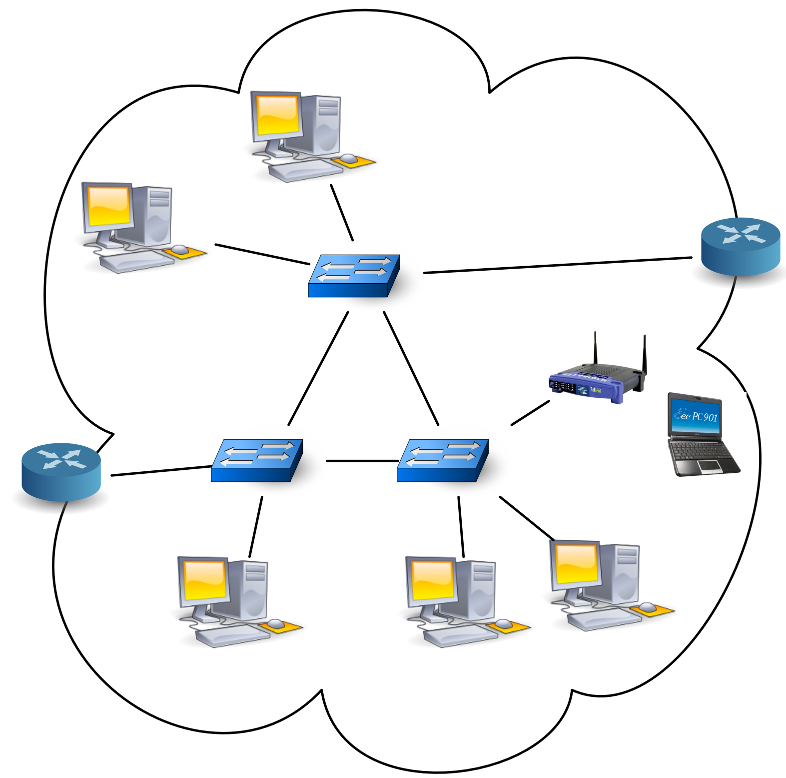
Local Area Networks Mechanisms

Claude Chaudet

LAN architecture

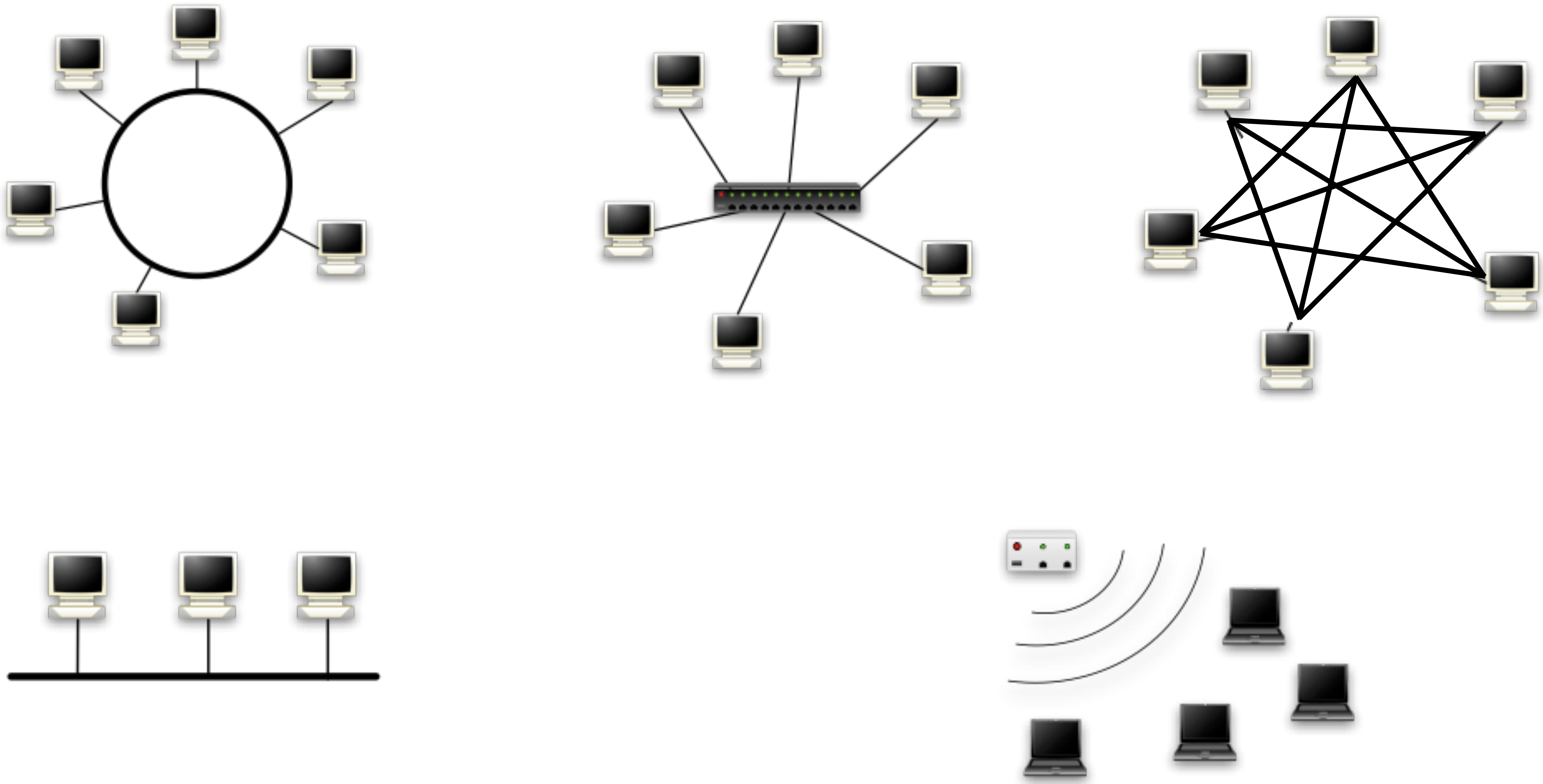
- **Behind (or between) routers, there are several devices**
  - End hosts & Link-layer interconnection devices (switches, bridges, Wi-Fi access points)
    - A LAN can be seen as a layer-2 network
    - Each equipment has IP addresses, which is not *necessary* within the LAN (but used anyway)



- **Characteristics**
  - Constrained to a single user / organization
  - Maximum size from hundreds to thousands of meters
    - Size is limited by physical constraints
  - Under the administrative control of a single authority
- **Technology**
  - Ethernet: 1 Gb/s
  - WiFi: 108 Mb/s, 54 Mb/s, 11 Mb/s
  - Typical delay: a few milliseconds

TELECOM ParisTech

LAN topologies examples



vendredi 25 octobre 13

LAN Addressing: MAC addresses

- **Why addresses?**
  - Not everyone needs to pass every frame at the upper layers
- **“MAC” Addresses**
  - Need to be unique for a given local area network
  - Source and destination addresses are included in the frame
  - One address per network adapter (thus, hosts may have >1)
  - Flat address space (as opposite to IP hierarchical space)
- **Typical address size is 48 bits**
  - 2^{48} (~ 10^{14}) individual addresses
  - Address collision rare but may happen => addresses are writable
- **Typical (e.g.., Ethernet, WiFi) address semantic:**
  - First 3 bytes: constructor identifier, allocated by IEEE
  - Last 3 bytes: constructor identifier, allocated by constructor
- **Special addresses: FF:FF:FF:FF:FF:FF used for broadcast**

Commutation devices: switches

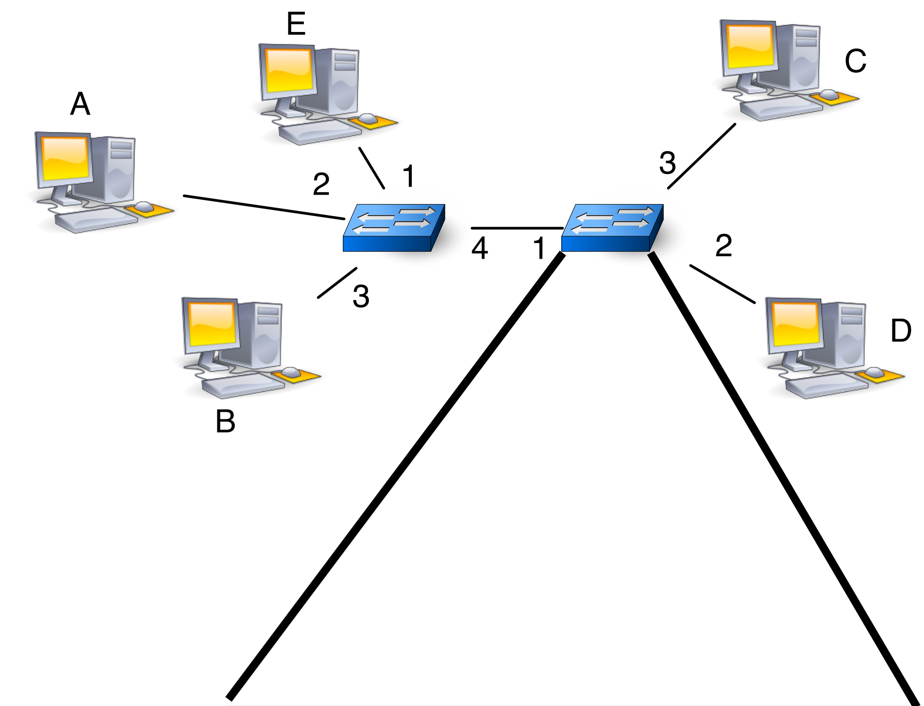
- **A switch forwards frames based on the destination MAC address**

- **Operation modes:**
  - Store-and-forward mode
    - Reception of the whole frame, integrity check, buffer storage, forwarding on an output interface
  - Cut-through mode
    - Look only at the header before starting retransmission
    - Requires less memory but higher load on the links
    - Not much used today, as memory price decreases

- **Deals with a huge amount of frames**
  - 1 Gb/s ; 1500 bytes frames => 83 000 frames per second
  - Buffer size depends on the number of interfaces, on the throughput and on the time required to process a frame

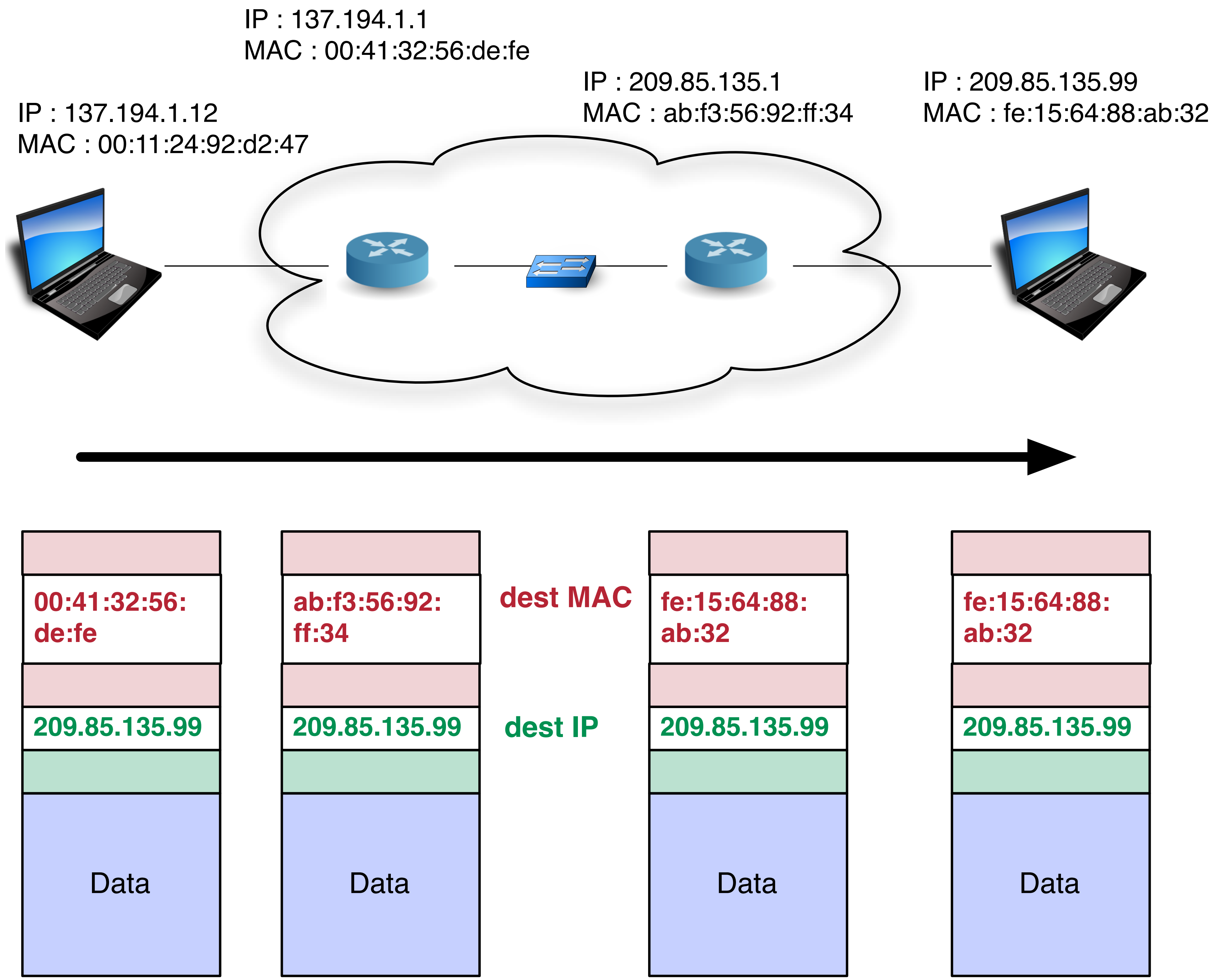
Switches commutation table

- **Switches keep track of which terminal is connected on which interface**
  - Use of a commutation table
- **Table updates:**
  - For every frame, examine the source address
  - Update table, noting that information
  - Entries expire after a certain timeout
- **Simple algorithm:**
  - Allows nodes mobility
  - Deals with nodes failures
  - Does not require dedicated communication



