Networking Basics

Christian Benvenuti
(christian.benvenuti@libero.it)
[http://benve.info]

ICTP-ITU-URSI School on Wireless Networking for Scientific Applications in Developing Countries
(5-23 February 2007, Trieste, Italy)

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?

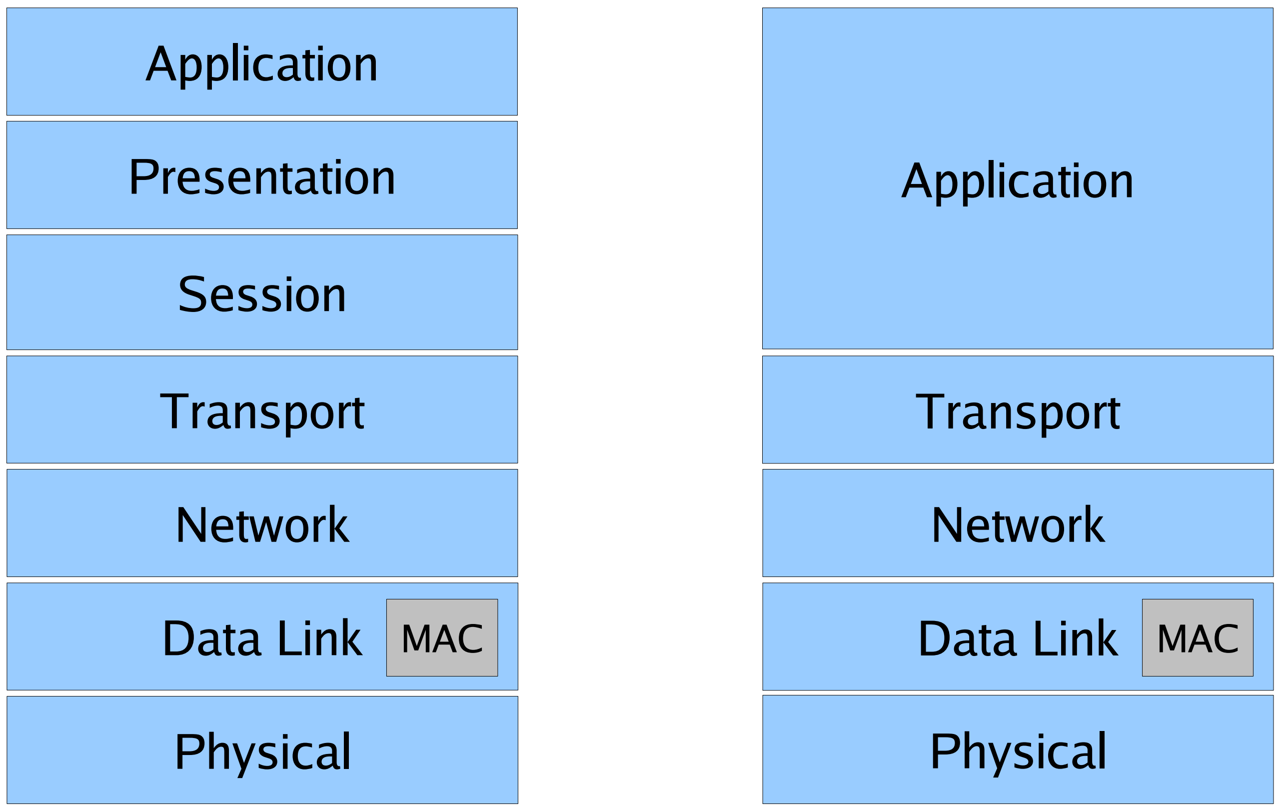
Agenda

- 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 network!

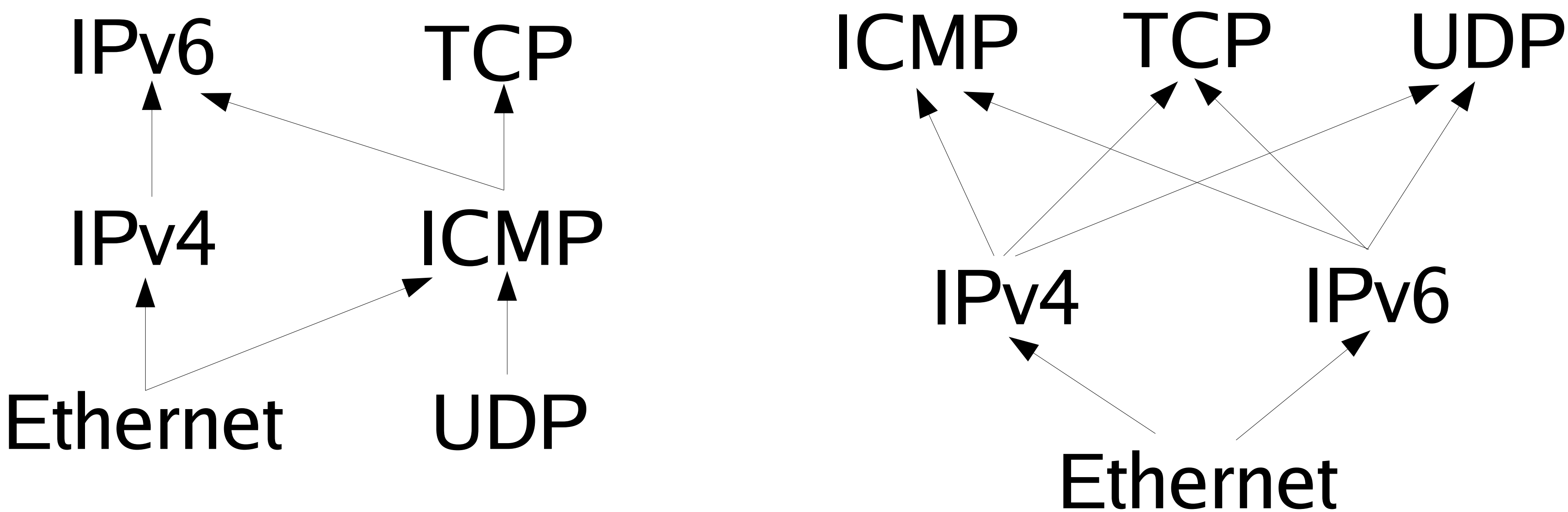
The most common slide you'll find in a “Networking Basics” presentation



ISO-OSI

TCP/IP

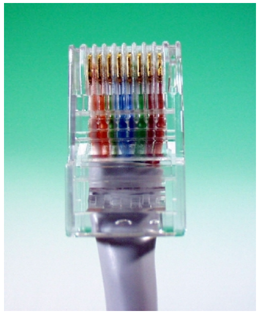
What's wrong here?



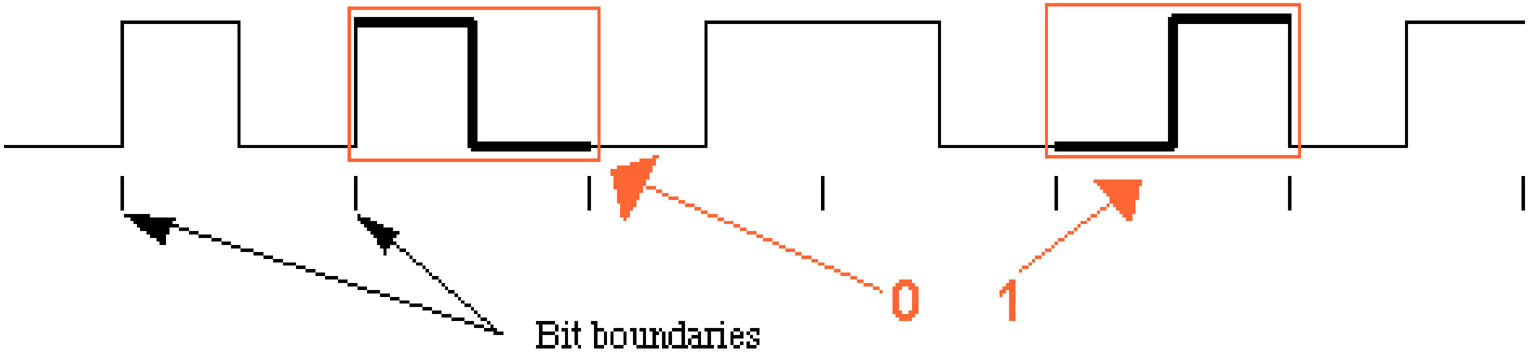
Why a network stack?

- 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, ...)



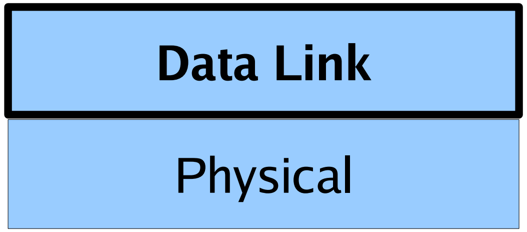
- Timing of bits (Sync vs Async)

Physical

TCP/IP

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

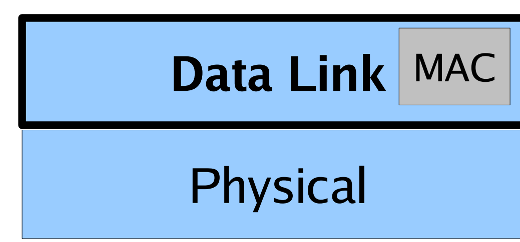
- Framing
- Error Detection/Correction (CRC)
- Simplex/Half/Full duplex
- ...



TCP/IP

The ***Data-Link*** Layer, (2/2)
The Medium Access Control (MAC)

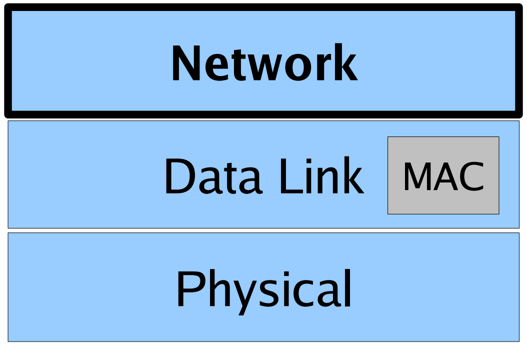
- Collision free
  - Static (TDMA, ...)
  - Dynamic (Token, Bitmap, ...)
- Not Collision Free
  - CSMA [p-persistent] /CD
- Wireless mediums use different algorithms. (ask the wireless gurus in the room ...)



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
  - ...

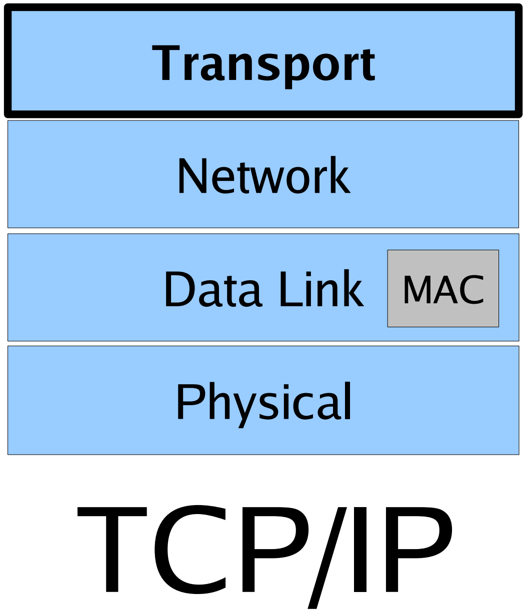


TCP/IP

The ***Transport*** Layer

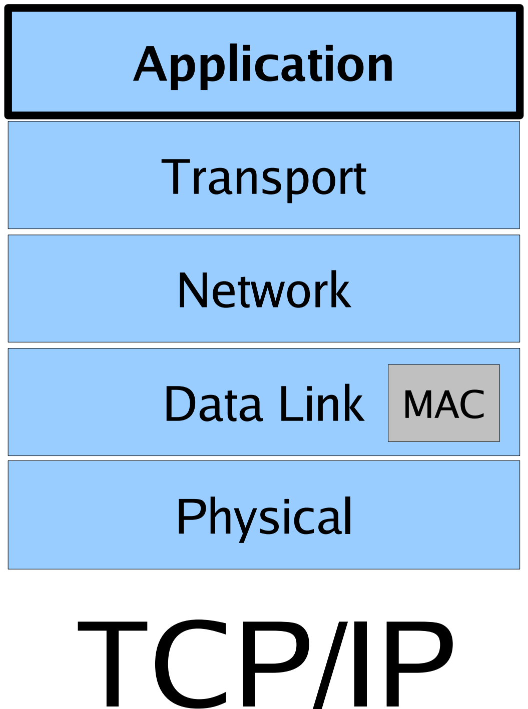
- (Logical) Connection multiplexing
- Flow control
- Error detection (data corruption)
- ...

