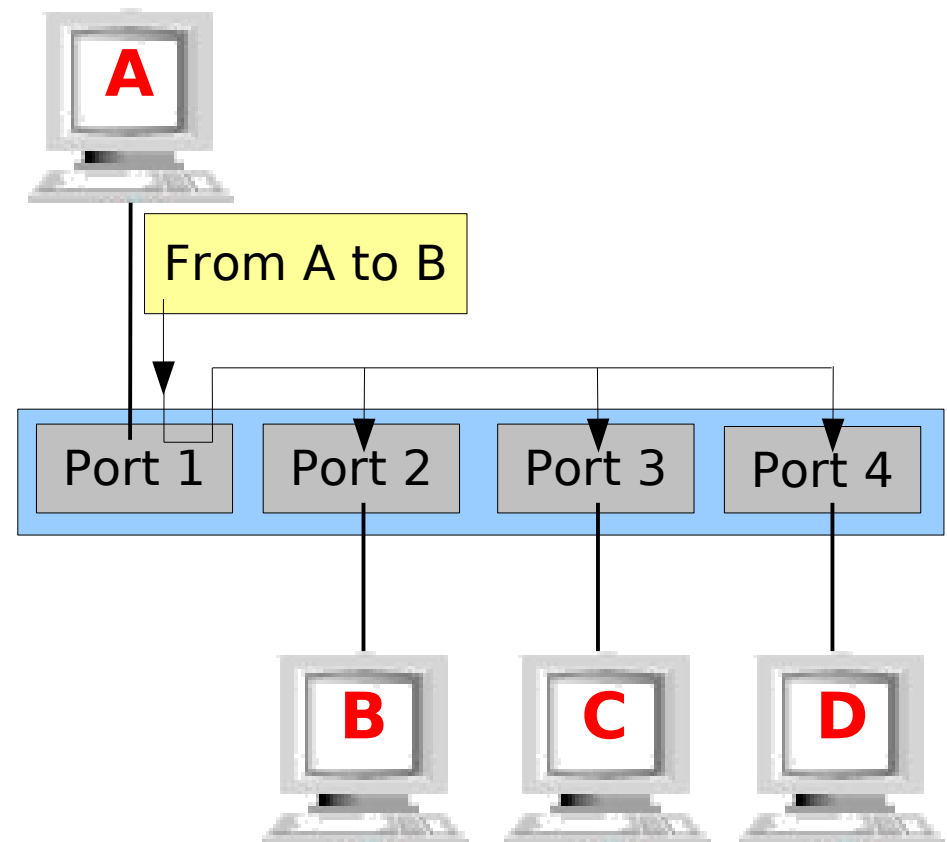
- Static VS Dynamic
- On a small network, static routing is sufficient.
  - Dynamic routing may be used anyway to provide some kind of high availability (to handle router failures)
- On bigger networks the use of dynamic routing becomes necessary
  - Depending on the size of the network and your exact requirements, different protocols are available for the job.

# Routing VS Switching

- They operate at different layers, but:
  - L3 Routing tables VS L2 Forwarding databases
  - L3 Routing protocols VS L2 Spanning Tree
  - The hierarchical configuration model (access, distribution, core) applies to both.
- Switching is complex too!
  - Do not associate switching with the low-end 4-port switches!
  - Spanning Tree Protocol/s
  - Virtual LANS (VLANs)
  - ...