Dest.	Interf.
A	1
B	1
C	3
D	2
E	1

Addresses Scope



Address Resolution Protocol: ARP

IP - MAC correspondance: ARP table

- **Every equipment working at routing level owns two addresses:**
  - A MAC address, allocated by the manufacturer
  - An IP address, allocated by the network administrator

- **How is the match realized?**
  - ARP (Address Resolution Protocol)
- **Every node (terminal, router, ...) has an internal matching table**

```
 infres-164.enst.fr (137.194.164.1) at aa:0:5:0:a4:1 on en0 [ethernet]
infres4.enst.fr (137.194.164.4) at 0:3:ba:3a:2f:a1 on en0 [ethernet]
infres5.enst.fr (137.194.164.5) at aa:0:5:0:a4:5 on en0 [ethernet]
fiona.enst.fr (137.194.164.31) at 0:c:6e:b8:93:4e on en0 [ethernet]
nirgua.enst.fr (137.194.164.46) at 0:16:76:90:12:22 on en0 [ethernet]
chaudet.enst.fr (137.194.164.58) at 0:d:93:61:dc:5e on en0 [ethernet]
deserec1.enst.fr (137.194.164.81) at 0:19:d1:a0:4:39 on en0 [ethernet]
```

ARP protocol

- **Layer 3 control protocol**
  - Manipulates IP addresses

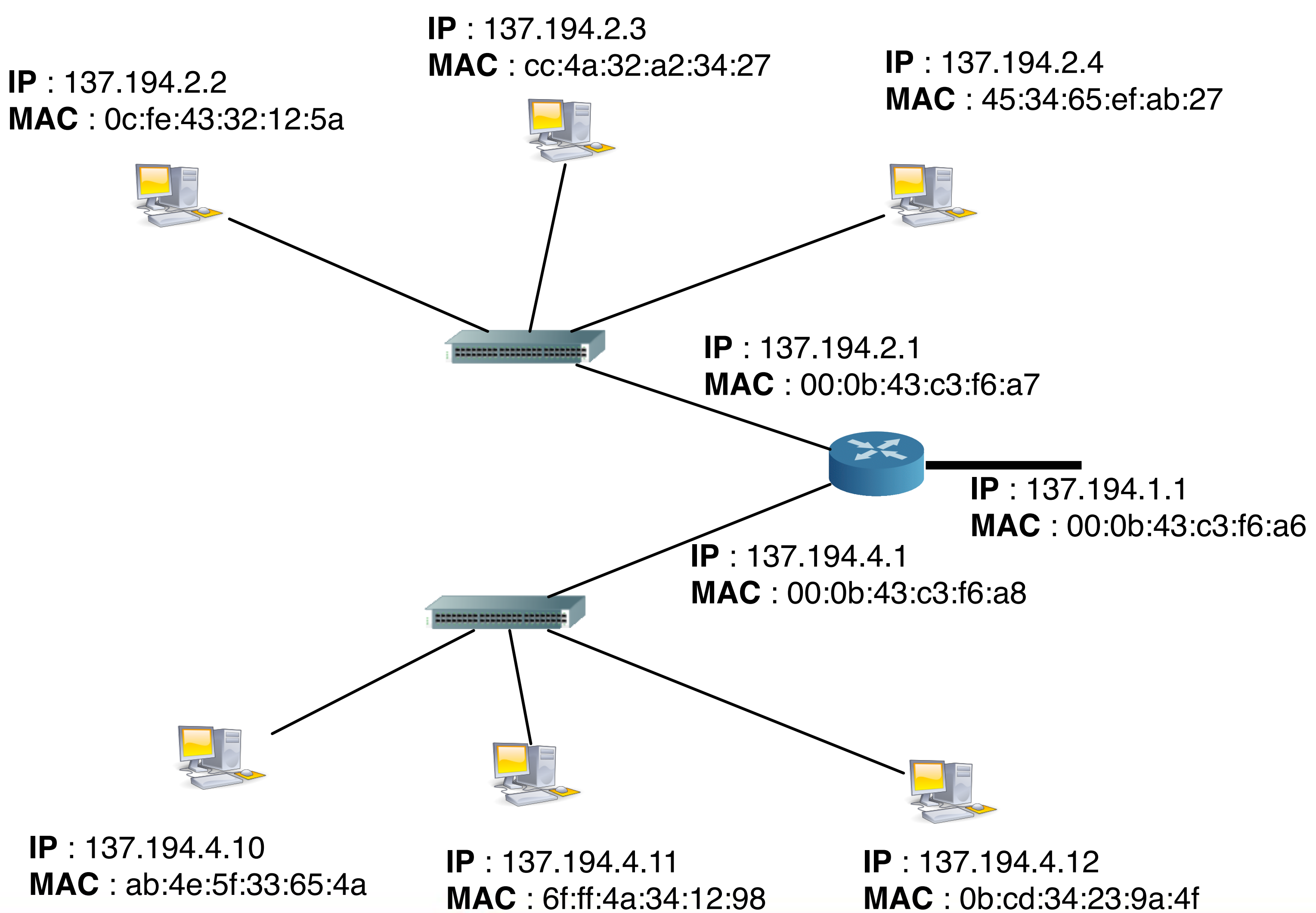
- **Works on a request-response mechanism**

- **When an IP packet needs to cross a layer-2 "cloud"**
  - Examine the IP address in the ingress router
  - Look for the corresponding MAC address in the table
  - If the MAC address is unknown, buffer the packet and send a request "who owns IP address x.x.x.x", broadcasted on the LAN
  - If this address is present on the network, the terminal answers with an unicast frame.

TELECOM ParisTech

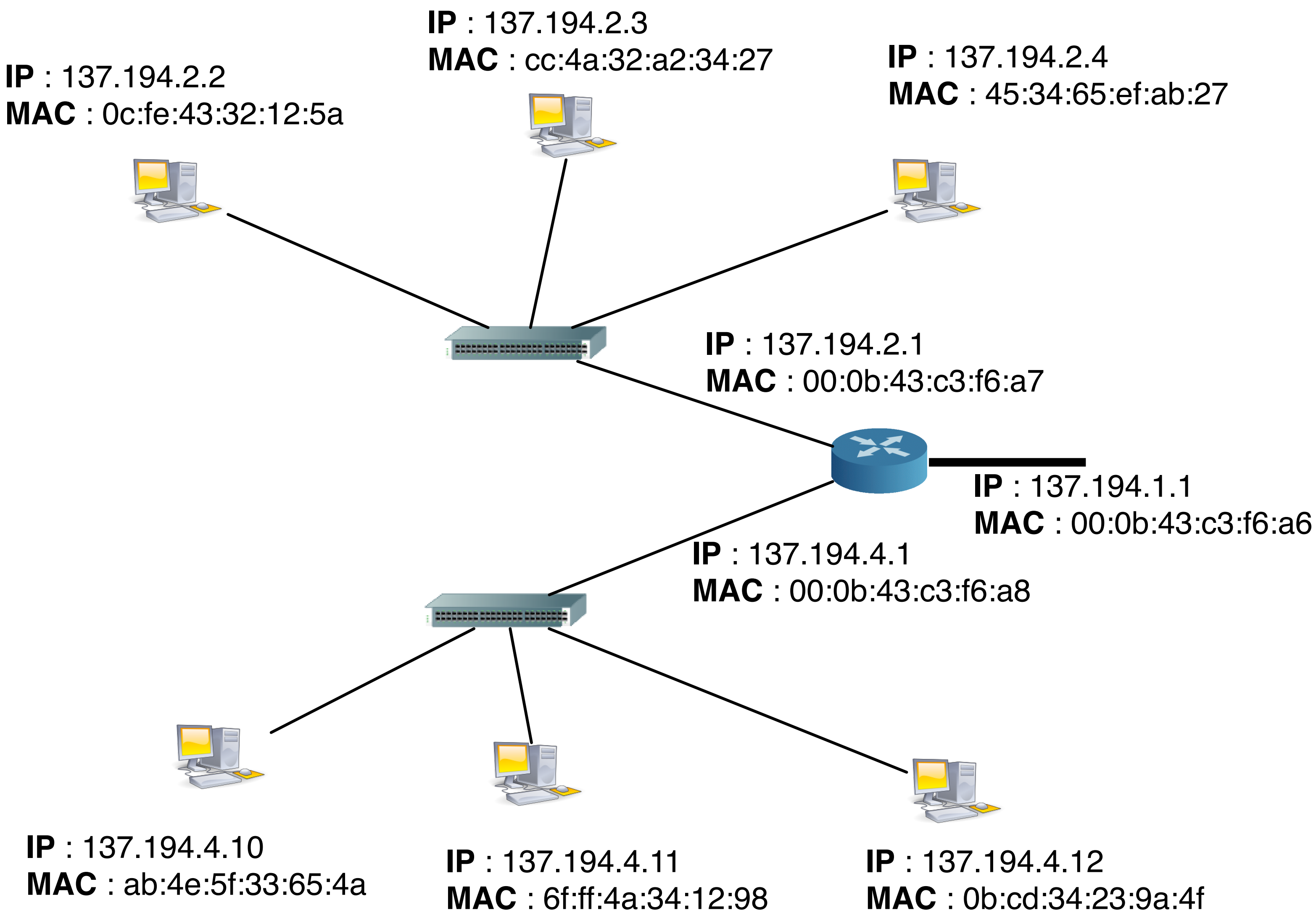
ARP: example

- **Example of a network composed of two sub-networks**
  - Network: 137.194.0.0 / 16
  - Sub-network 1: 137.194.2.0 / 24 Gateway: 137.194.2.1
  - Sub-network 2: 137.194.4.0 / 24 Gateway: 137.194.4.1



Within one sub-network

- **Broadcasted ARP request**
  - Only the machine that owns the address answers



TELECOM ParisTech

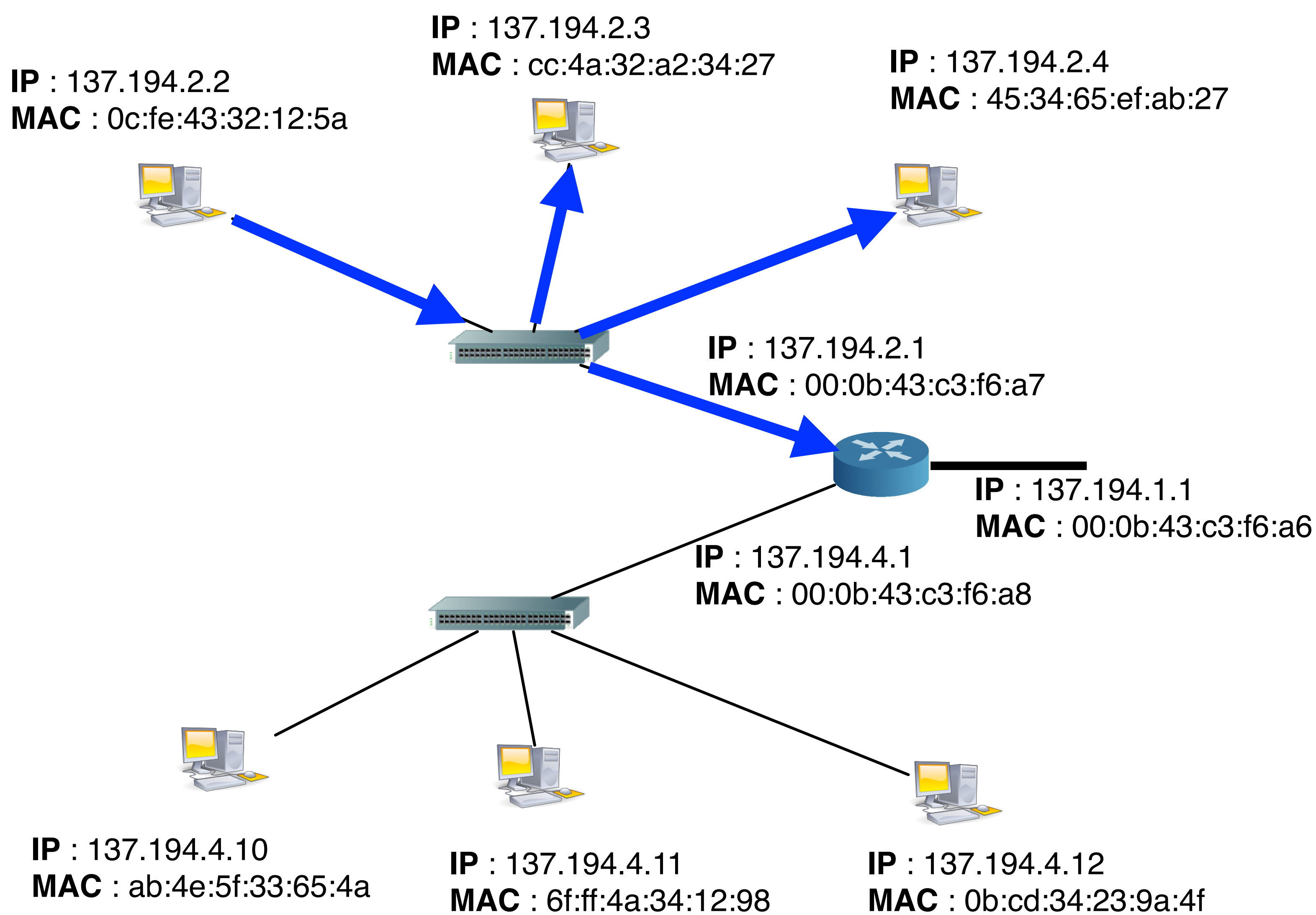
Within one sub-network

Broadcasted ARP request

- Only the machine that owns the address answers

Request

Source IP	137.194.2.2
Source MAC	0c:fe:43:32:12:5a
Dest IP	137.194.2.4
Dest MAC	ff:ff:ff:ff:ff:ff