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

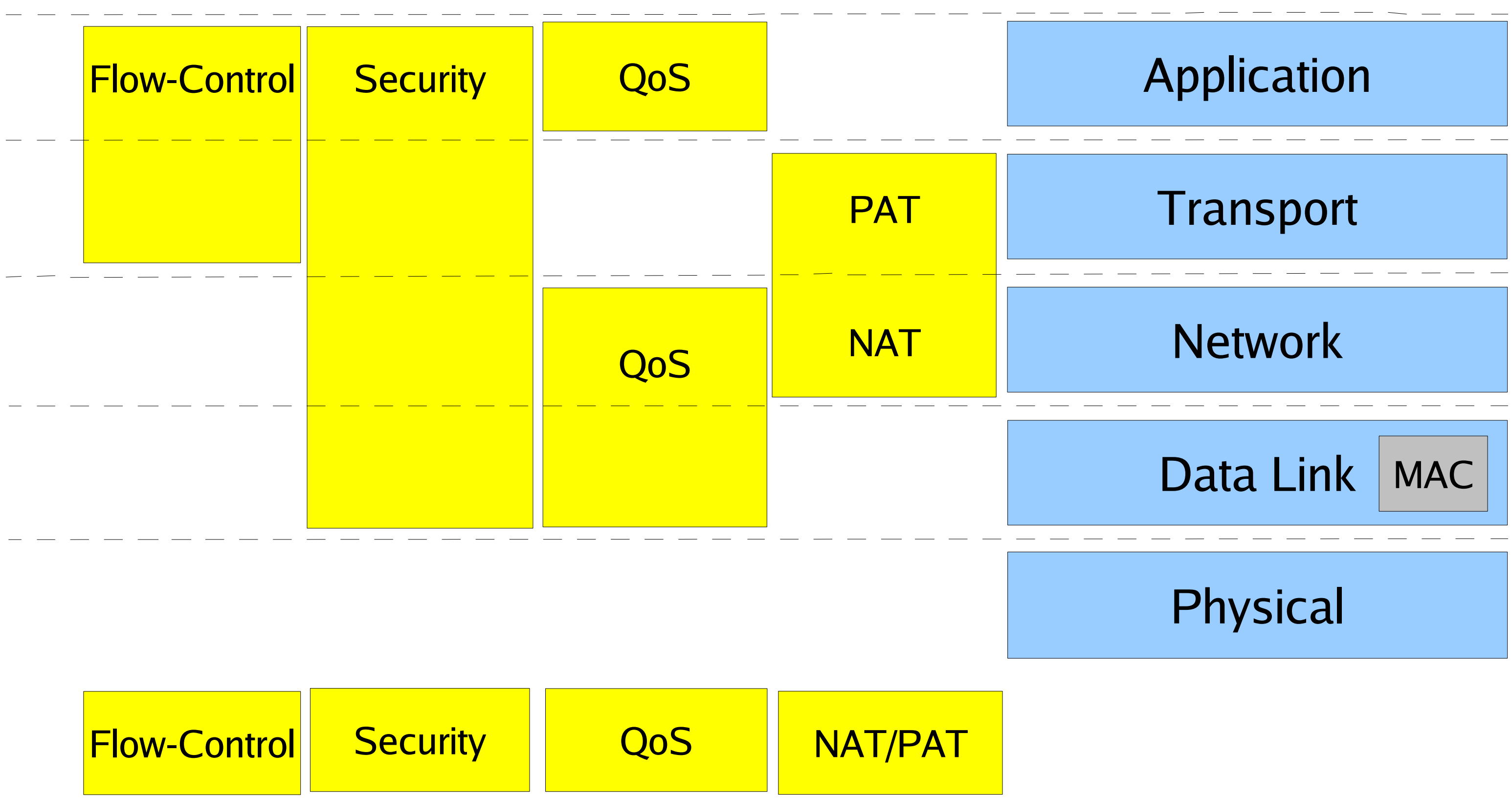


The *Application* Layer

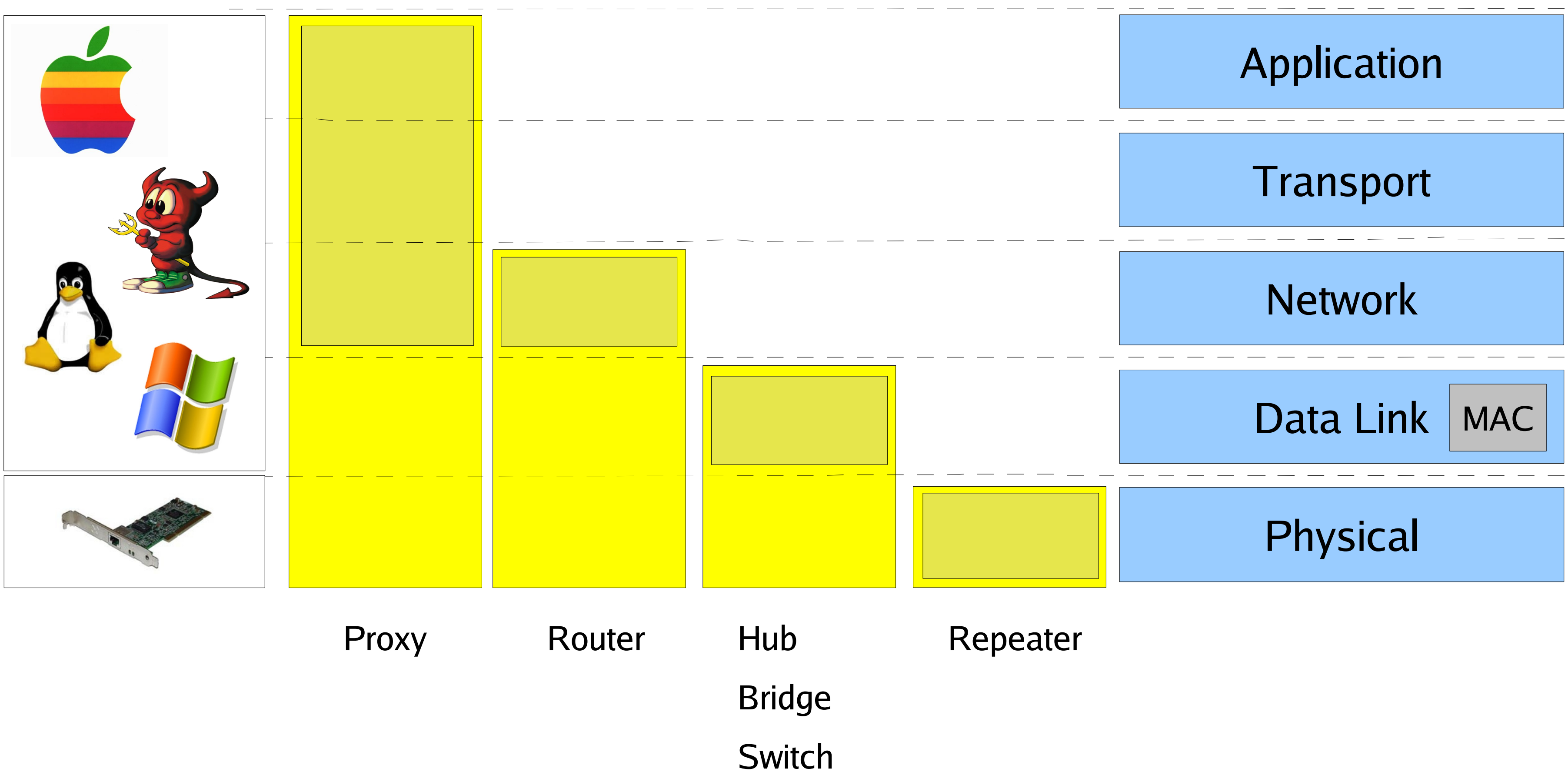
- This is the layer where the applications users interface with are located.



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



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

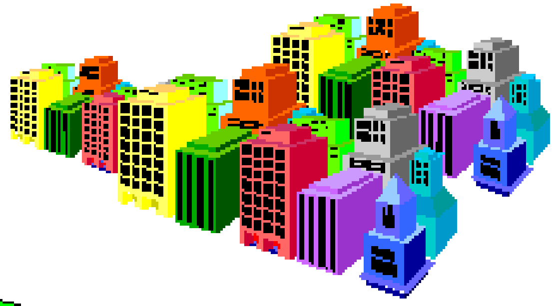


What about Firewalls, Intrusion Detection Systems, etc ?

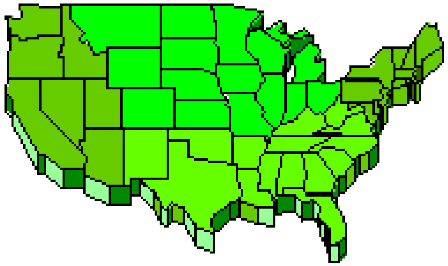
Network **size** and topology

- Size
  - Local Area Network (LAN)

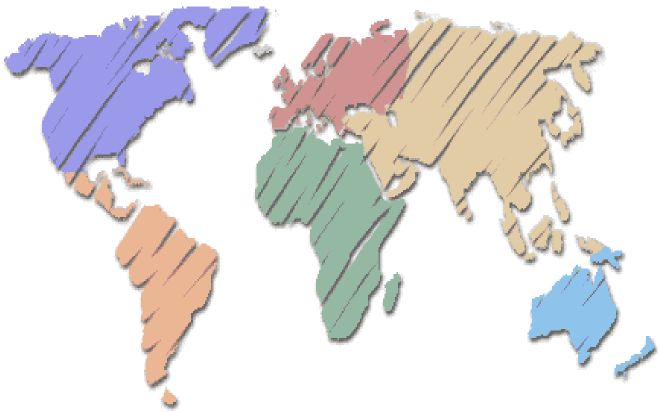
  - Metropolitan Area Network (MAN)



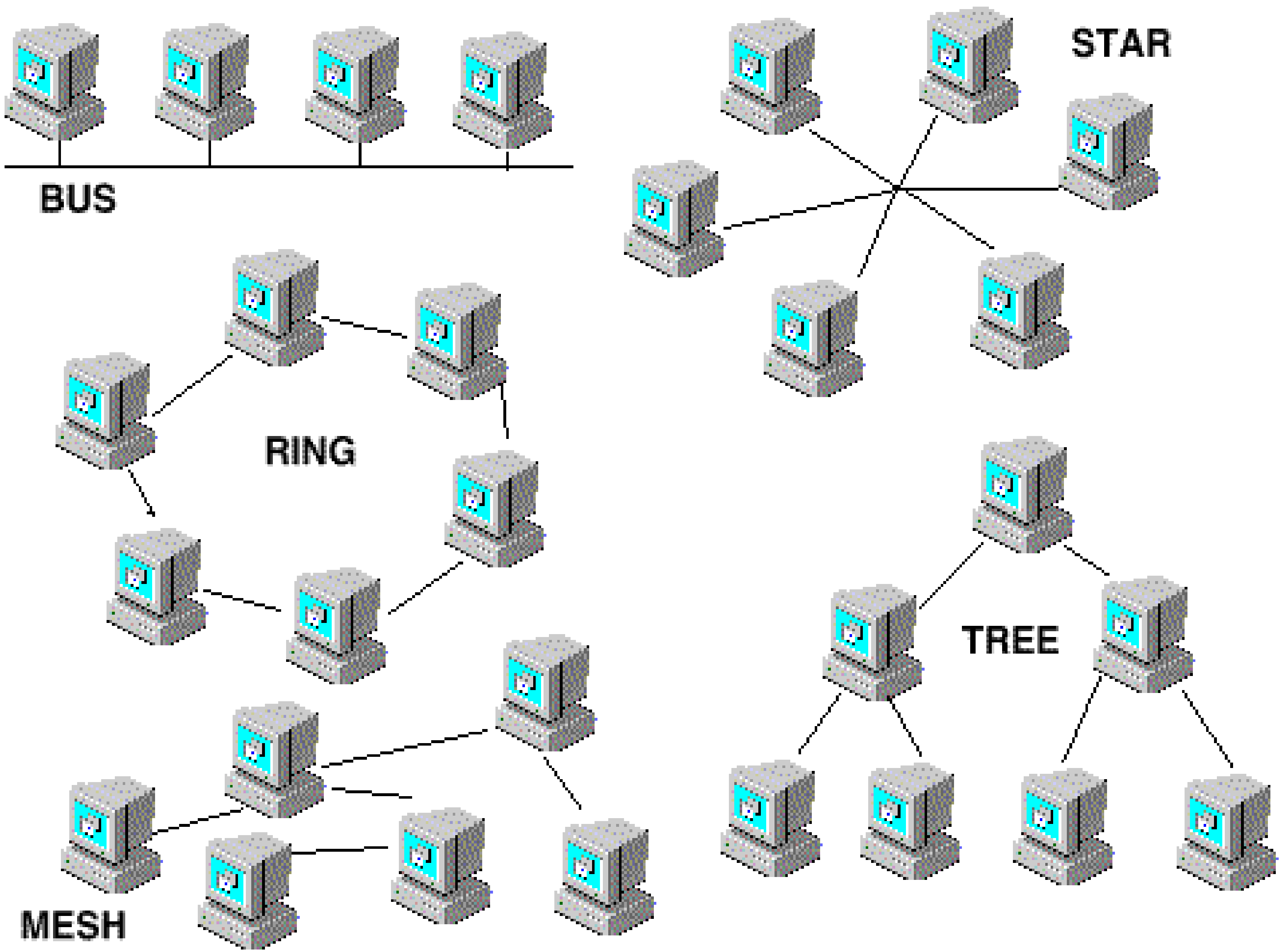
  - Wide Area Network (WAN)