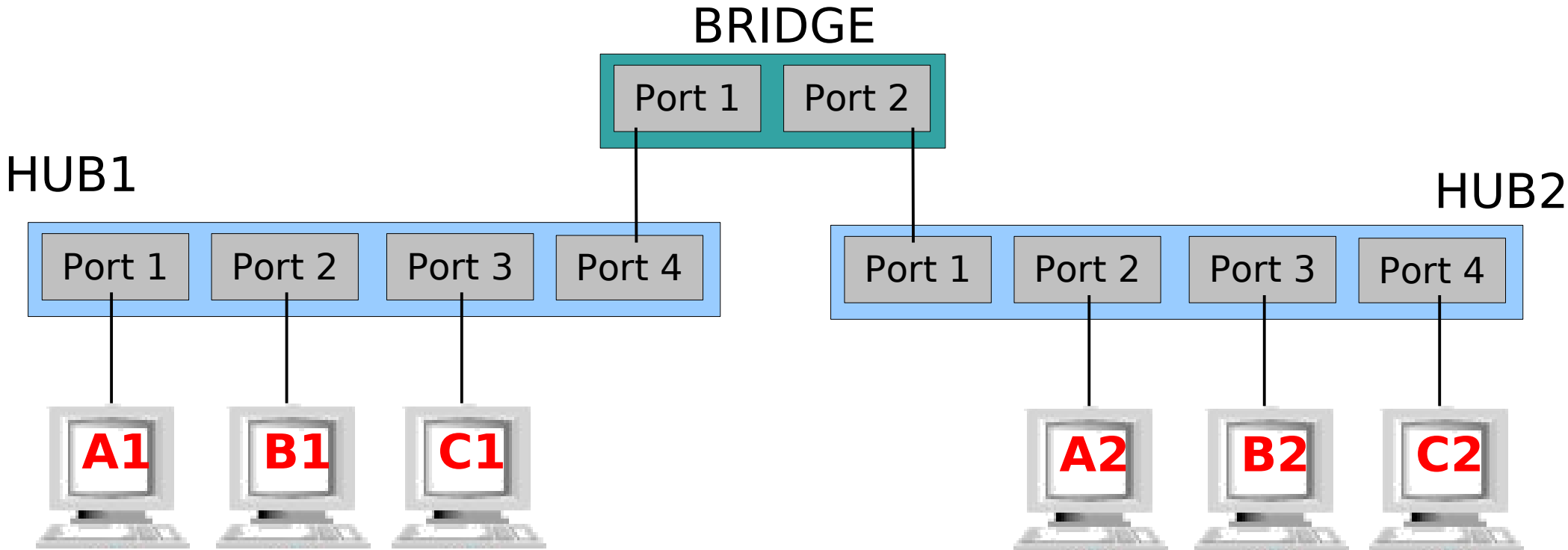
# HUB

- Cheap
- Low performance



Can A talk to B while C talks to D?

# Bridge



Case 1

From B1 to C1

Can A1 talk to A2 while C1 talks to D1?

Case 2

From B1 to C2

Can A1 talk to B1 while A2 talks to B2?

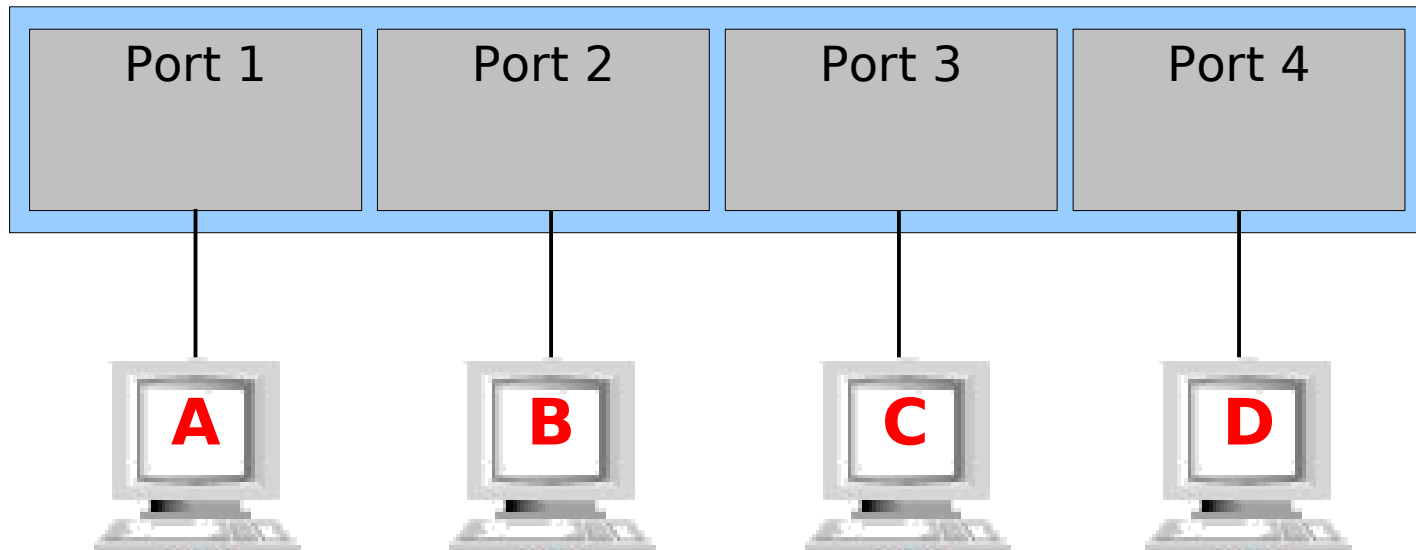
Case 3

From B2 to B1

Can A1 talk to B1 while C1 talks to C2?