Within one sub-network

Broadcasted ARP request

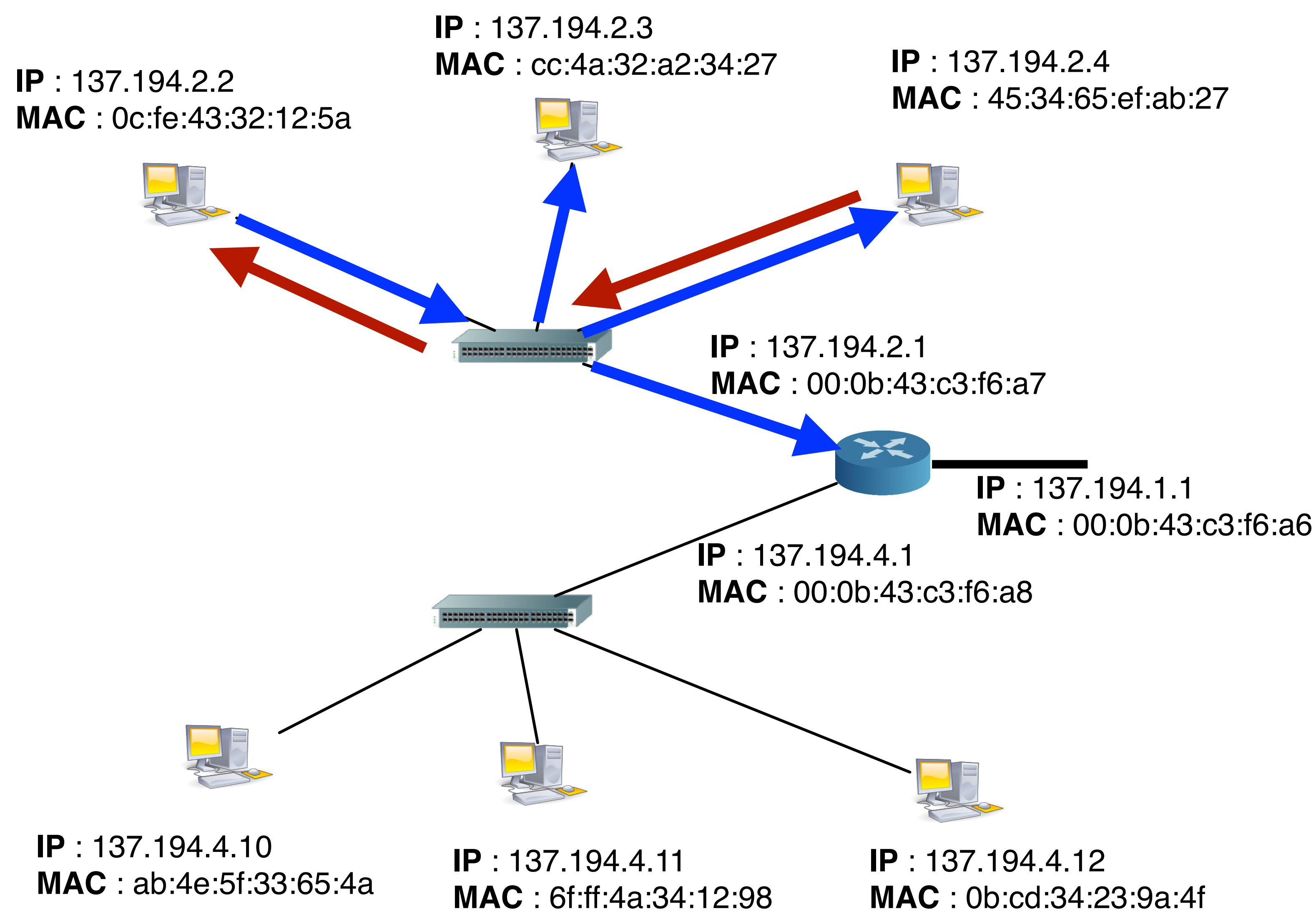
- Only the machine that owns the address answers

Request

Source IP	137.194.2.2
Source MAC	0c:fe:43:32:12:5a
Dest IP	137.194.2.4
Dest MAC	ff:ff:ff:ff:ff:ff

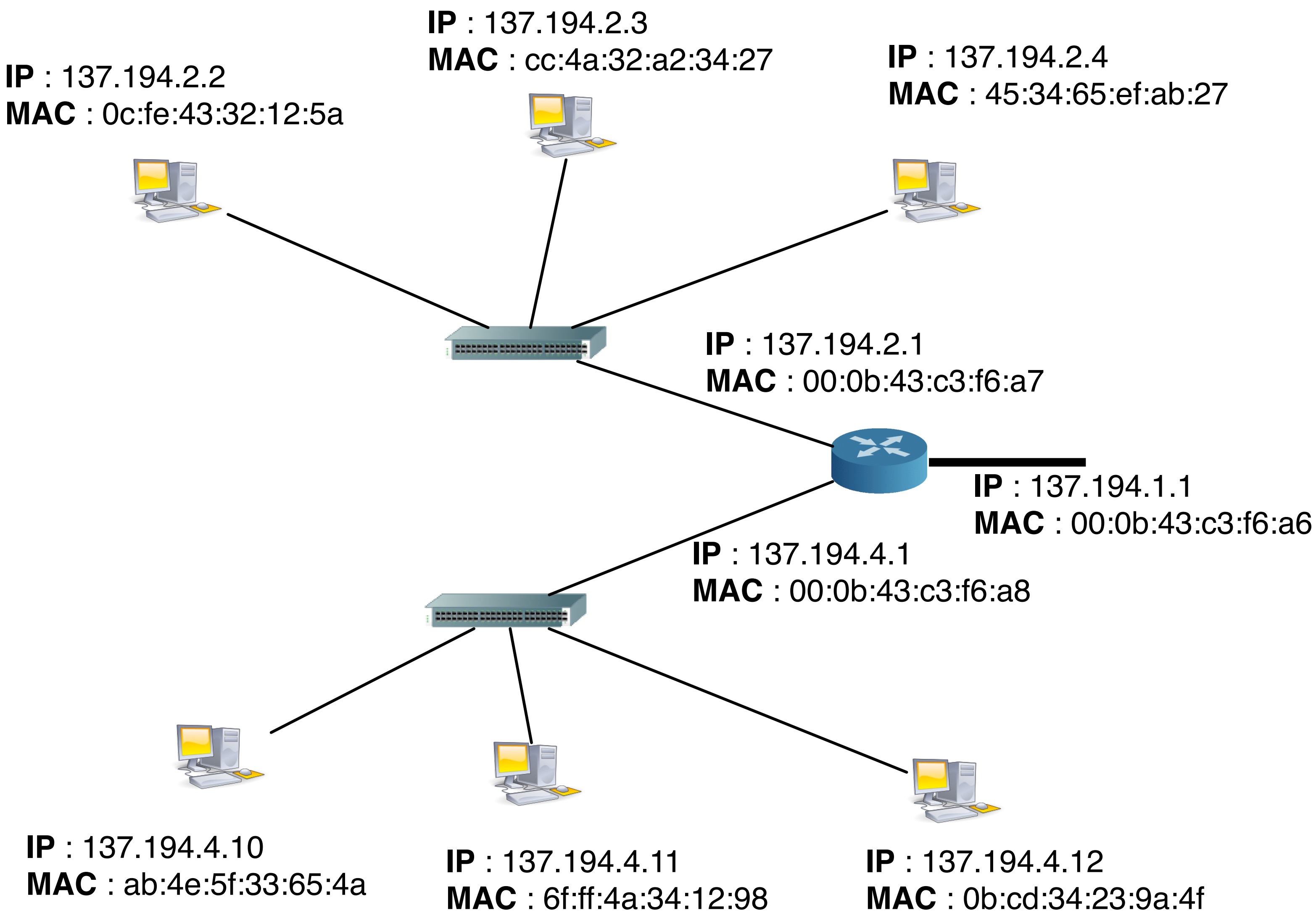
Answer

Source IP	137.194.2.4
Source MAC	45:34:65:ef:ab:27
Dest IP	137.194.2.2
Dest MAC	0c:fe:43:32:12:5a



Between sub-networks

- **We do not aim for the destination, but for the gateway**
  - The rest of the process is similar

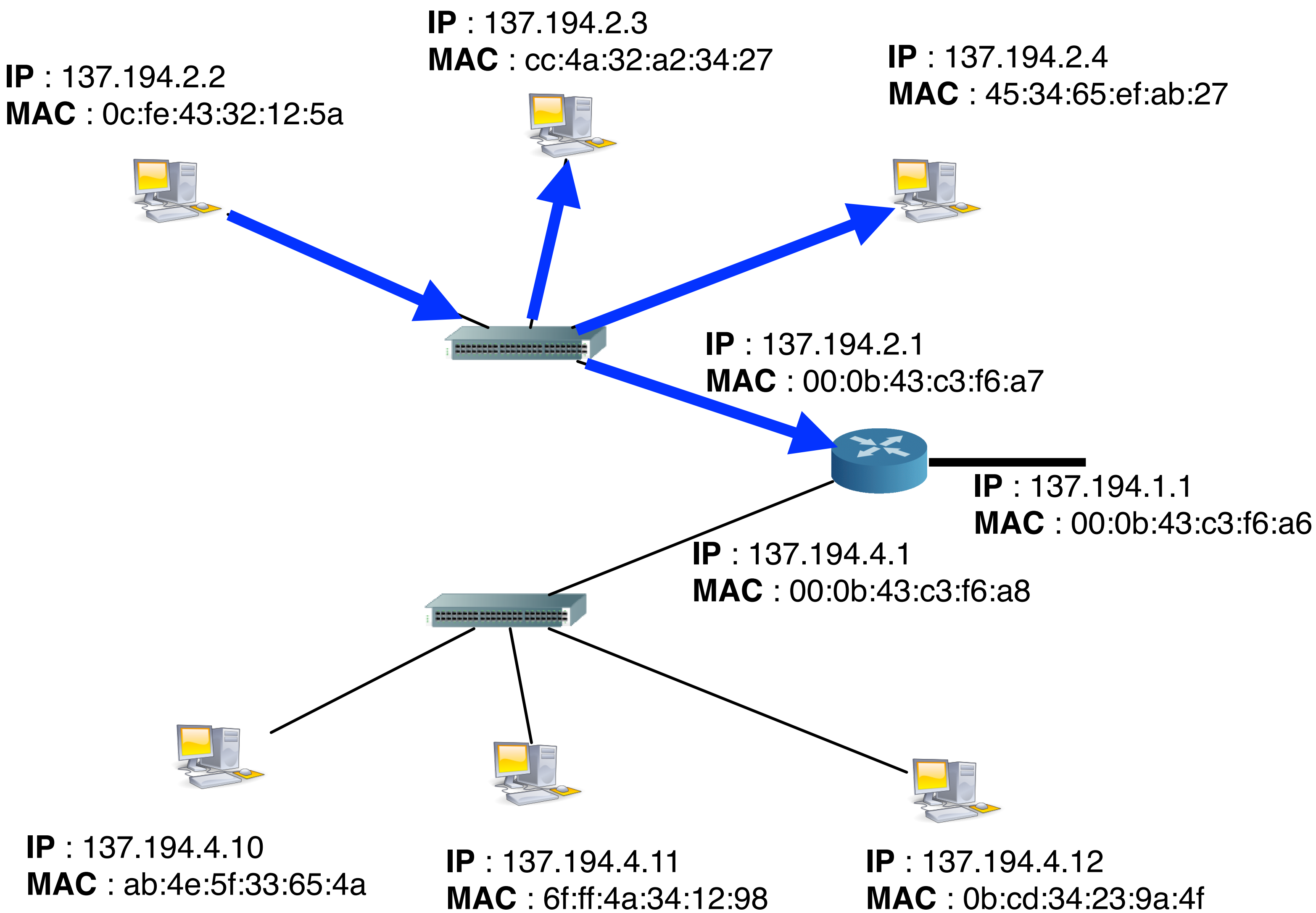


Between sub-networks

- **We do not aim for the destination, but for the gateway**
  - The rest of the process is similar

Requête

Source IP	137.194.2.2
Source MAC	0c:fe:43:32:12:5a
Dest IP	137.194.2.1
Dest MAC	ff:ff:ff:ff:ff:ff



Between sub-networks

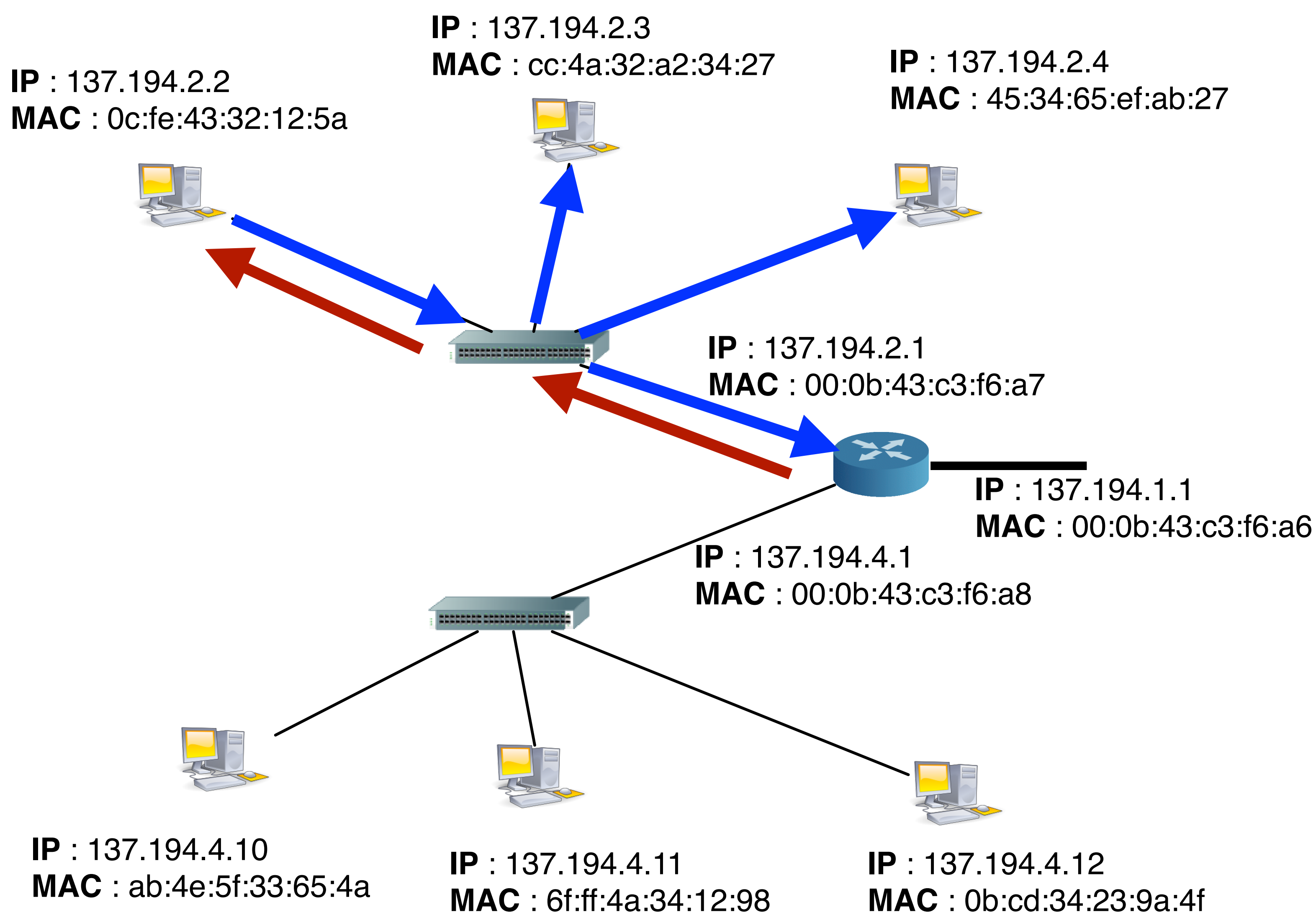
- **We do not aim for the destination, but for the gateway**
  - The rest of the process is similar