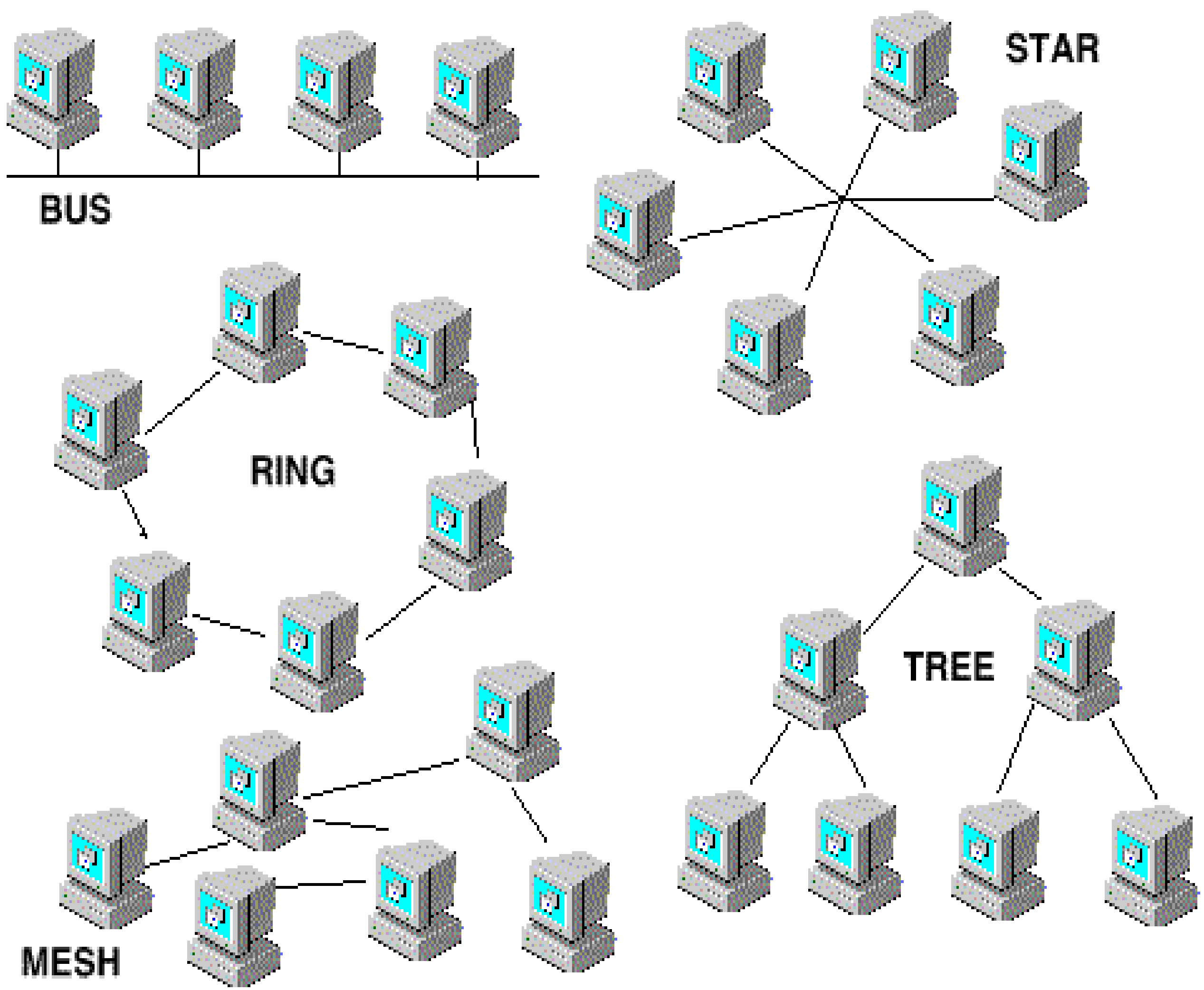
  - The Internet



Network size and **topology**



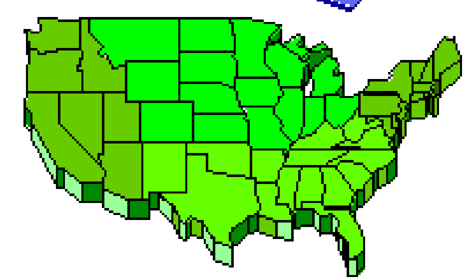
What combinations do you think make more sense?



LAN

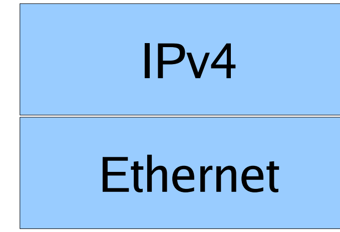
MAN

WAN



Internet

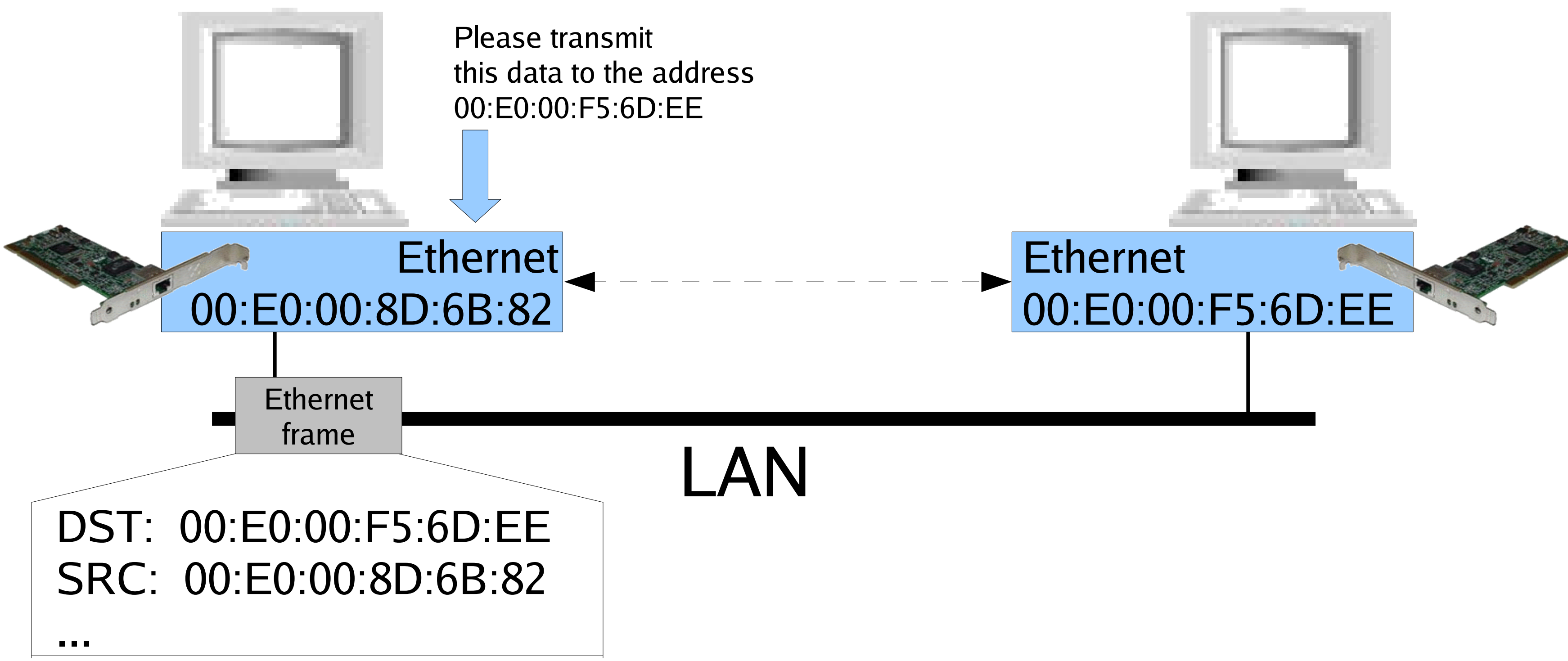
The L2/L3 de-facto standards

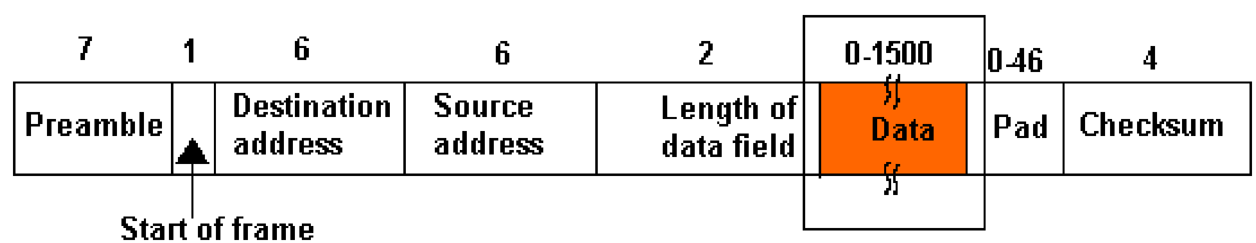


- Wired LANs
  - Ethernet
    - Fast/Giga/10-Giga
- Wireless LANs
  - 802.11b, … <this will be the focus of the next three weeks>
- Internet Protocol Version 4 (IPv4)
  - Its younger brother IPv6 is not yet as widely used as IPv4.

Network (IPv4)

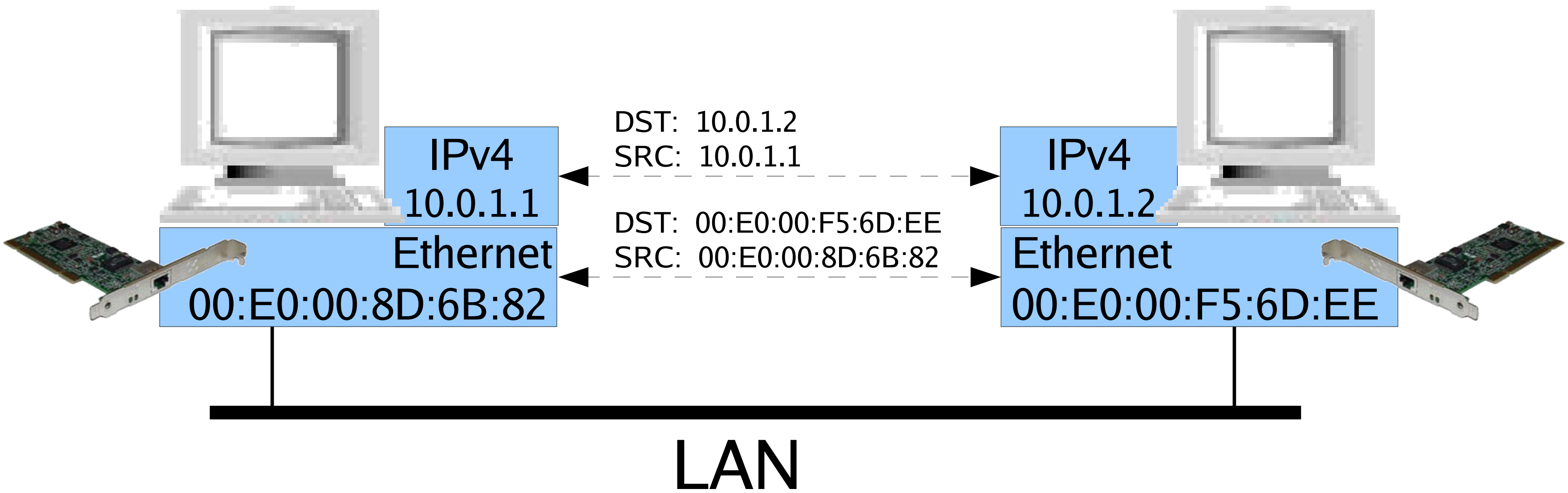
Data-Link (Ethernet)





Structure of an Ethernet frame

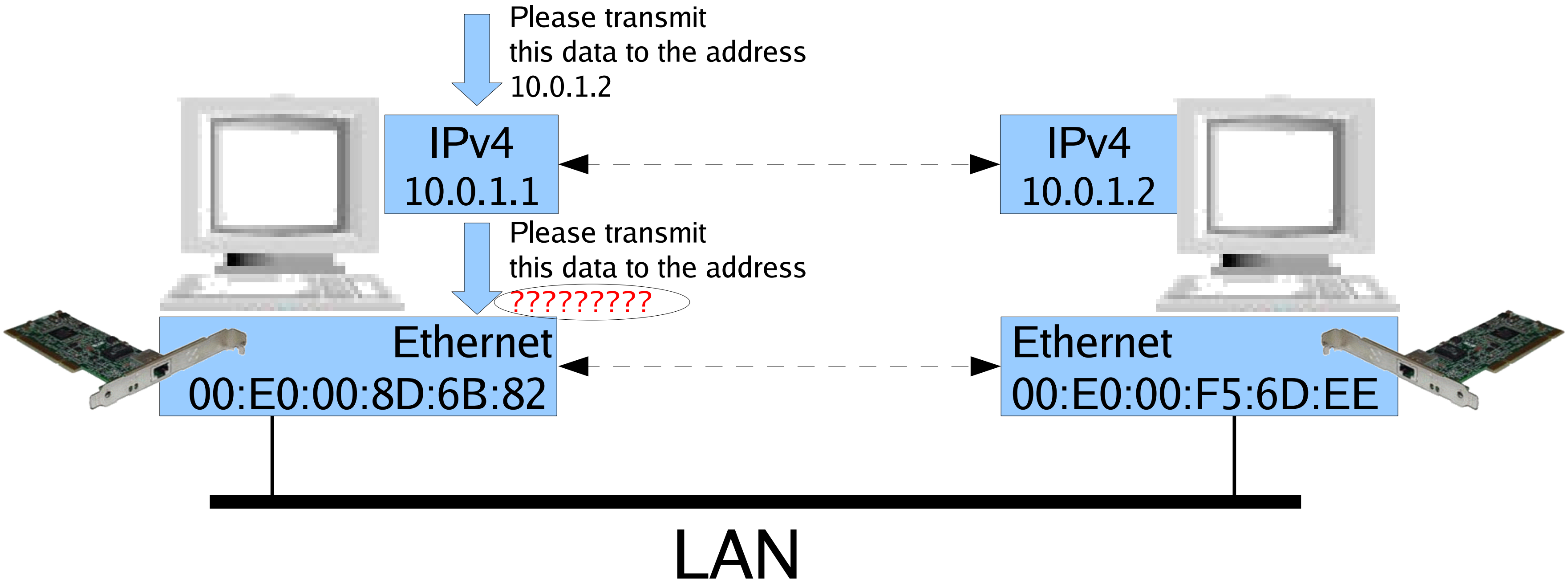
Network (Ipv4) ⟷ Data-Link (Ethernet)