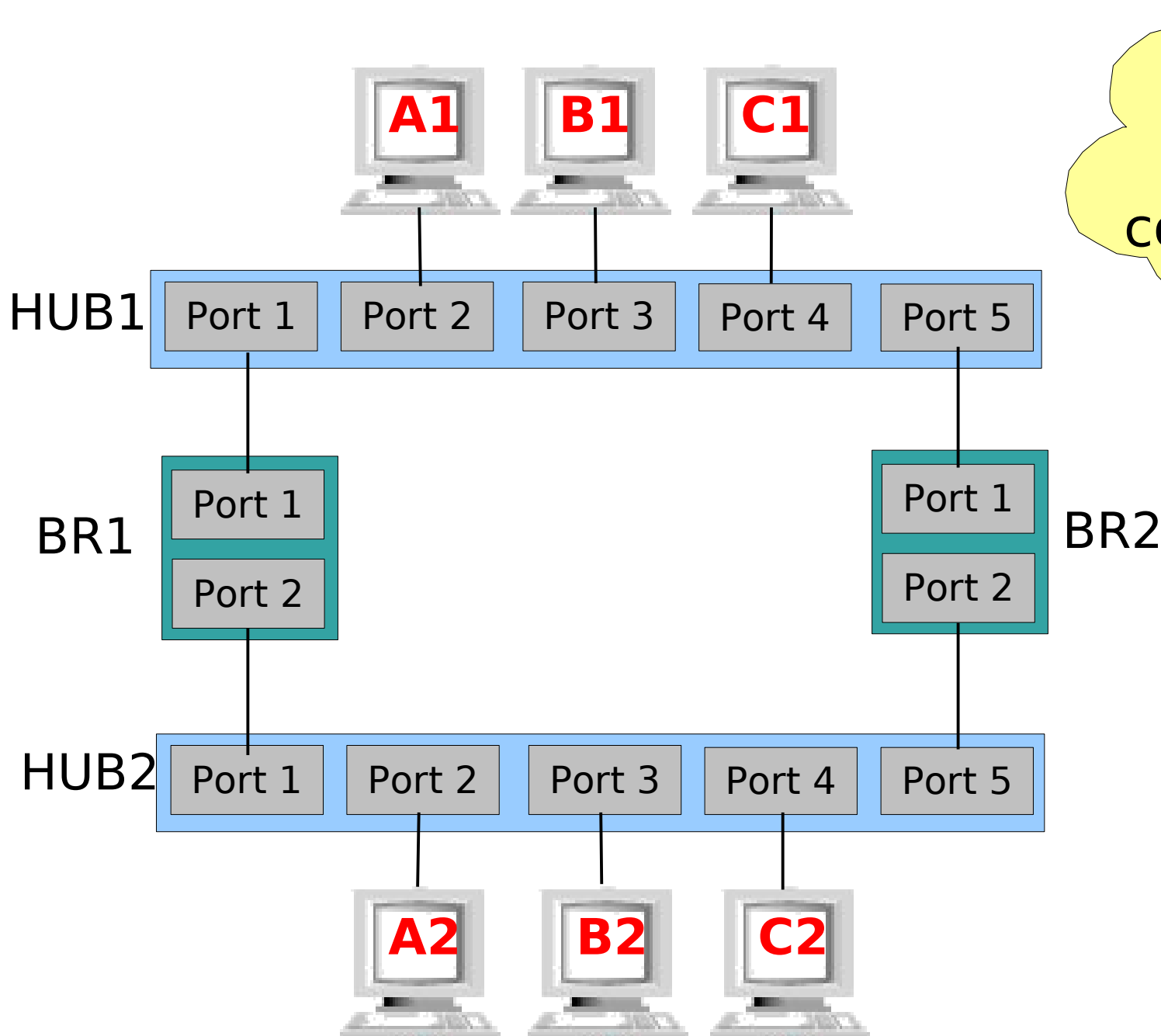
# SWITCH

(sort of multi-port bridge)



Can A talk to B while C talks to D?