Requête

Source IP	137.194.2.2
Source MAC	0c:fe:43:32:12:5a
Dest IP	137.194.2.1
Dest MAC	ff:ff:ff:ff:ff:ff

Réponse

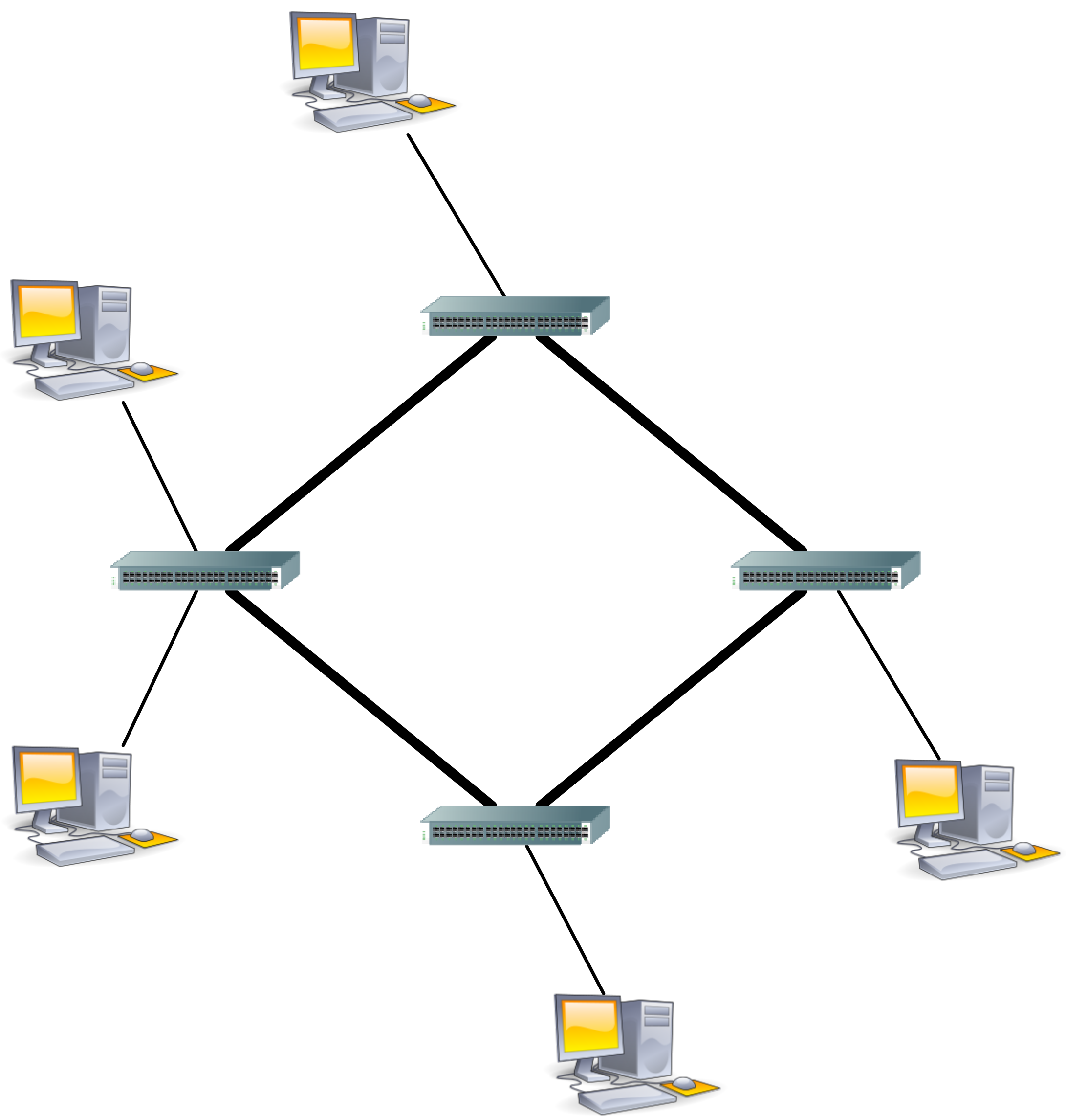
Source IP	137.194.2.1
Source MAC	00:0b:43:c3:f6:a7
Dest IP	137.194.2.2
Dest MAC	0c:fe:43:32:12:5a



Spanning Tree Protocol

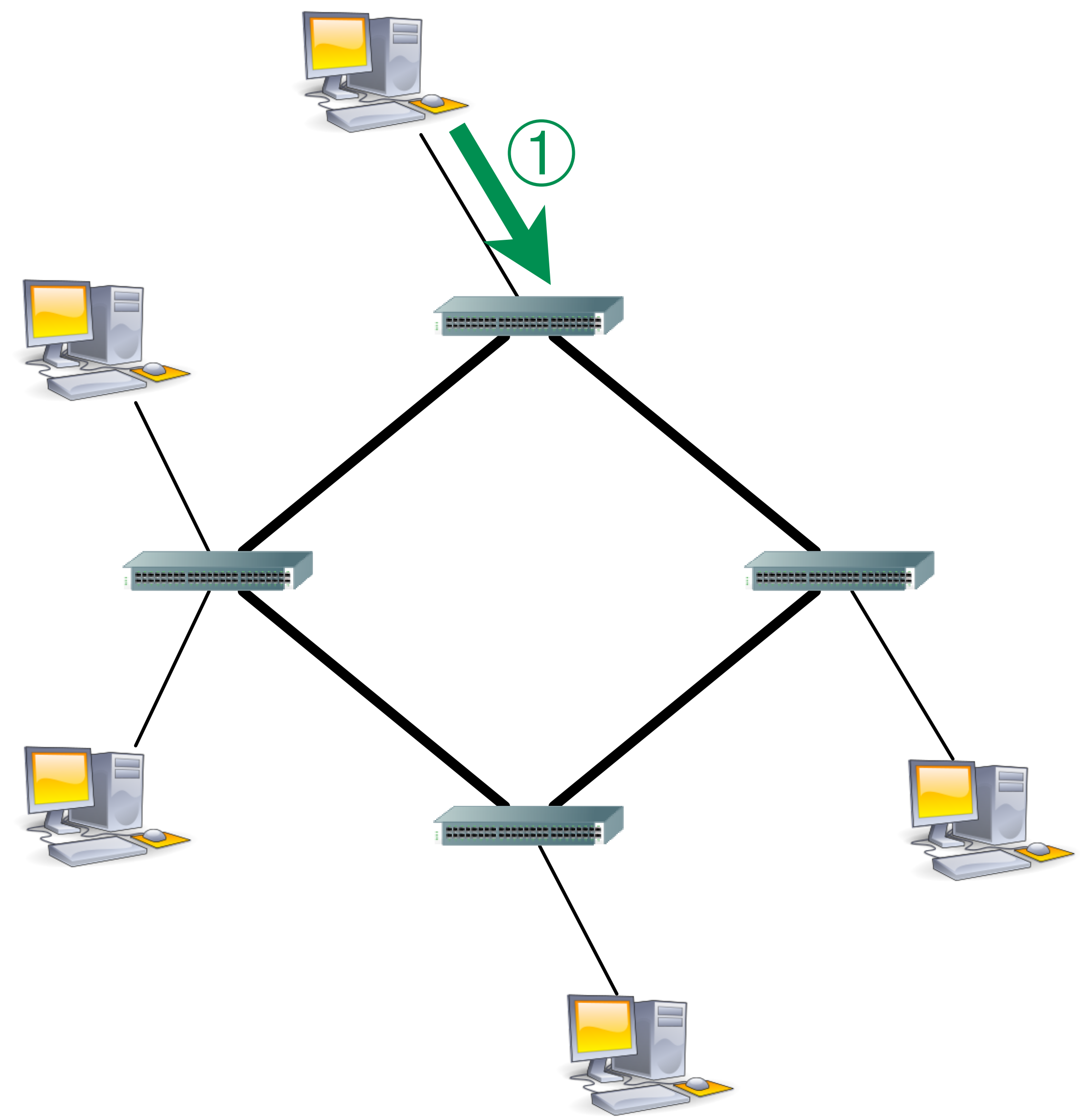
LAN limits

- **Addressing scheme is « flat »**
  - Selecting the correct output interface requires the lookup into the commutation table
  - Size of the table increases with the number of stations
  - Remember : 83 000 frames / sec / input interface
- **No Broadcast frames filtering**
  - A LAN constitutes a unique broadcast domain
  - Broadcast frames become a problem when redundancy appears in the topology



LAN limits

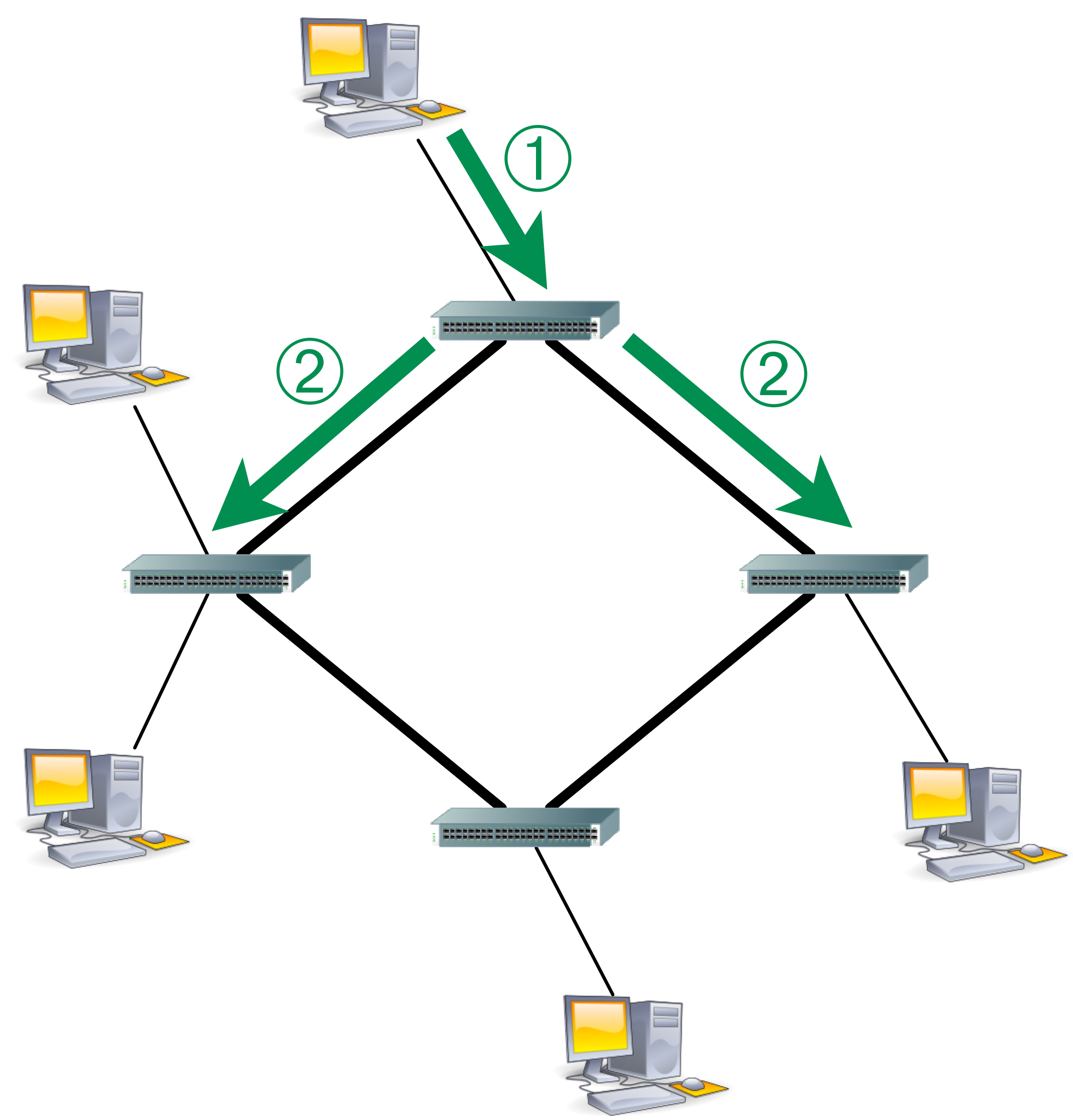
- **Addressing scheme is « flat »**
  - Selecting the correct output interface requires the lookup into the commutation table
  - Size of the table increases with the number of stations
  - Remember : 83 000 frames / sec / input interface



- **No Broadcast frames filtering**
  - A LAN constitutes a unique broadcast domain
  - Broadcast frames become a problem when redundancy appears in the topology