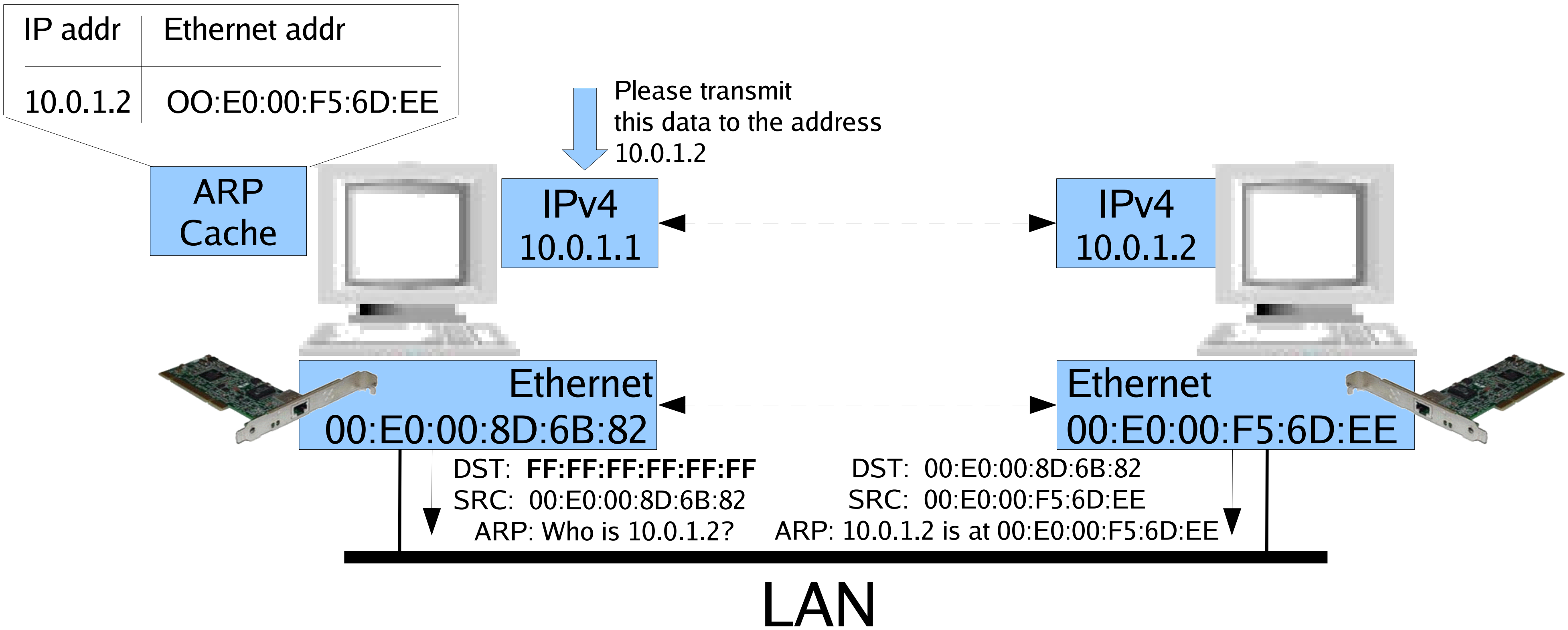
0	4	8	16	19	31
VERS	LEN	Type of Service	Total Length		
Identification			Flags	Fragment Offset	
TTL		Protocol	Header checksum		
source IP address					
destination IP address					
Options			..|.. padding		
data					

7	1	6	6	2	0-1500	0-46	4
Preamble	Start of frame	Destination address	Source address	Length of data field	Data	Pad	Checksum

Network (IPv4) ⟷ Data-Link (Ethernet)



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*, L2:*Datagram* L4:*Segment*, L5:*Message*
  - Generic: *Packet*

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

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

IPv4

Basic hdr: 20 Bytes

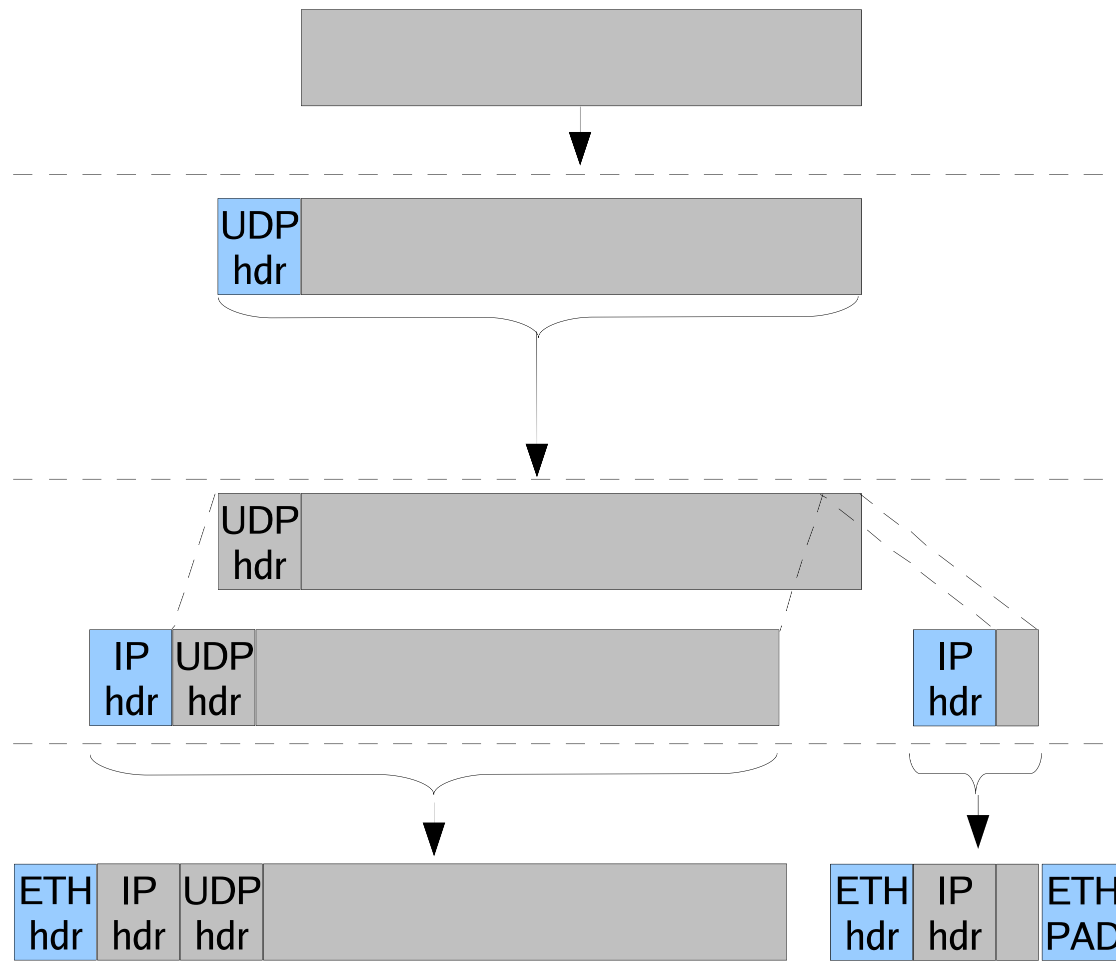
Min: /

Max: 64 KBytes [PMTU ...]

Ethernet

Min: 64 Bytes

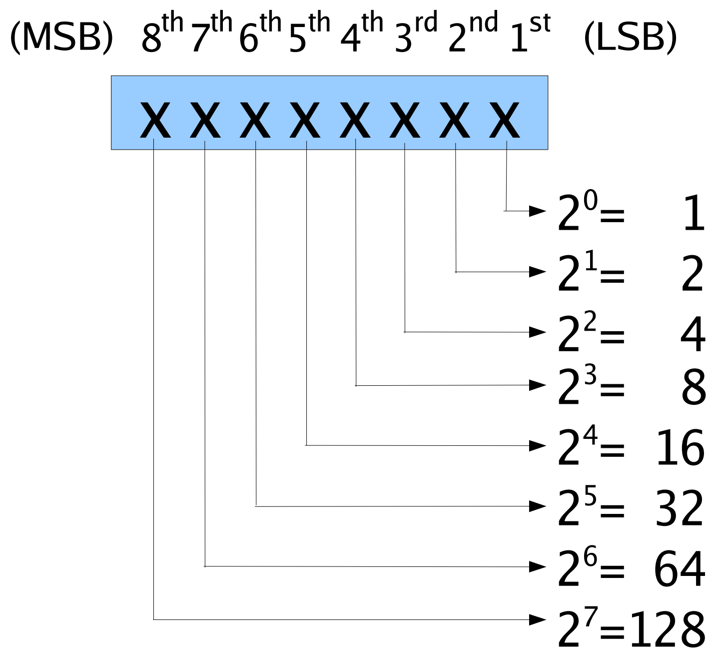
Max: 1514/18 Bytes (configurable)



More on the **Network** Layer

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

Binary numbers



2^8= 256
2^9= 512
2^{10}= 1024
2^{11}= 2048
2^{12}= 4096
2^{13}= 8192
2^{14}=16384
2^{15}=32768
2^{16}=65536

(A) Commonly used with broadcasts

(B) Commonly used with netmasks

Binary	Decimal	Hex
00000000	0	0x00
00000001	1	0x01
00000010	2	0x02
00000100	4	0x04
00001000	8	0x08
00010000	16	0x10
00100000	32	0x20
01000000	64	0x40
10000000	128	0x80

(A)

Binary	Decimal	Hex
00000000	0	0x00
00000001	1	0x01
00000011	3	0x03
00000111	7	0x07
00001111	15	0x0F
00011111	31	0x1F
00111111	63	0x3F
01111111	127	0x7F
11111111	255	0xFF

(B)