# What if a bridge/switch fails?

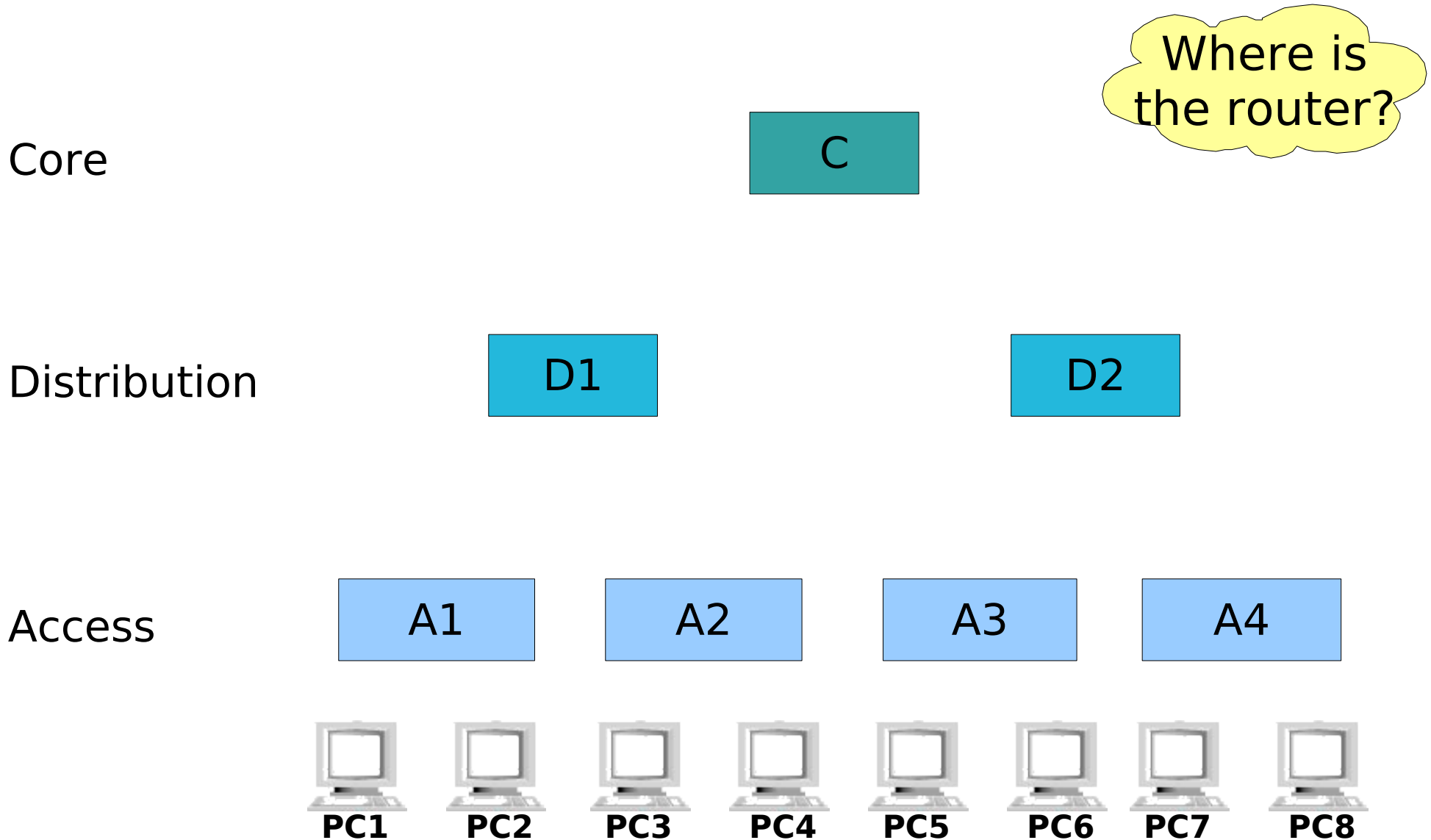


How good is this configuration?