LAN limits

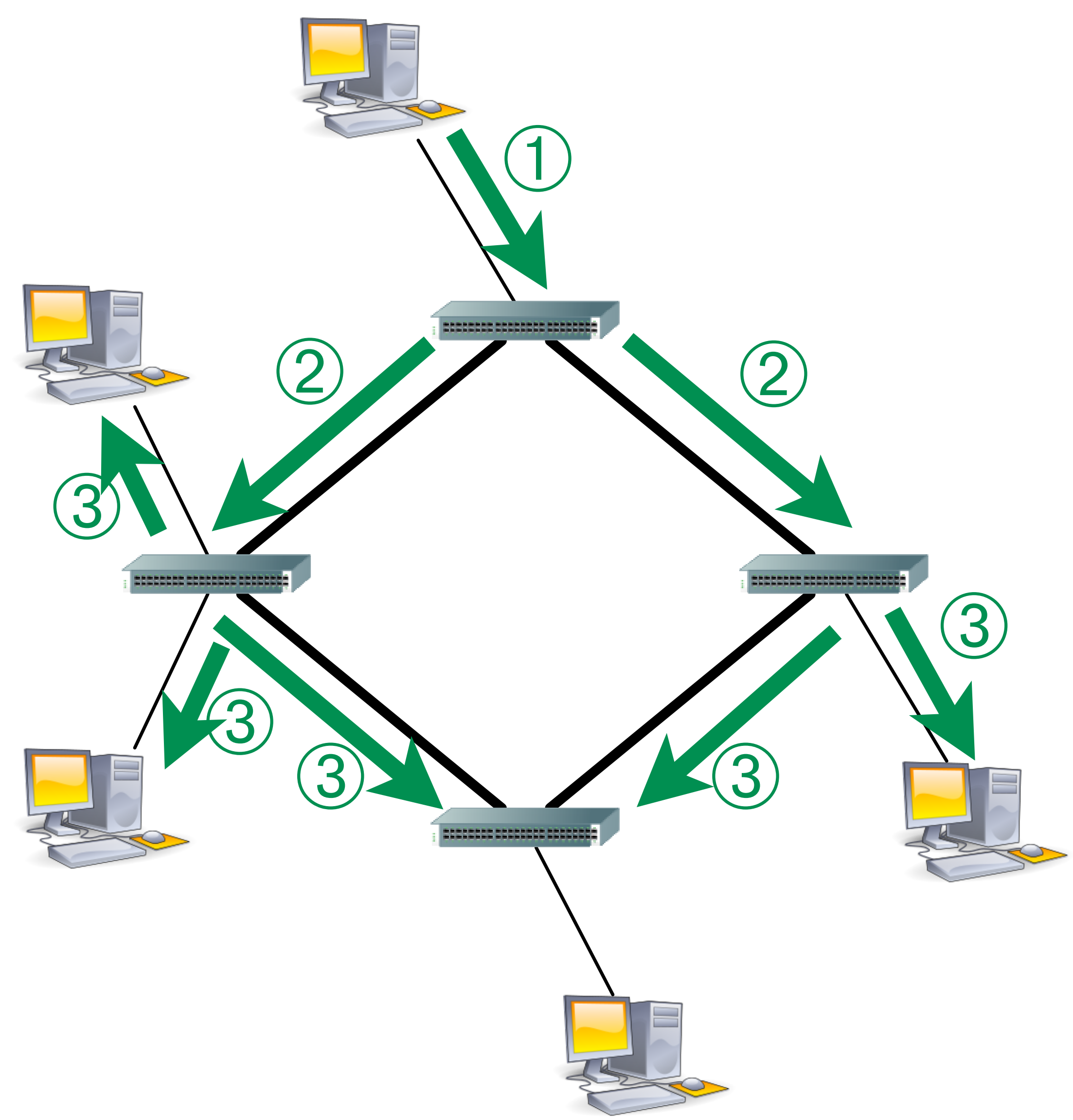
- **Addressing scheme is « flat »**
  - Selecting the correct output interface requires the lookup into the commutation table
  - Size of the table increases with the number of stations
  - Remember : 83 000 frames / sec / input interface



- **No Broadcast frames filtering**
  - A LAN constitutes a unique broadcast domain
  - Broadcast frames become a problem when redundancy appears in the topology

LAN limits

- **Addressing scheme is « flat »**
  - Selecting the correct output interface requires the lookup into the commutation table
  - Size of the table increases with the number of stations
  - Remember : 83 000 frames / sec / input interface

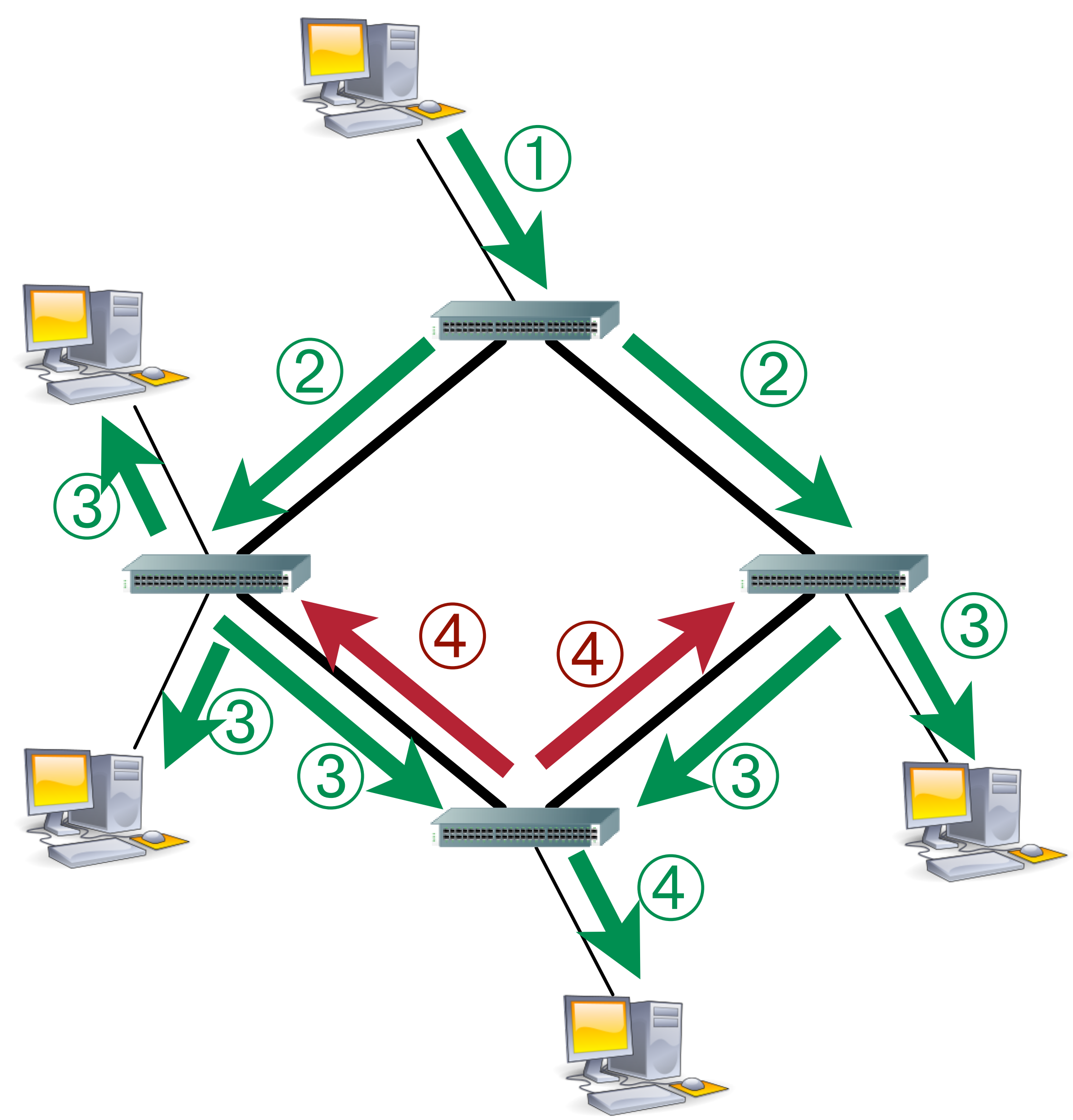


- **No Broadcast frames filtering**
  - A LAN constitutes a unique broadcast domain
  - Broadcast frames become a problem when redundancy appears in the topology

TELECOM ParisTech

LAN limits

- **Addressing scheme is « flat »**
  - Selecting the correct output interface requires the lookup into the commutation table
  - Size of the table increases with the number of stations
  - Remember : 83 000 frames / sec / input interface
- **No Broadcast frames filtering**
  - A LAN constitutes a unique broadcast domain
  - Broadcast frames become a problem when redundancy appears in the topology



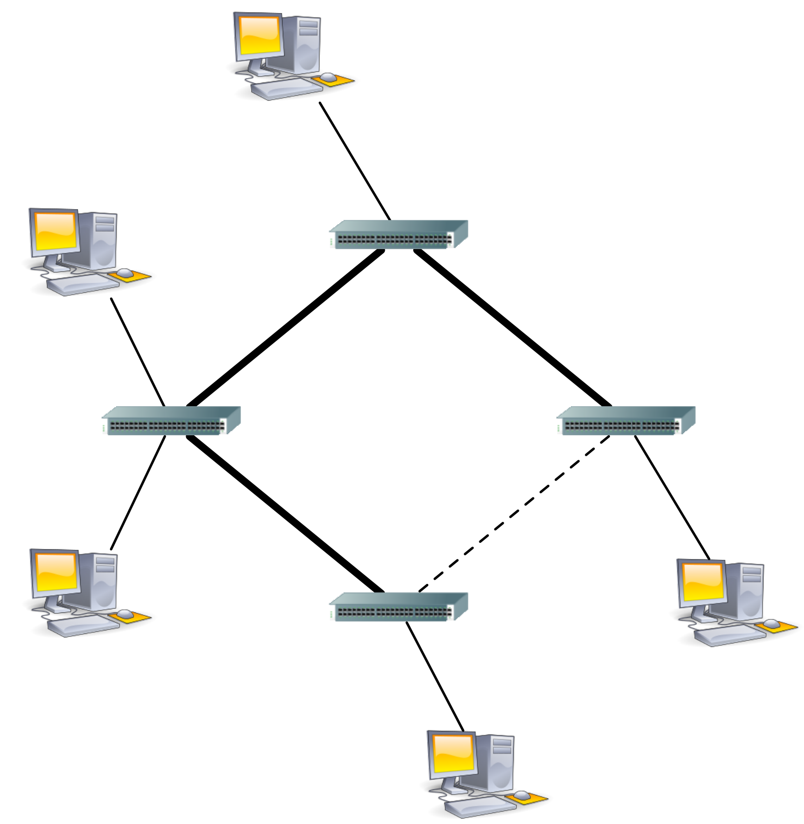
Loops resolution

- **When a loop is present in the network, due to redundancy, a broadcast frame may turn forever**

- **Solution #1: remember an ID for every frame**
  - Requires a lot of memory
    - Depends on the maximum amount of time a frame may stay in the network
  - Requires a lot of calculation
    - Check the frame ID against the whole table every time
- **Solution #2: impose a maximum distance that frames are allowed to travel**
  - Requires storing information in the frames
  - Sub-optimal
- **Solution #3: de-activate (by software) some redundant interfaces**
  - Redundancy is not effective anymore in the network
  - Interfaces may be re-activated when required (failure)
  - Which interfaces to de-activate ?

Spanning Tree Protocol - properties

- **IEEE 802.1D standard**
- **Extract from the topology a sub-spanning tree**
  - Elect the tree root
  - Select a subset of the links in order to reach every node
- **Distributed protocol**
  - Direct communication between neighbor switches
  - No multi-hop frame transmission
- **Adaptive mechanism**
  - Detect and resolve links failures



TELECOM ParisTech

General behavior

- **The protocol searches, in the network, using only local communication, the switch**
  - Who has been explicitly designated by the network administrator
    - Configuration of priorities between switches by network administrators
    - Default value: 32768
    - Higher priority = lower value (down to 0)
  - If two priorities are equal (e.g.: no explicit configuration), the lowest MAC address is chosen
    - Non-ambiguous criterion
    - Ensures that every switch in the network will consider the same node as the tree root
- **Every switch then tries to determine the best shortest path towards the tree root**
  - When two paths are available, select the shortest one
  - When two same length paths are available, select the one for whom the next hop has the highest priority / the lowest MAC address

A distributed algorithm (not IEEE 802.1d)

- **Initialization**
  - Select myself as the root of the tree
  - Distance = 0 ; father = myself

- **Every bridge periodically sends to its direct neighbors:**
  - Its priority and address
  - The priority and address of the node it considers to be the root of the tree
  - The distance that separates it from the root

- **When receiving such a message, a switch examines the contents:**
  - If the node announces a better root than the current one
    - Replace the root by the one selected by the emitting node
    - distance = distance declared by the emitting node + 1
    - Father = emitting node
  - If the root is identical and the emitting node is a better father (prio, ID or distance)
    - Replace father ID and distance with data deduced from the emitting node
  - Else: ignore the message

Performance

- **Convergence in O(network diameter) messages**
  - Permanent emission and update process to react to the network failures

- **Compromise between convergence speed and the load introduced on the network**
  - Tree convergence may be long when topologies get large and/or complex

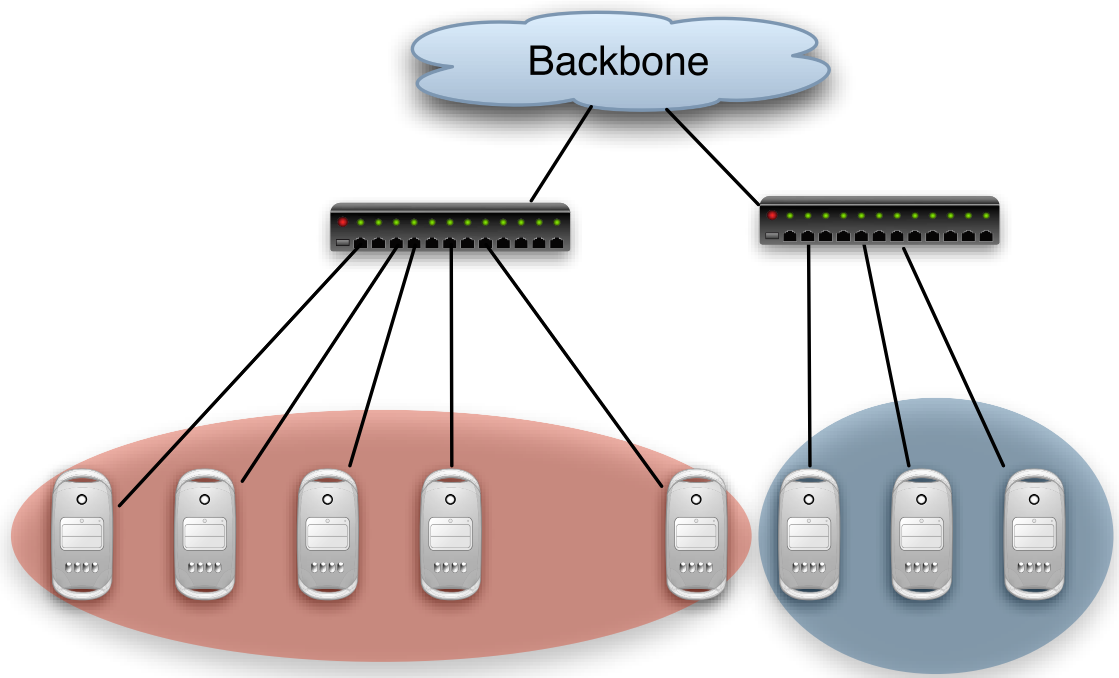
- **Links throughputs have increased, though**
  - It is possible to send updates more often than with the initial release of the STP
  - Rapid Spanning Tree (IEEE 802.1w)
  - IEEE 802.11w also proposes to pre-select backup interfaces (i.e. alternate fathers in the tree) to react quickly to the failures.