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

Binary operators

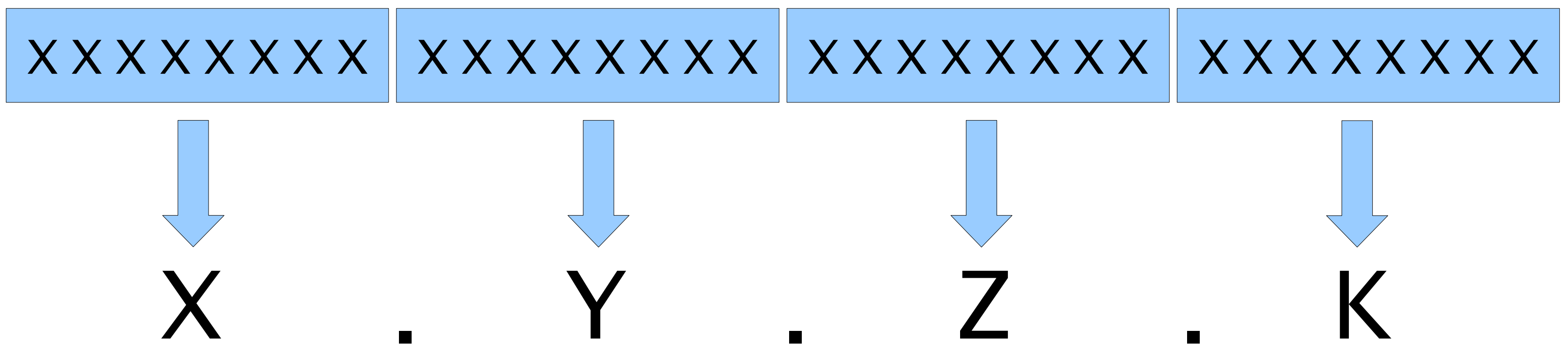
AND	0	1
0	0	0
1	0	1

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

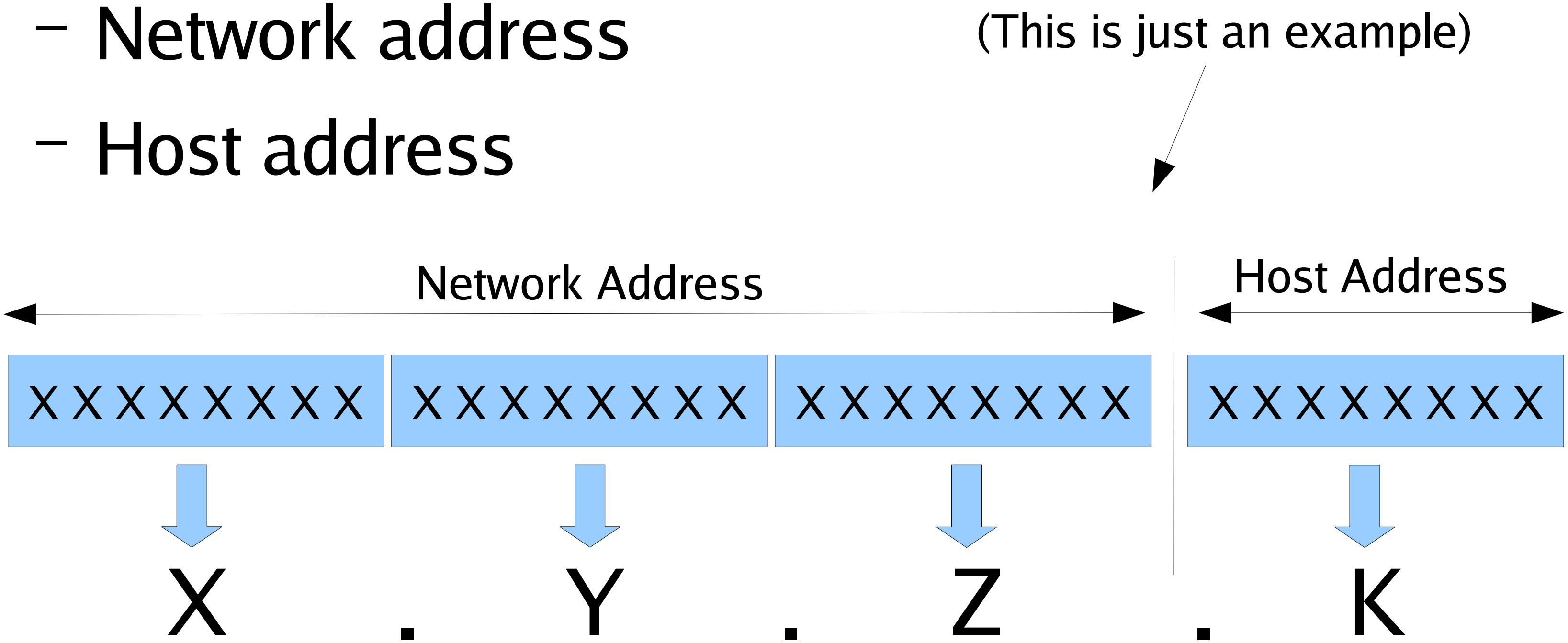
- 32 bits



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

Structure of an IPv4 address

- It consists of two components:
  - Network address
  - Host address

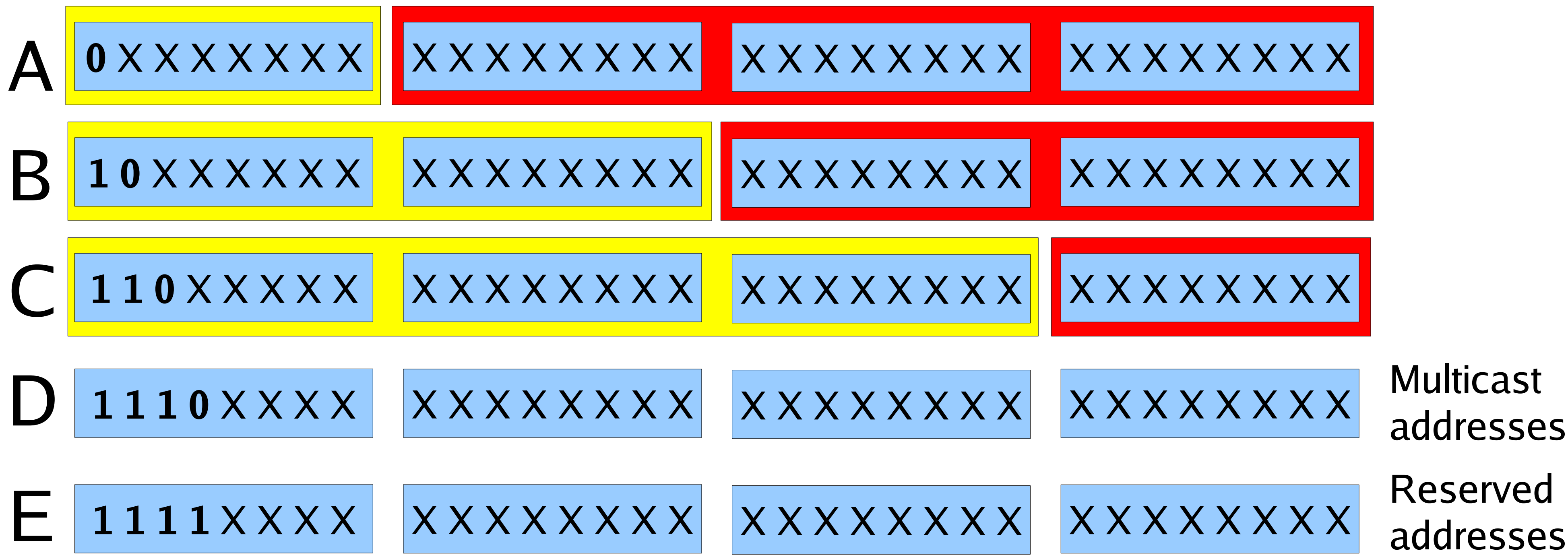


IP address: **X.Y.Z.K** netmask **255.255.255.0**
or
IP address: **X.Y.Z.K/24**

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

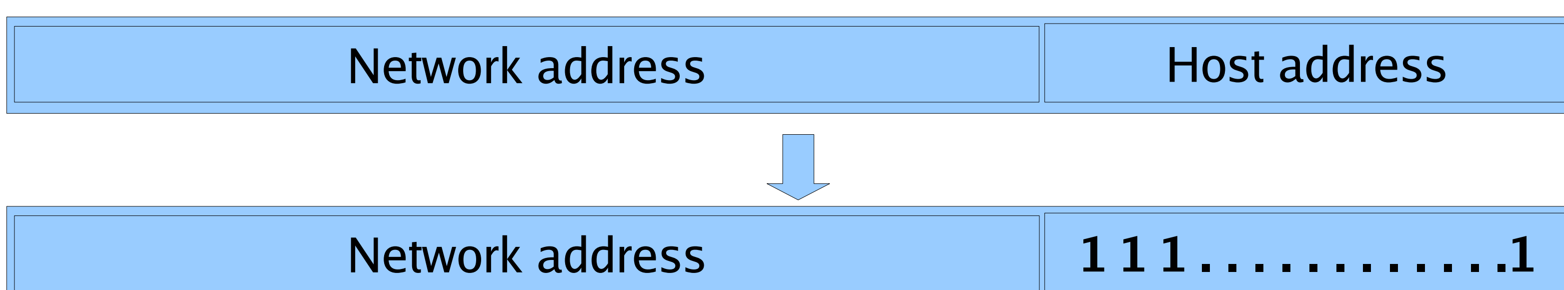
- **10**.0.0.0/8 1 x Class A 10 = **0** 0 0 0 1 0 1 0
- **172.16.0**.0/16 16 x Class B 172 = **1 0** 1 0 1 1 0 0
- **192.168**.0.0/16 256 x Class C 192 = **1 1** 0 0 0 0 0 0

- 127.0.0.0/8 loopback (valid only on the local host)

Unicast, Broadcast, Multicast

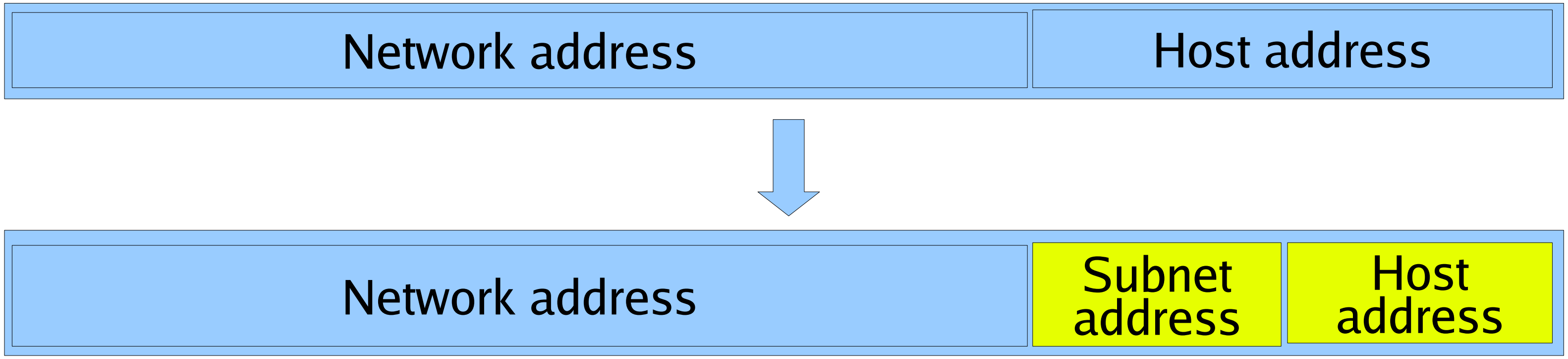
- The use of broadcast and multicast addresses is a convenient way for sending a packet to multiple recipients (link layer protocols use them too).
  - A broadcast message is addressed at all the hosts of a given network (or subnet).
    - Local VS Directed broadcasts
  - A multicast message is addressed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

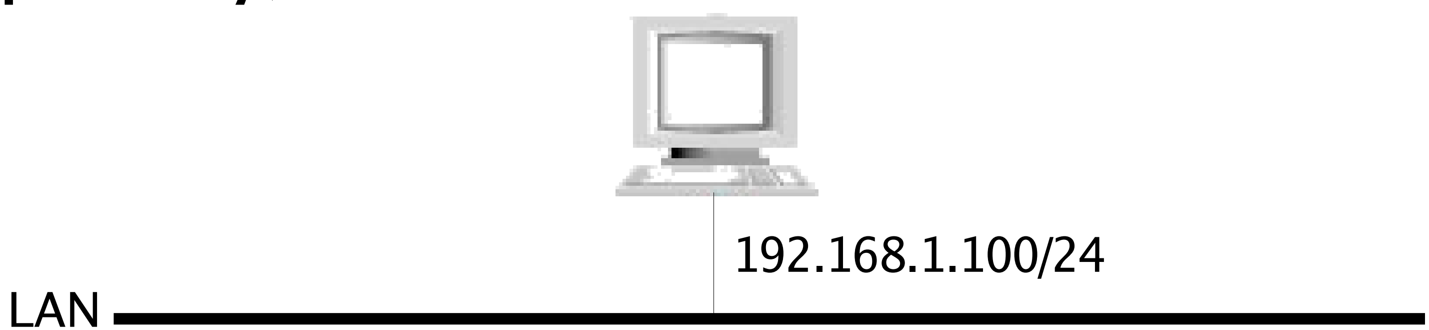
- The host address is split into two parts:
  - Subnet number
  - Host address



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

Example of subnetting (1/2)

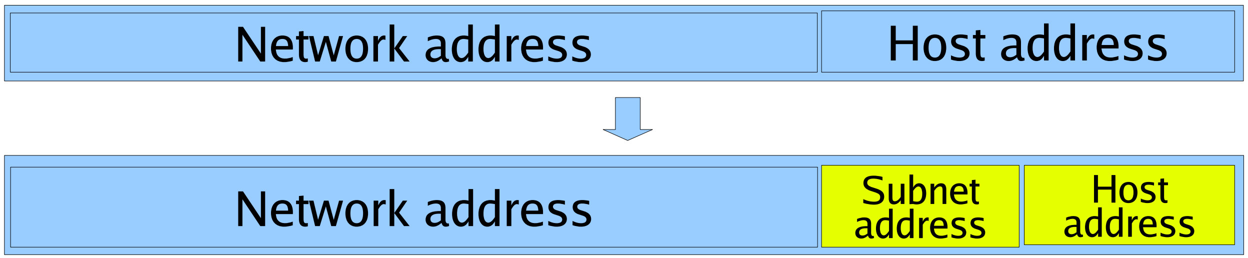
- The IP address and the associated netmask together tell you what other (neighbor) IP addresses are directly connected (i.e., only 1 hop away).



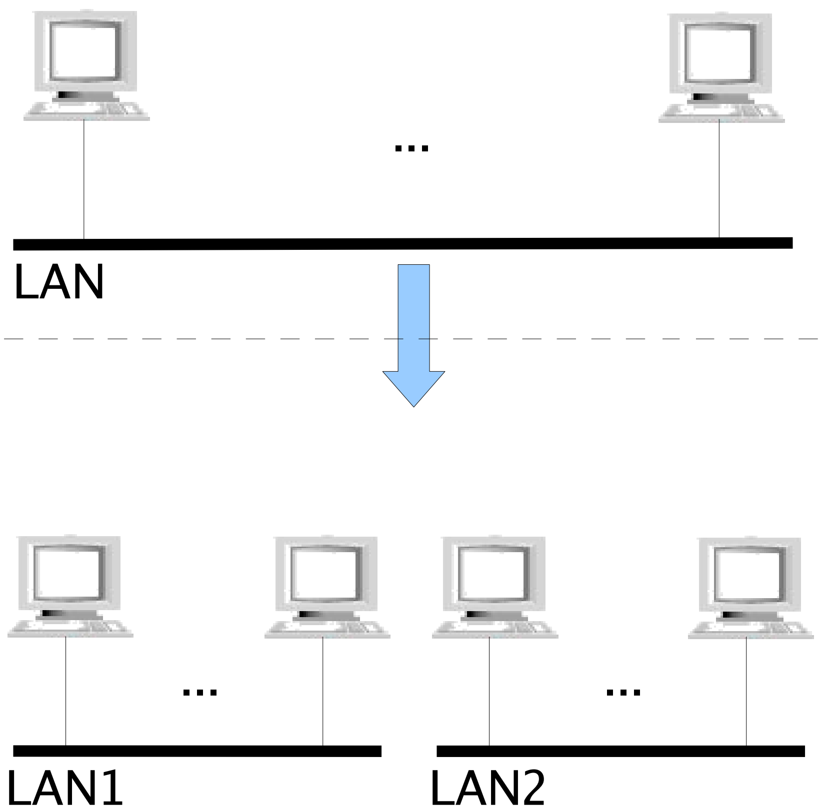
192.168.1.100/24 →

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

Example of subnetting (2/2)