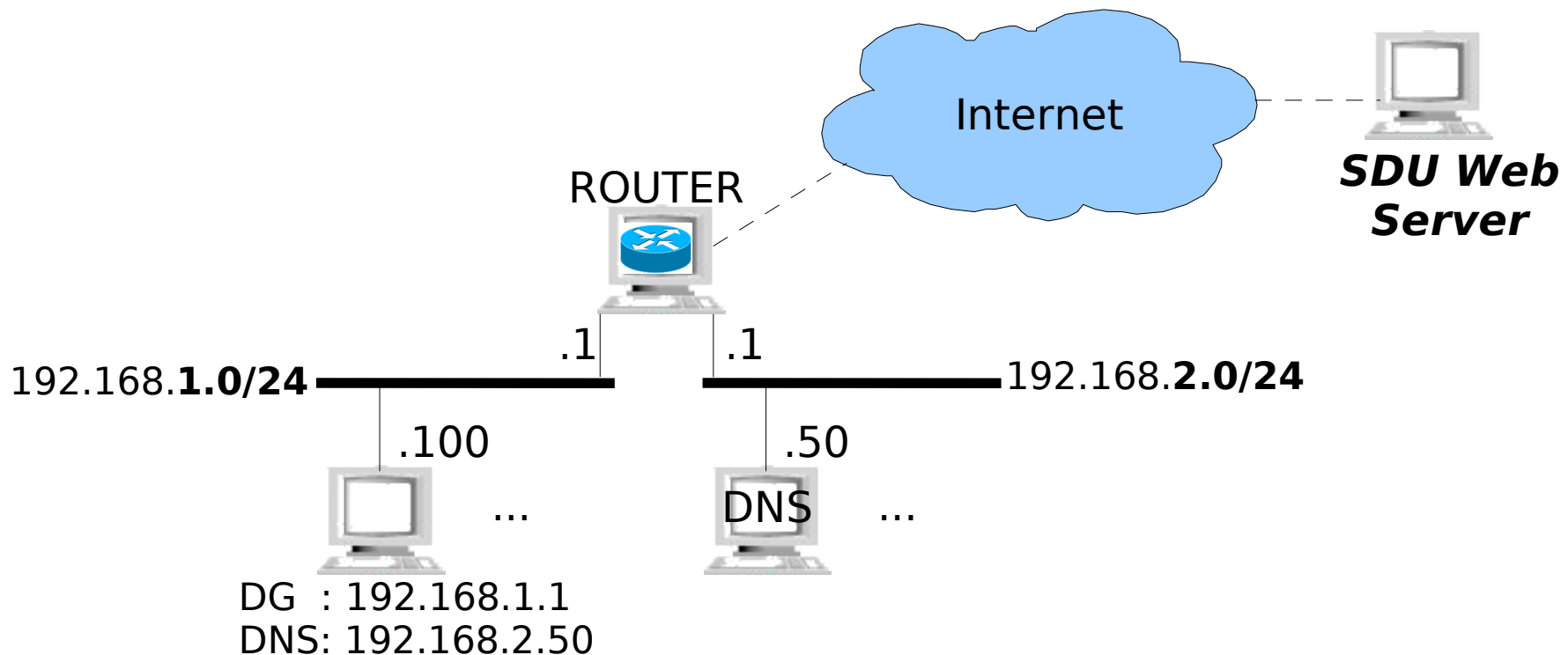
# VERY IMPORTANT

- **Never** introduce any loop in a L2 network unless:
  - You know (very well) what you are doing
  - You rely on a protocol like the Spanning Tree to disable the redundant links

# Let's play a little ...



# Domain Name System (DNS)



***FIREFOX: --> <http://sdu.ictp.it>***

Let's see how the host  
accesses the WEB server ...  
step by step ...

Any questions?

# Copyright Creative Commons

- This presentation is released under the Creative Common License:
  - Attribution, Noncommercial, Share Alike 2.5
  - (<http://creativecommons.org/licenses/by-nc-sa/2.5/>)
- Attribution
  - You must attribute the work in the manner specified by the author or licensor.
- Noncommercial.
  - You may not use this work for commercial purposes.
- Share Alike.
  - If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.