TELECOM ParisTech

Virtual LANs (VLAN)

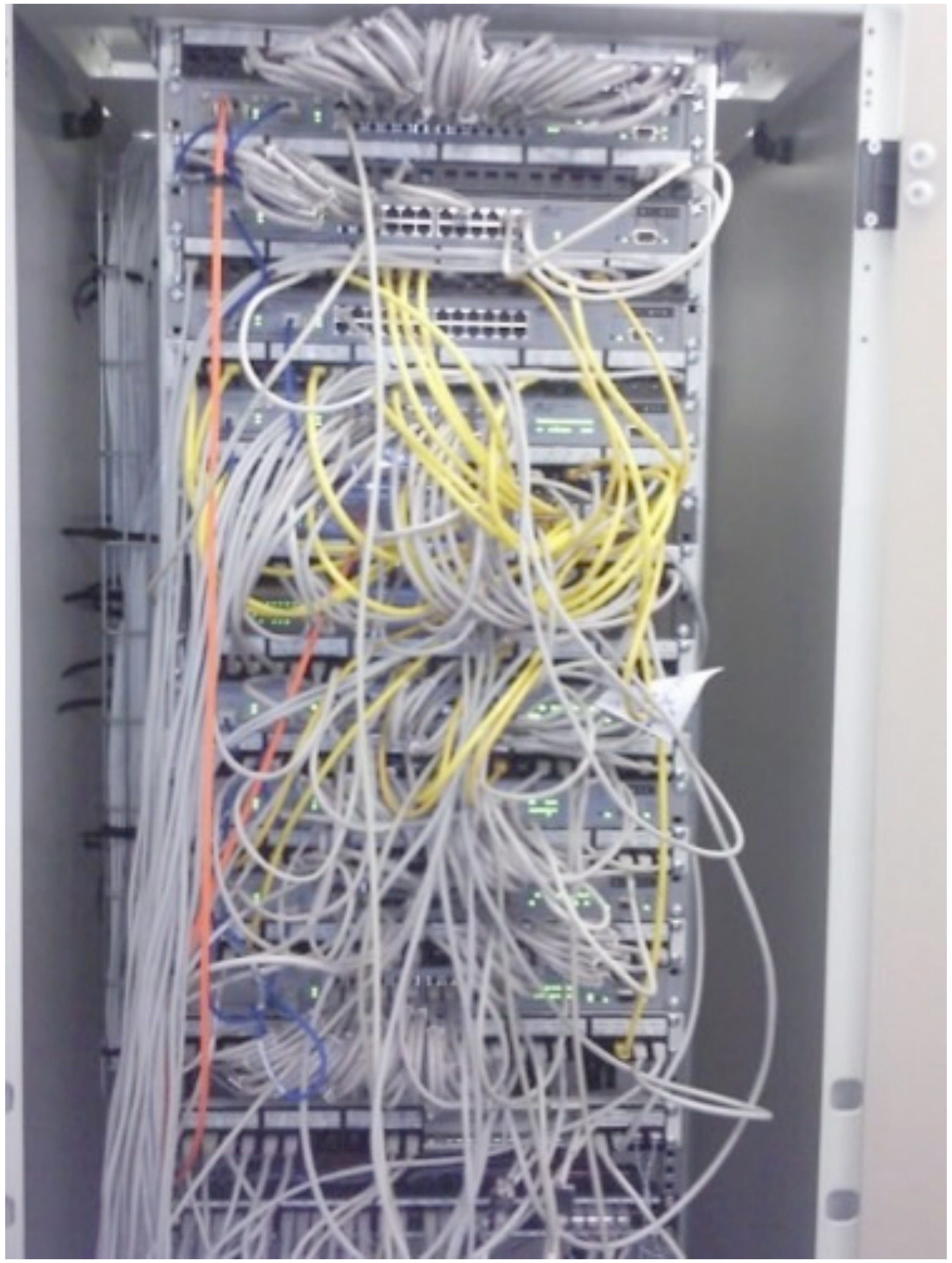
Switches-based segmentation

- **With switches:**
  - Every terminal is connected to a switch.
  - All switches are in the same room.
  - Evolution, mobility => Changing the switch the wire is attached to.



Devices-based segmentation

- **Segmentation: separate physically (cables) or logically (IP through routers) devices that do not belong to the same LAN**
- **Sometimes difficult to maintain...**
- **Network size is limited**
  - Addressing is non-hierarchical
  - IP Broadcasts reach the whole network.
- **No load balancing**



TELECOM ParisTech

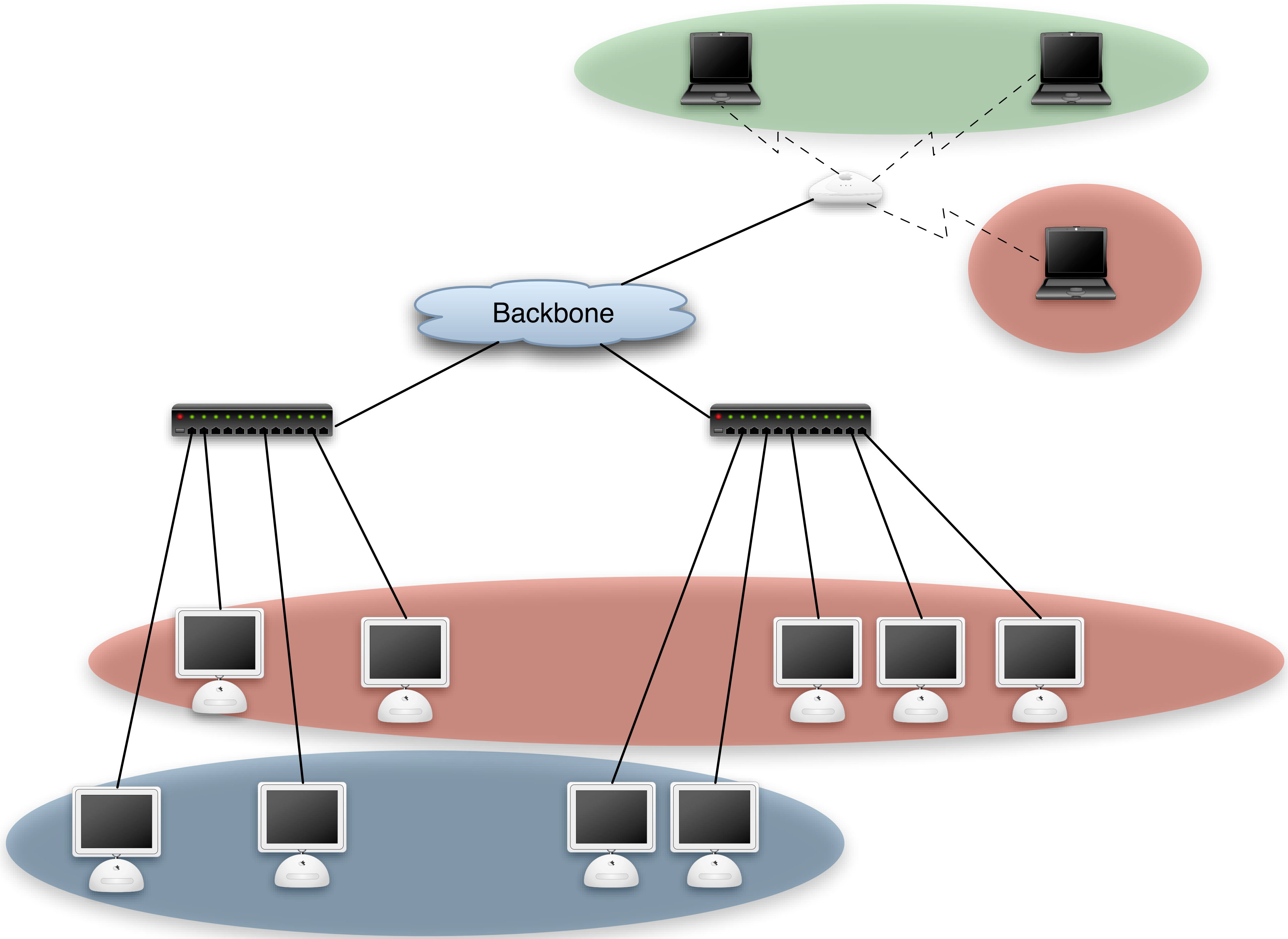
Soft Segmentation - VLANs

- **To each terminal, a VLAN ID is associated (number)**
  - Terminals sharing the same VLAN ID communicate as if they were on the same physical segment, even if they are not.
  - Machines having different VLAN IDs do not communicate directly, even if they are on the same physical segment.

- **The whole work is performed by evolved switches.**

- **Several ways to define VLANs**
  - By connection port on switches
    - Statical configuration, any mobility requires an administrator.
  - By MAC address
    - Explicit declaration of the MAC addresses, manual evolutions.
  - By IP subnetwork
    - Violates the layers-independence principle