Let's suppose we want 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

LAN1

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

LAN2

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

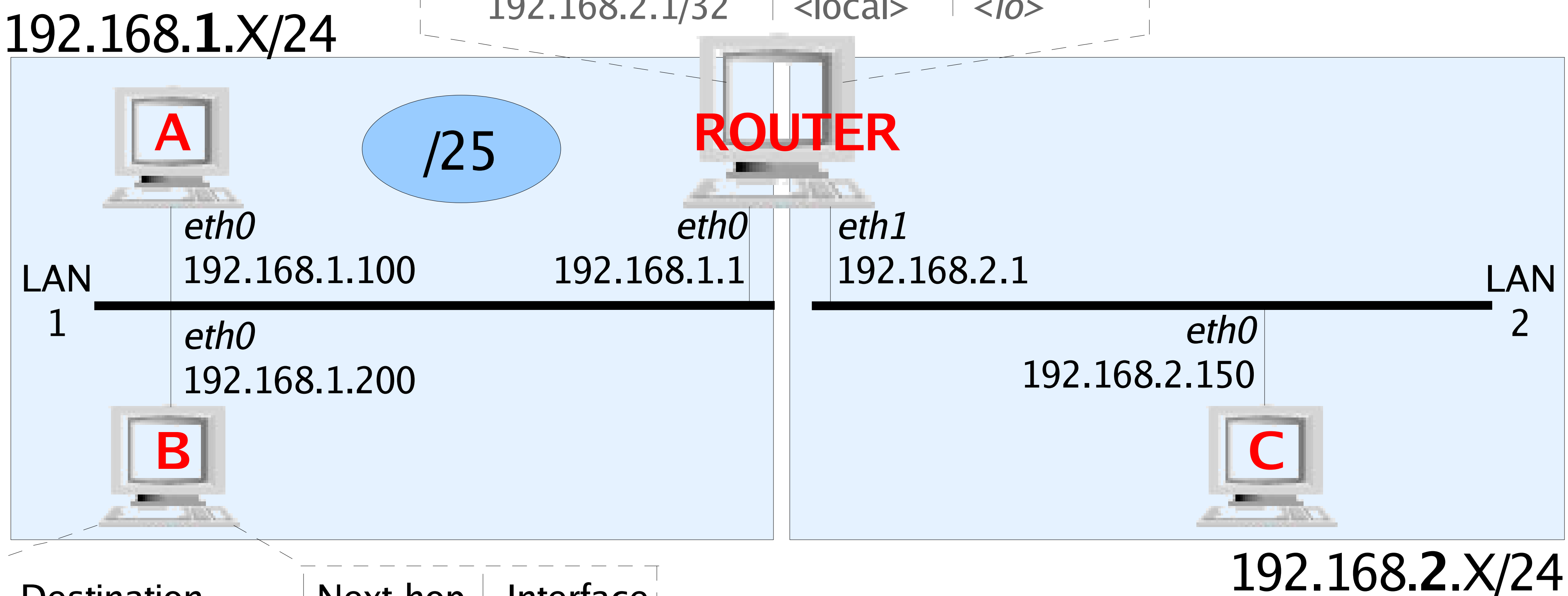
- Routing is the action needed to make it possible for hosts located in **different subnets** to communicate
- A Router is a network device that routes traffic.
- A routing table is a collection of routes that define how to reach a given network/subnet.
- A basic router routes traffic based on the destination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

Example

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

Routing table

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



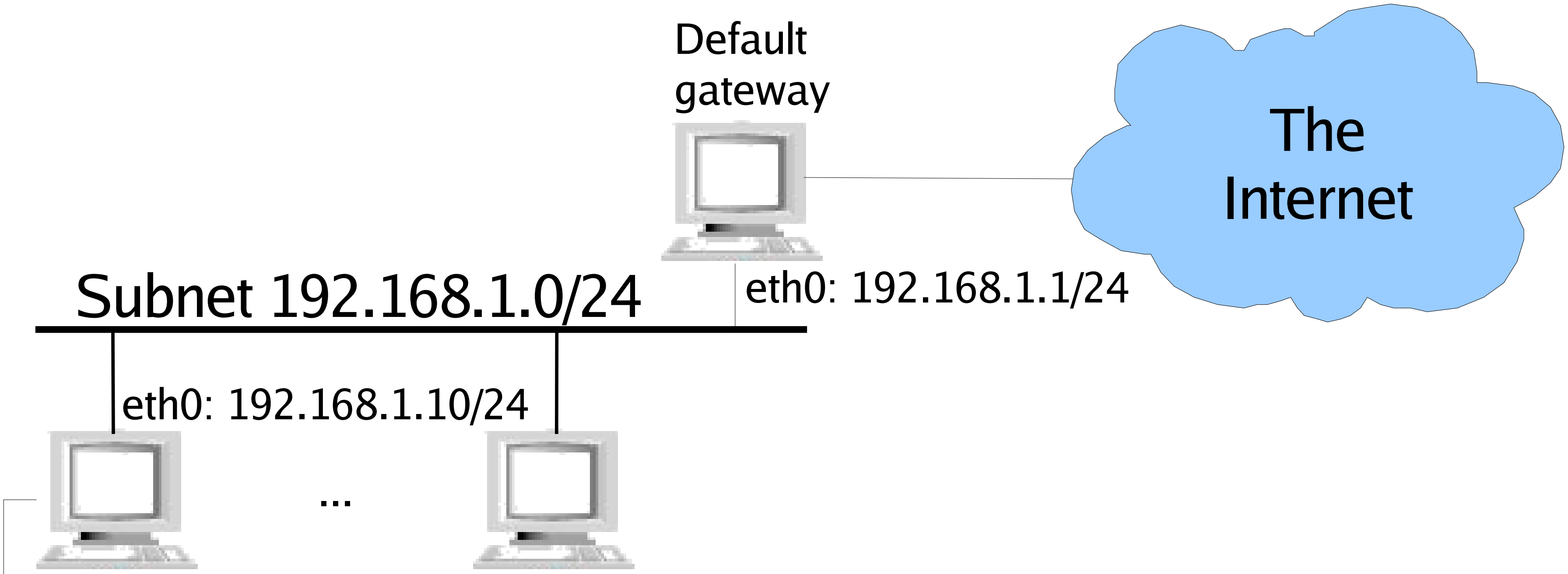
Destination	Next-hop	Interface
192.168.1.0/24	<direct>	*eth0*
192.168.1.200/32	<local>	<*lo*>

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 rarely do.

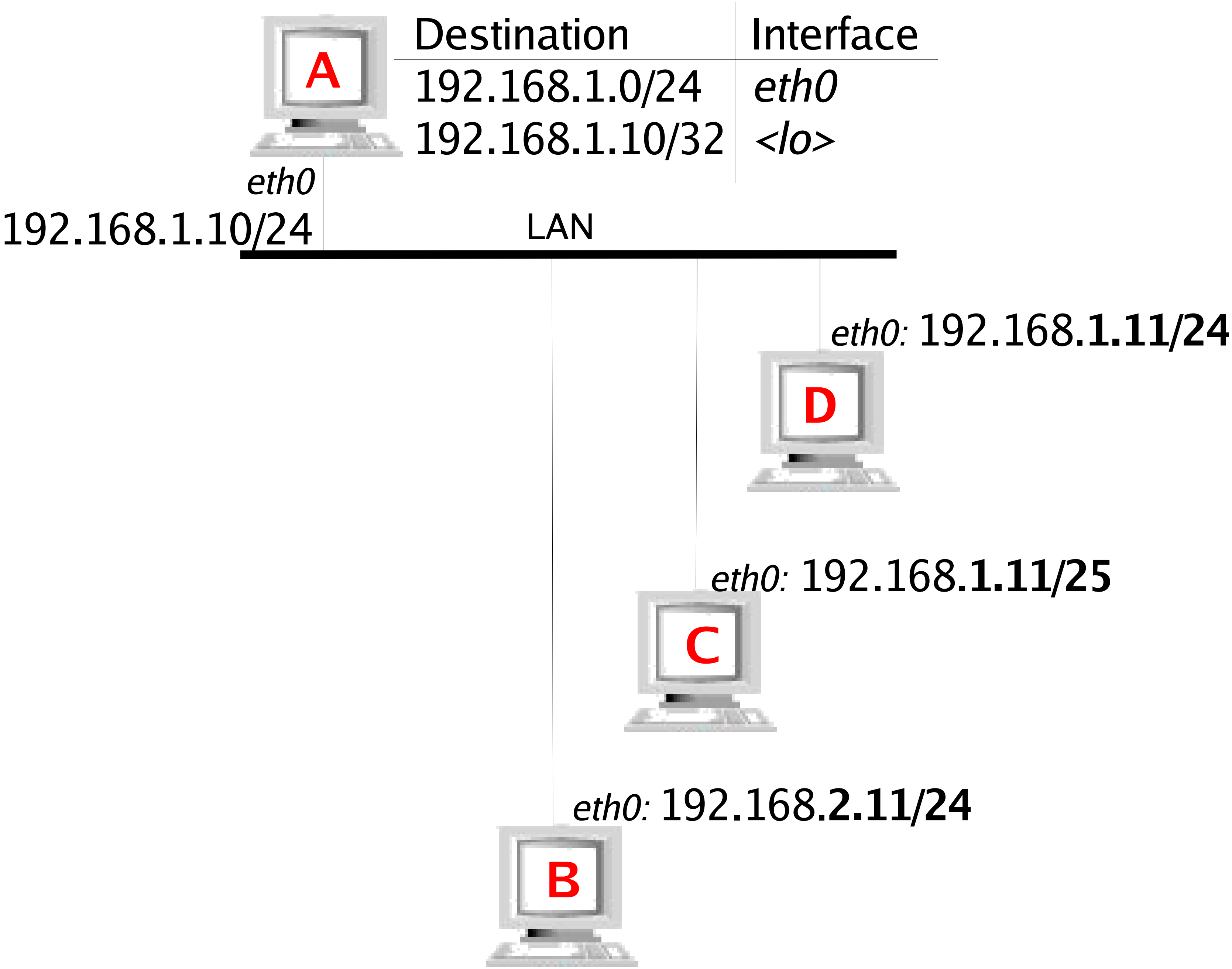
Example of default route



Destination	Next-hop	Interface
192.168.1.0/24	<direct>	eth0
0.0.0.0/0	**192.168.1.1**	**eth0**