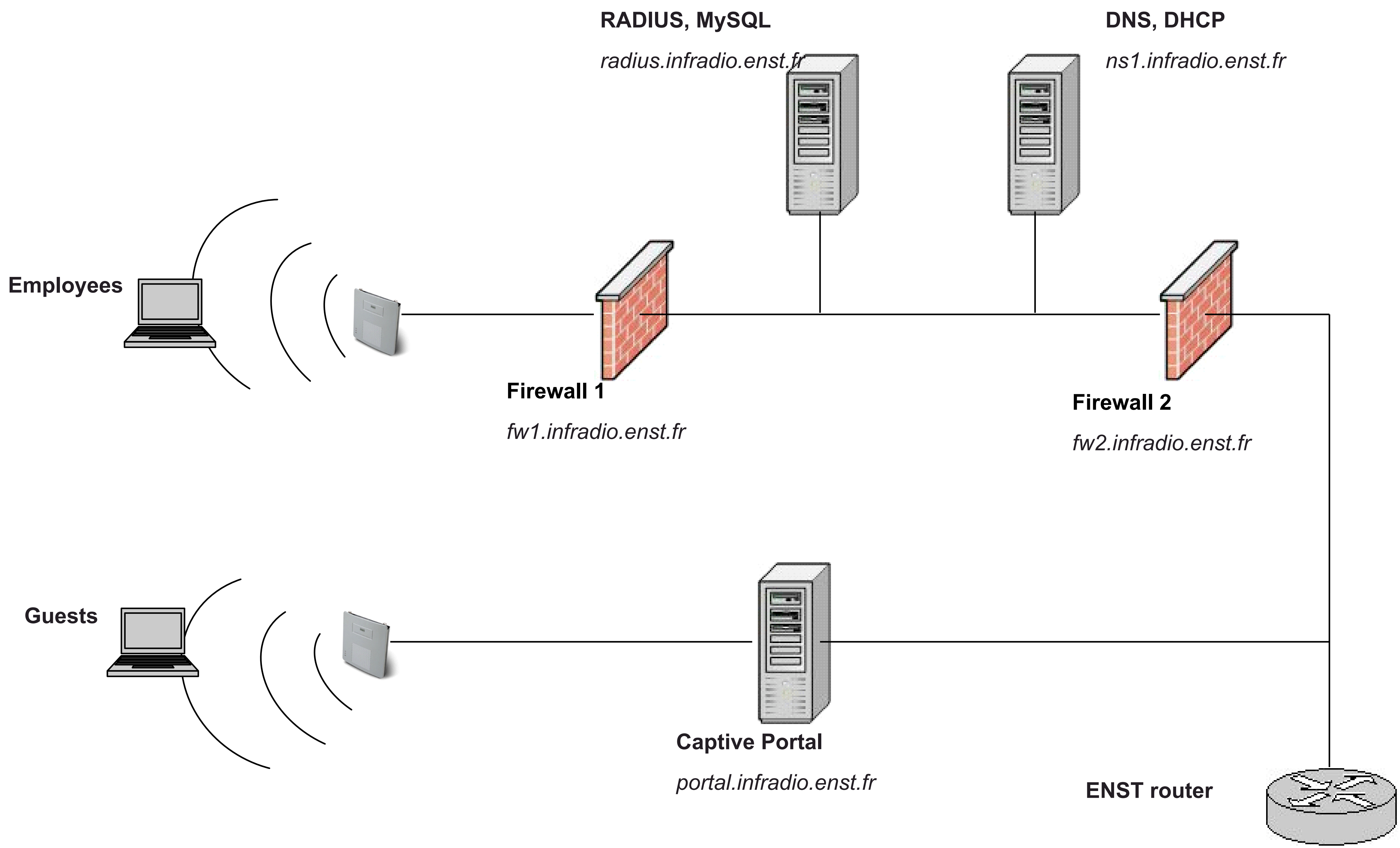
VLAN - example



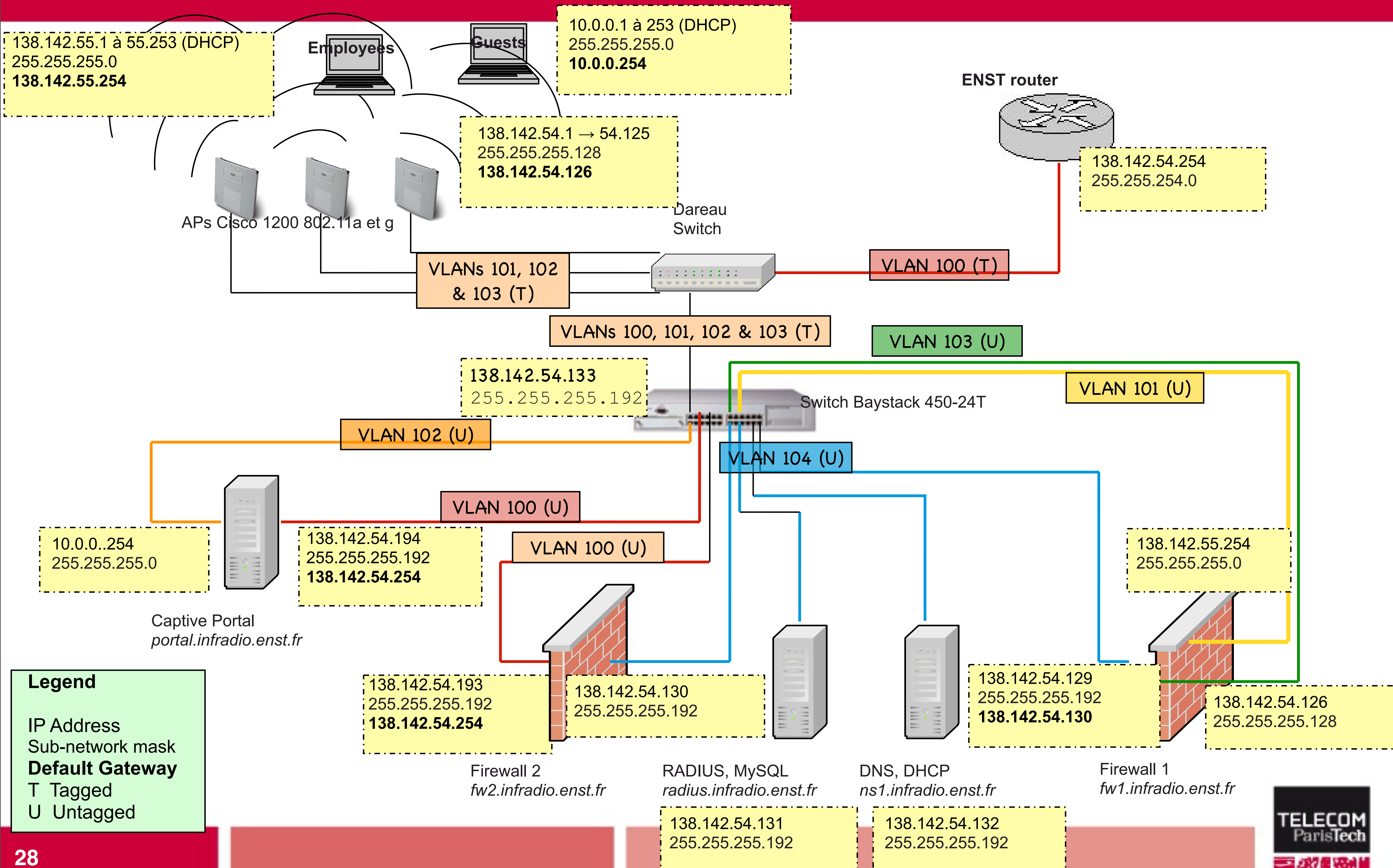
In practice

- **Not a new concept**
- **Standardization process has been long**
  - Many proprietary solutions (CISCO ISL, etc.)
- **Standard: IEEE 802.1Q**
  - Modification of the Ethernet header
  - Addition of a new field (4 bytes): VLAN ID
- **Switches can be configured manually or learn dynamically the VLANs associations.**
  - Examination of Ethernet frames
  - Learning of the correspondence between MAC address and VLAN IDs

Example: wireless network (logical vision)



In details...



138.142.55.1 à 55.253 (DHCP)
255.255.255.0
138.142.55.254
Employees
Guests
10.0.0.1 à 253 (DHCP)
255.255.255.0
10.0.0.254
ENST router
138.142.54.1 → 54.125
255.255.255.128
138.142.54.126
138.142.54.254
255.255.254.0
APs Cisco 1200 802.11a et g
Dareau Switch
VLANs 101, 102 & 103 (T)
VLAN 100 (T)
VLANs 100, 101, 102 & 103 (T)
VLAN 103 (U)
138.142.54.133
255.255.255.192
Switch Baystack 450-24T
VLAN 101 (U)
VLAN 102 (U)
VLAN 104 (U)
VLAN 100 (U)
VLAN 100 (U)
10.0.0..254
255.255.255.0
138.142.54.194
255.255.255.192
138.142.54.254
138.142.55.254
255.255.255.0
Captive Portal
portal.infradio.enst.fr
138.142.54.193
255.255.255.192
138.142.54.254
138.142.54.130
255.255.255.192
138.142.54.129
255.255.255.192
138.142.54.130
138.142.54.126
255.255.255.128
Firewall 2
fw2.infradio.enst.fr
RADIUS, MySQL
radius.infradio.enst.fr
DNS, DHCP
ns1.infradio.enst.fr
Firewall 1
fw1.infradio.enst.fr
138.142.54.131
255.255.255.192
138.142.54.132
255.255.255.192
Legend
IP Address
Sub-network mask
Default Gateway
T Tagged
U Untagged

In practice

- **List of the different ports on a switch**

Port Configuration

Port	Trunk	Status	Link	LnkTrap	Autonegotiation	Speed Duplex
1		[Enabled]	Up	[On]	[Enabled]	[100Mbs / Full]
2		[Enabled]	Up	[On]	[Enabled]	[100Mbs / Full]
3		[Enabled]	Up	[On]	[Enabled]	[100Mbs / Full]
4		[Enabled]	Down	[On]	[Enabled]	[ ]
5		[Enabled]	Up	[On]	[Enabled]	[100Mbs / Full]
6		[Enabled]	Down	[On]	[Enabled]	[ ]
7		[Enabled]	Down	[On]	[Enabled]	[ ]
8		[Enabled]	Down	[On]	[Enabled]	[ ]
9		[Enabled]	Down	[On]	[Enabled]	[ ]
10		[Enabled]	Down	[On]	[Enabled]	[ ]
11		[Enabled]	Up	[On]	[Enabled]	[100Mbs / Full]
12		[Enabled]	Down	[On]	[Enabled]	[ ]
13		[Enabled]	Up	[On]	[Enabled]	[100Mbs / Full]
14		[Enabled]	Up	[On]	[Enabled]	[100Mbs / Full]

More...

Press Ctrl-N to display choices for additional ports..
Use space bar to display choices, press <Return> or <Enter> to select choice.
Press Ctrl-R to return to previous menu. Press Ctrl-C to return to Main Menu.

TELECOM ParisTech

In practice

- **Configuration of a tagged (shared) port**

```
                    VLAN Display by Port

               Port:        [ 1  ]
               PVID:        1003
               Port Name:   Port 1
   VLANs      VLAN Name                    VLANs      VLAN Name
 ---------  ----------------             ---------  ----------------
     6      SIAV
     18     InfRadio SIAV
     1000   InfRadio DMZ
     1001   InfRadio Perm
     1002   InfRadio Invit
     1003   InfRadio Mgmt
     1004   InfRadio Test 1
     1005   InfRadio Test 2
     1006   InfRadio Test 3

Use space bar to display choices, press <Return> or <Enter> to select choice.
Press Ctrl-R to return to previous menu.  Press Ctrl-C to return to Main Menu.
```

In practice

- **Configuration / Visualization of a VLAN**

```
                         VLAN Configuration

  Create VLAN:       [ 1000 ]           VLAN Type:           [   Port-Based    ]
  Delete VLAN:       [      ]           Protocol Id (PID):   [      None       ]
  VLAN Name:         [ InfRadio DMZ ]   User-Defined PID:    [ 0x0000 ]
  Management VLAN:   [ Yes ]            VLAN State:          [     Active      ]

                          Port Membership
            1-6       7-12     13-18     19-24
          ------    ------    ------    ------


 Unit #1  T-----    ------    -U-U-U    -U-UUU




Enter VLAN Number: 1000
KEY: T = Tagged Port Member, U = Untagged Port Member, - = Not a Member of VLAN
Use space bar to display choices or enter text.
Press Ctrl-R to return to previous menu.  Press Ctrl-C to return to Main Menu.
```

Data Link Layer

Courtesy of a slight part of this course belongs D. Rossi (Telecom ParisTech)

Claude Chaudet

Data Link Layer: Sub-layers

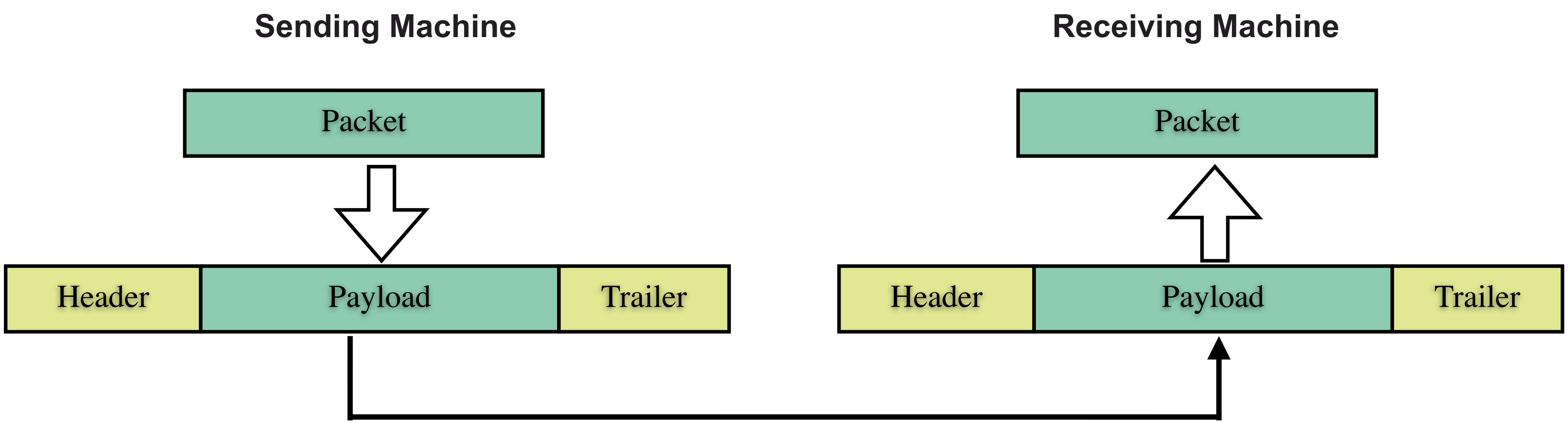
- **As the role of this layer is huge, it is often sub-divided into two sub-layers:**

- **Medium Access Control (MAC)**
  - define rules to access (and possibly share) the link resources
  - Define addresses of communicating entities

- **Logical Link Control (LLC)**
  - Provide service to network layer
  - Framing
  - Error Detection/Correction
  - Flow Control

Framing

- **Frame**
  - Sequence of bits handed to the PHY layer
  - Payload = network-layer packet
  - Prepended by a header, terminated by a trailer