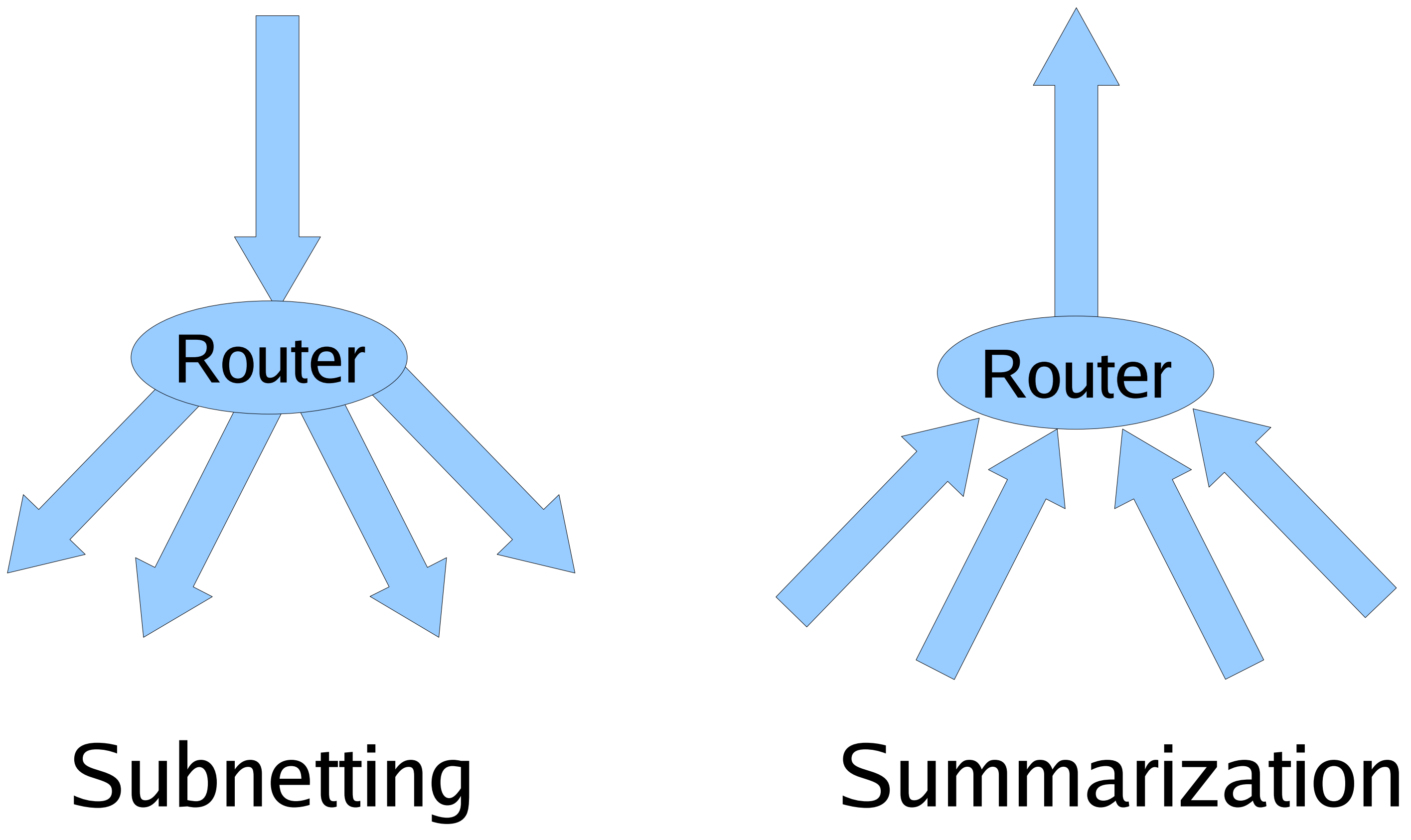
Exercise

Destination	Interface
192.168.1.0/24	*eth0*
192.168.1.10/32	*<lo>*

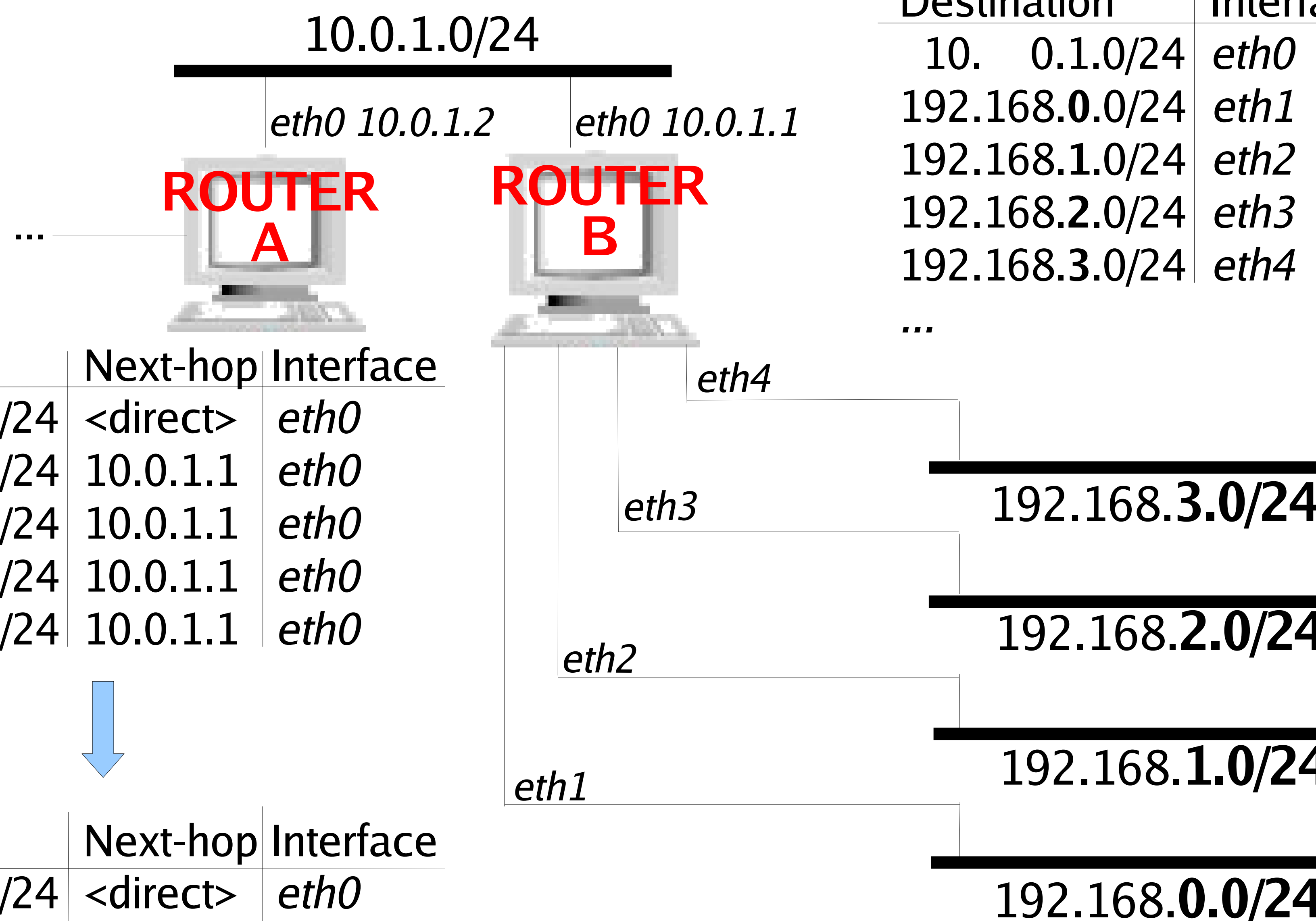


Summarization

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



Example of summarization

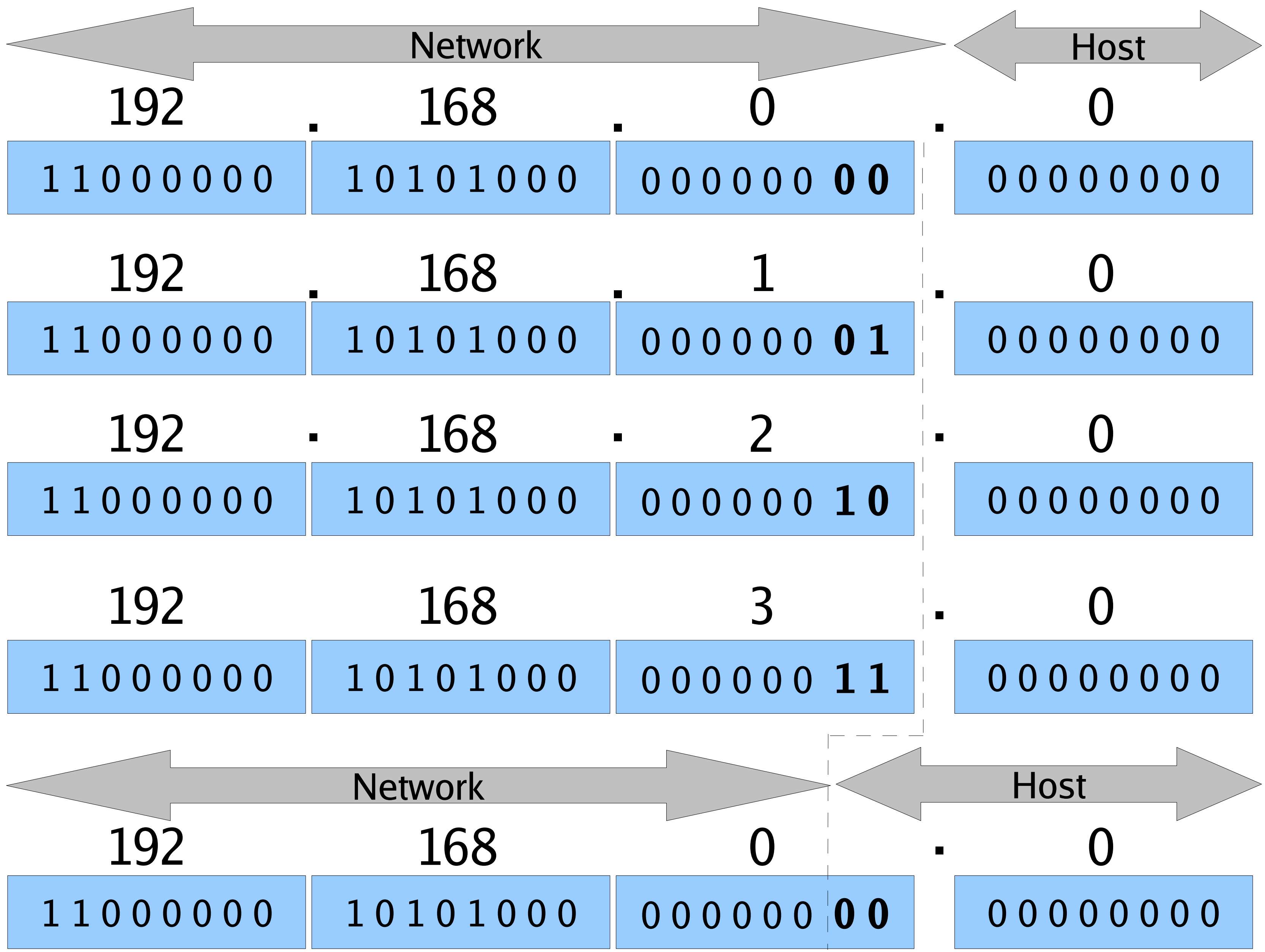


Destination	Interface
10.0.1.0/24	*eth0*
192.168.**0**.0/24	*eth1*
192.168.**1**.0/24	*eth2*
192.168.**2**.0/24	*eth3*
192.168.**3**.0/24	*eth4*
...	

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*
192.168.**0**.0/**22**	10.0.1.1	*eth0*
...		

Summarization: playing with the bits



Routing

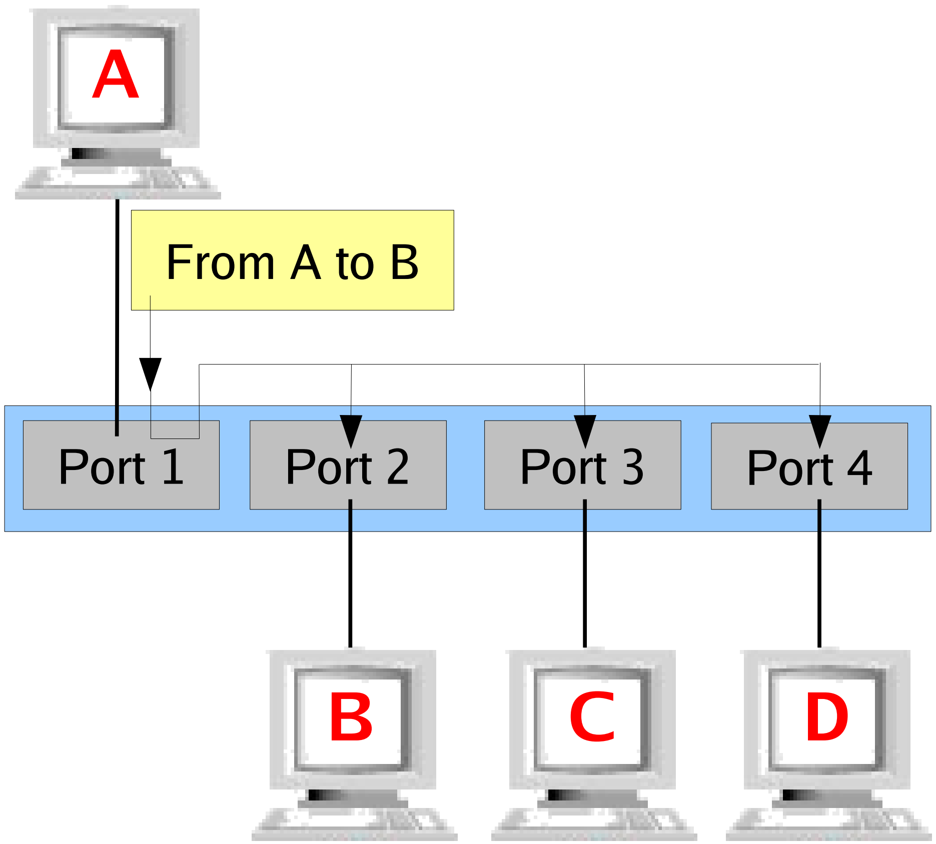
- Static VS Dynamic
- On a small network, static routing is sufficient.
  - Dynamic routing may be used anyway to provide some kind of high availability (to handle router failures)
- On bigger networks the use of dynamic routing becomes necessary
  - Depending on the size of the network and your exact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

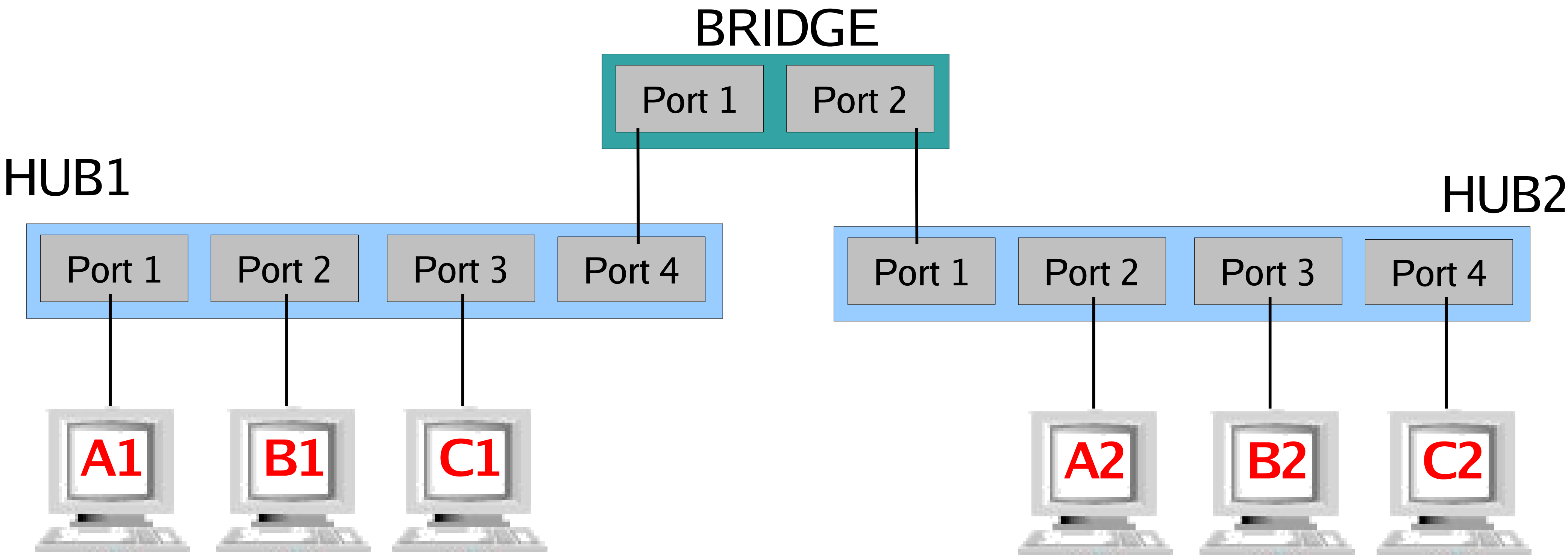
HUB

- Cheap
- Low performance



Can A talk to B while C talks to D?

Bridge



Case 1	From B1 to C1
Case 2	From B1 to C2
Case 3	From B2 to B1

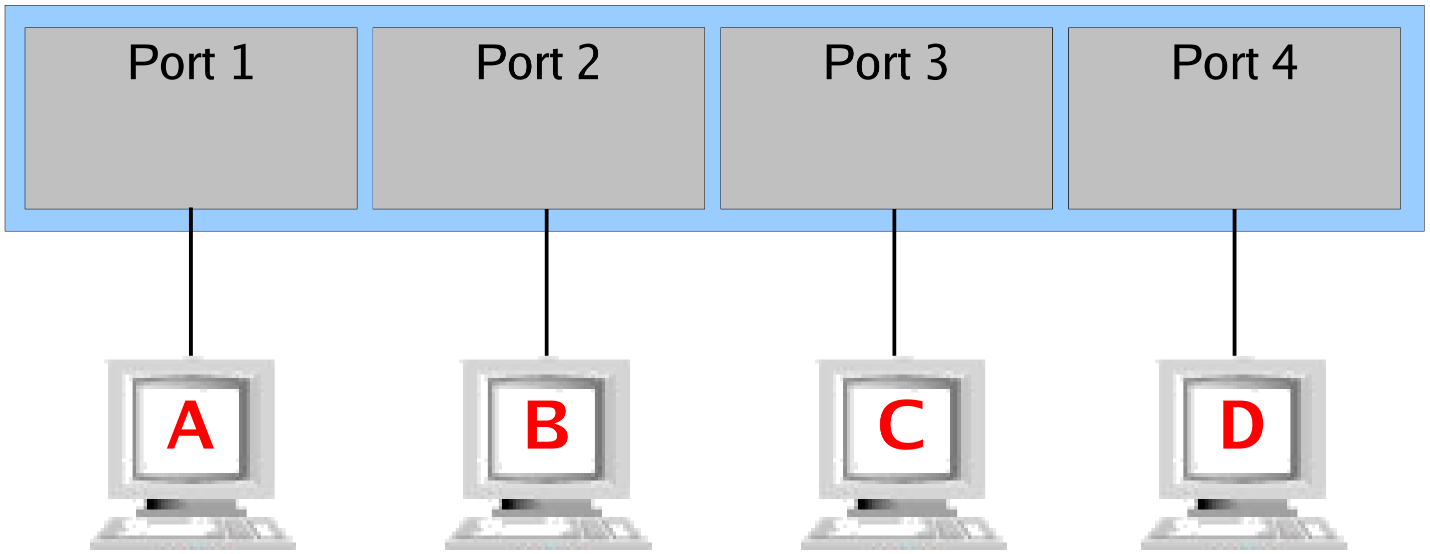
Can A1 talk to A2 while C1 talks to D1?

Can A1 talk to B1 while A2 talks to B2?

Can A1 talk to B1 while C1 talks to C2?

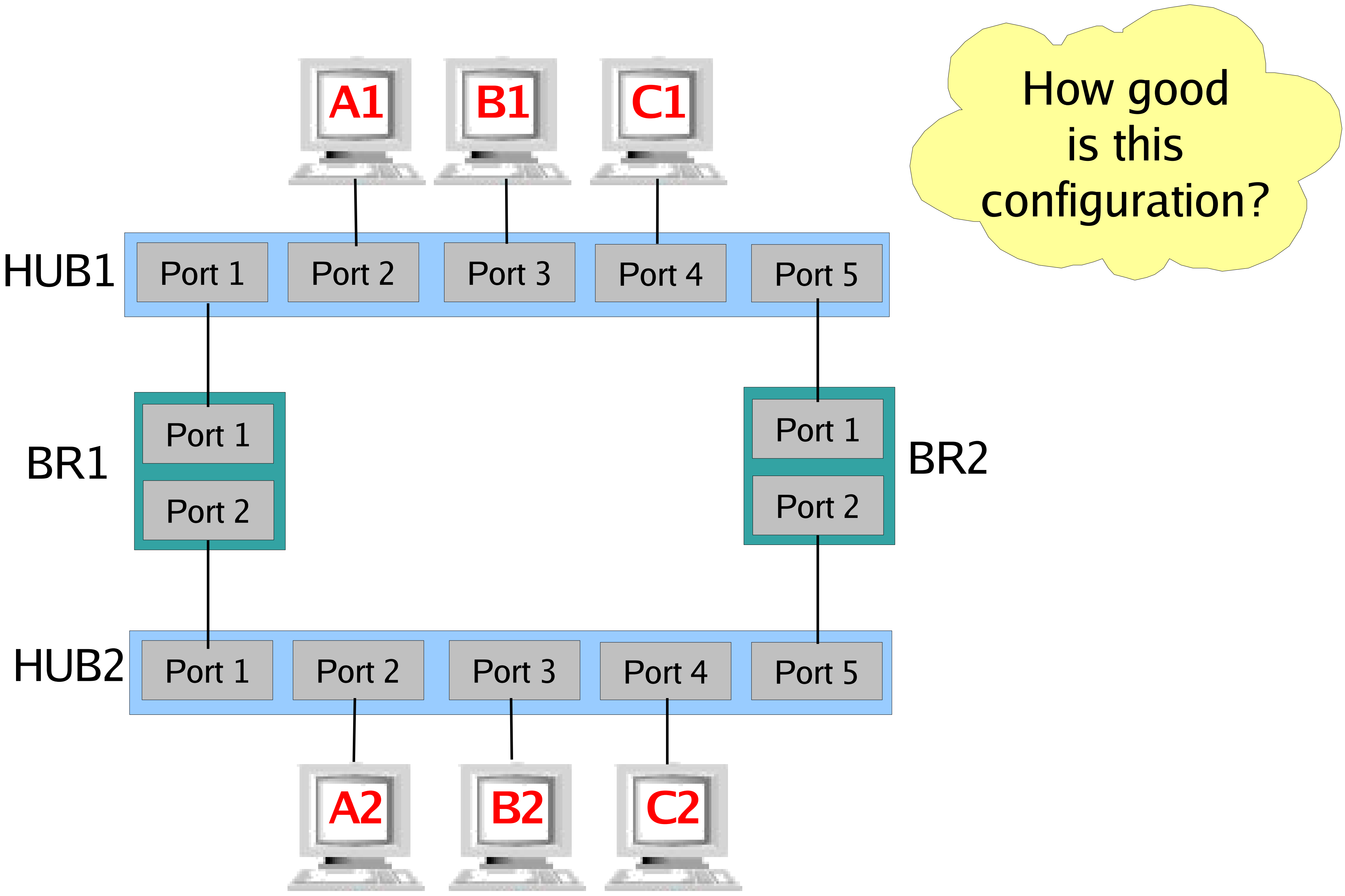
SWITCH

(sort of multi-port bridge)



Can A talk to B while C talks to D?

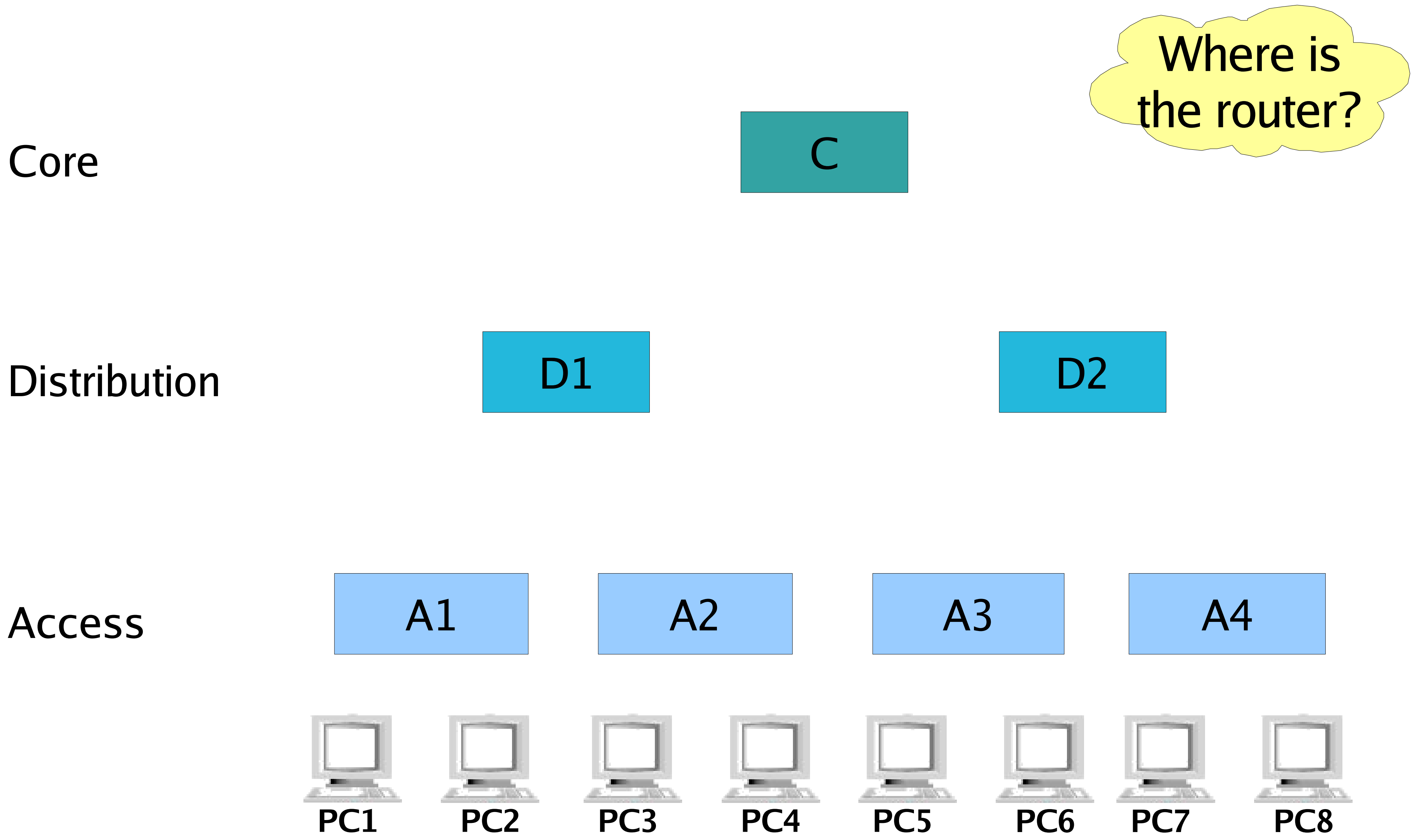
What if a bridge/switch fails?



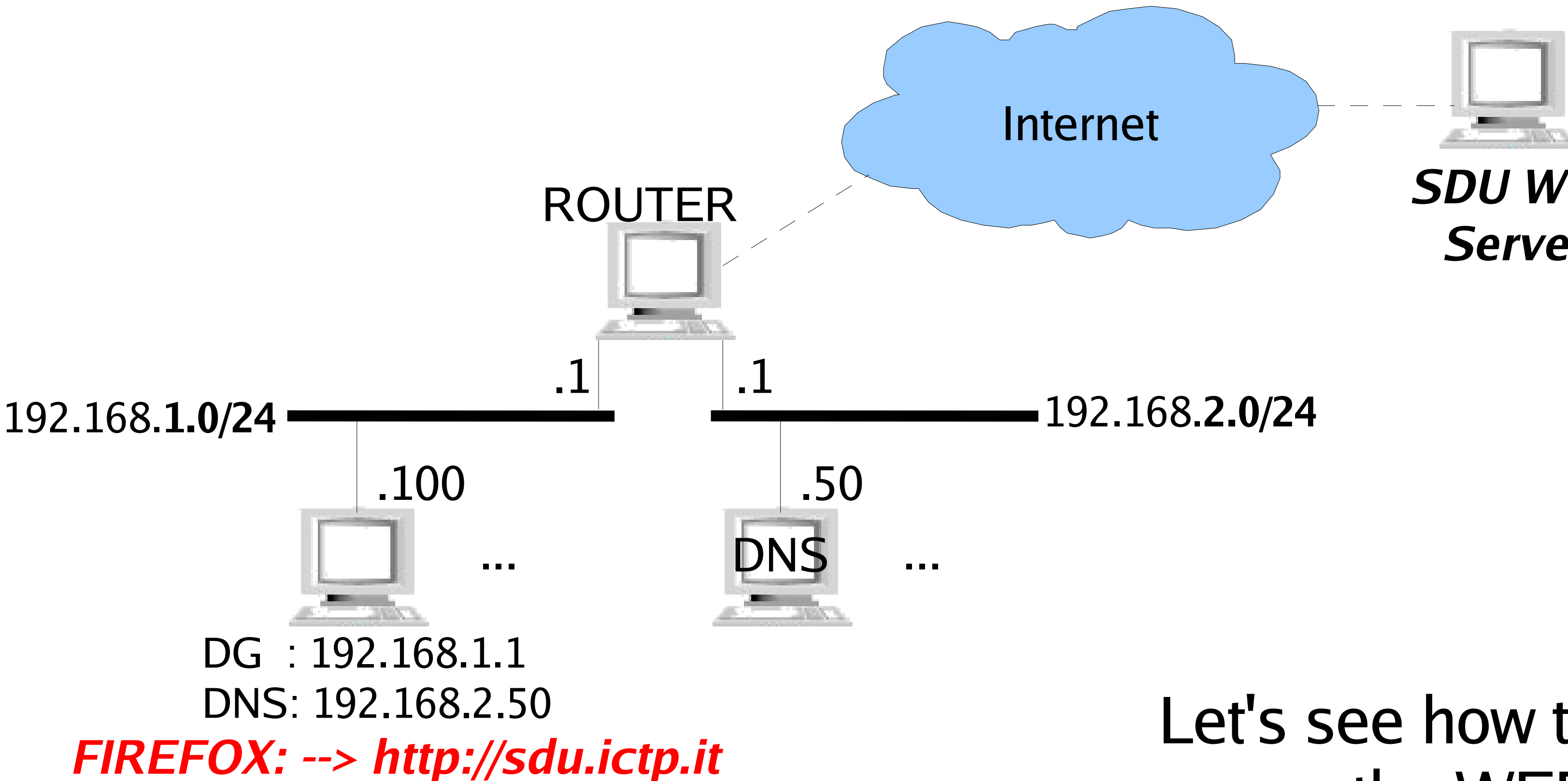
VERY IMPORTANT

- **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)



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

Any questions?

Copyright (cc) Creative Commons