- **So, at first sight it seems easy, but…**

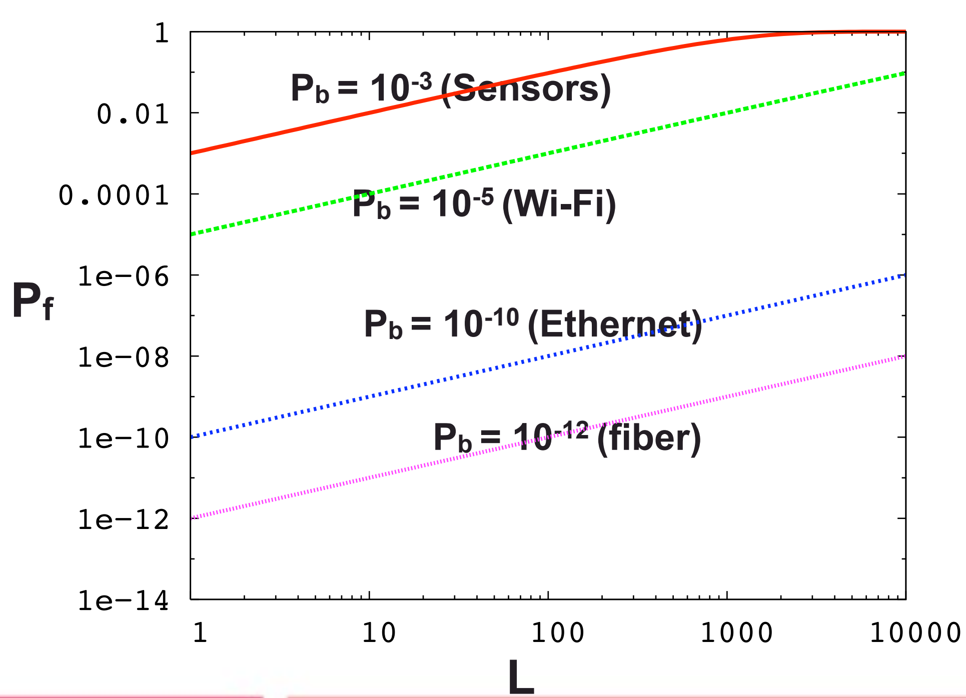
Framing

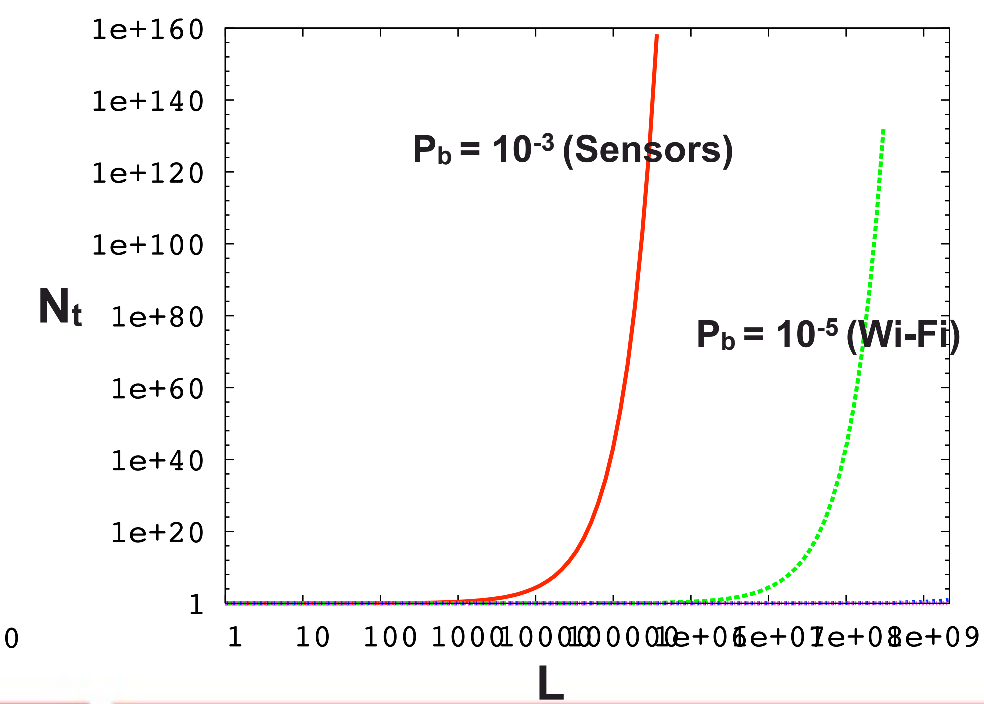
- **How to choose the frame length L?**
  - Small L: higher overhead per frame (given header length H)
  - Big L: many transmission attempts (on non-ideal channel)

$$N_{tx} = \sum_{i=0}^{+\infty} i \cdot P_f^{i-1} \cdot (1 - P_f) = (1 - P_f) \cdot \sum_{i=0}^{+\infty} i \cdot P_f^{i-1} = \frac{1}{1 - P_f}$$

  - with P_b= bit error probability

    P_f = frame error probability = $1-(1-P_b)^L$





TELECOM ParisTech

Framing with limited size

- **Packets are split into multiple frames**
  - Maximum frame length depends on the medium: MTU (Maximum Transfer Unit)

- **Usually, a flow is cut into pieces at higher layers to optimize performance**
  - Remember for later in the course

Framing - Frames separation

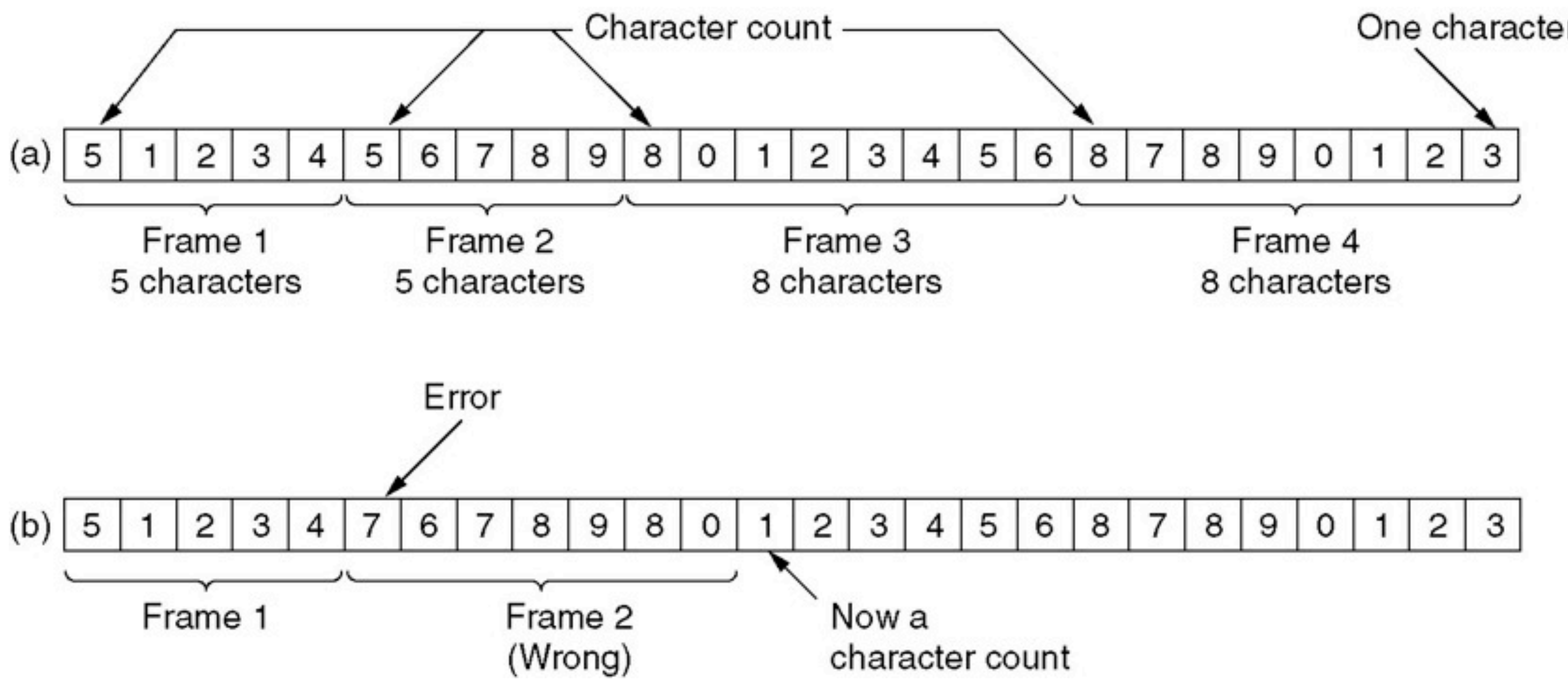
- **How to differentiate, on a medium, successive frames?**

- **Use timing to detect start and end of frames?**
  - Frames may have variable length
  - Stations may loose clock synchronization

- **Other methods?**
  - Character count
  - Flag-bytes with byte-stuffing
  - Start and end flags with bit-stuffing
  - Physical layer code violation

Framing: Character Count

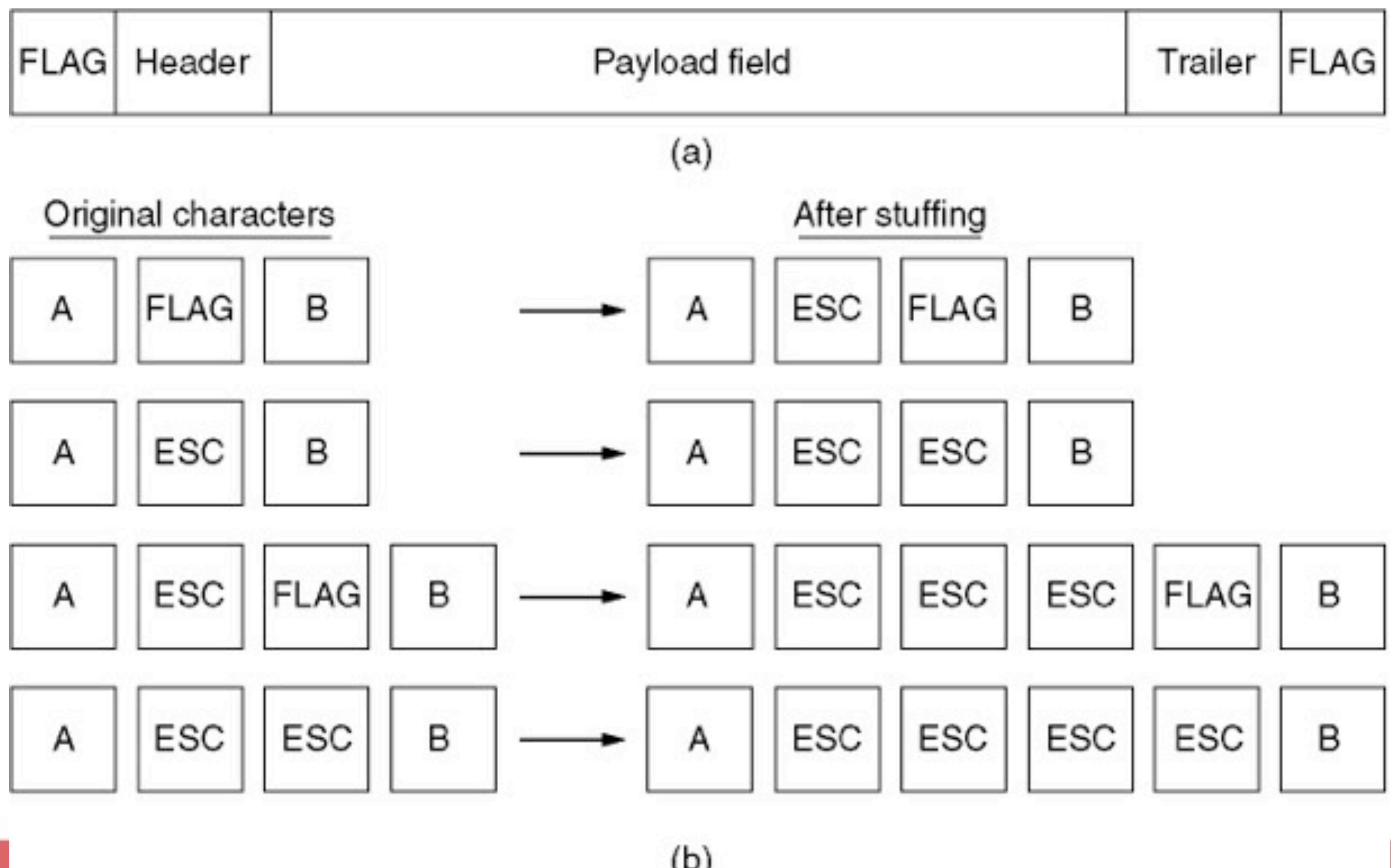
- Character count means relying on the size field of the frame header…. but in case of error!?



- Typically in combination with one of next methods

Framing: Flag-Bytes

- **Use a *flag* sequence: 01111110**
  - if data contains *flag* => *escape*
  - if data contains *escape* => *escape* again!
  - disadvantage: works only for 8-bit codes



Framing: Bit-stuffing

- **Use (the same)** ***flag*** **sequence: 01111110**
  - 01111110 in data => 011111010
  - Receiver de-stuff the 0 after 5 bits set to 1
  - 01111100 in data => 011111000, no problem

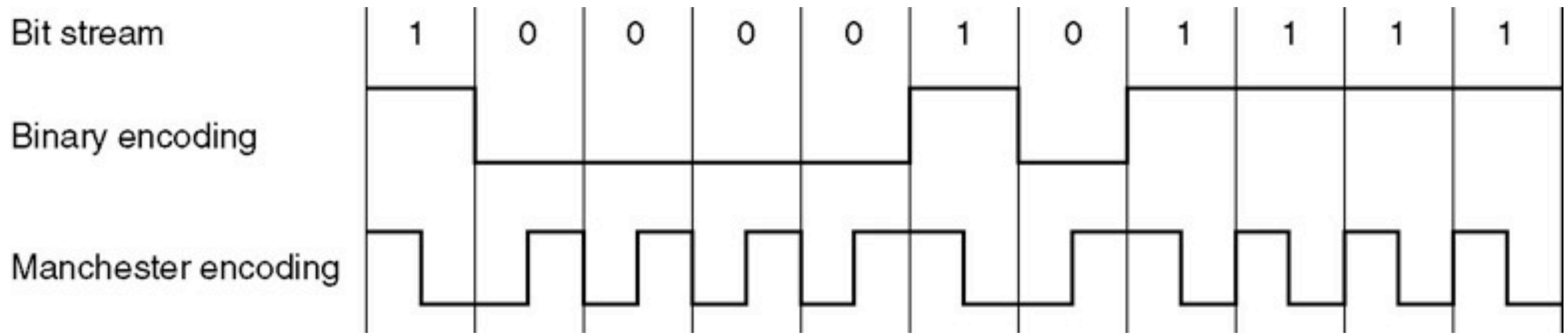
Original: 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Stuffed: 0 1 1 0 1 1 1 1 1 **0** 1 1 1 1 1 **0** 1 1 1 1 1 **0** 1 0 0 1 0

Stuffed bits

Destuffed: 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Framing: Physical layer coding



- **Exploit the fact that PHY coding:**
  - adds redundancy,
    - i.e., 1 bit represented with usually more than 1 symbol
  - the signals are usually DC balanced (sum of the volt.second = 0)
    - transition in the middle
  - So, high-high and low-low codes are not used for data
    - use them as delimiters!

Error Handling

- **Now we can detect, receive, decode a frame:**
  - Has the frame been received correctly?
  - Has the frame been received at all?

- **Now, let's focus on the first kind of errors**
  - Need ways to detect (and possibly prevent) errors:
  - Notice that is possible that errors go undetected
  - Correcting error is more costly than detecting them (need more bits and more processing)

- **Transmission errors:**
  - Rare in fiber, but rather common in wireless medium
  - Errors in radio environment tend to come in bursts
    - Advantage: affect less frames
    - Disadvantage: harder to correct

Error Handling: Detection/Correction

- **Handling transmission errors:**
  - Error detection
  - Forward error correction (FEC)

- **Information theory**
  - Things get complicated real quick
  - Take simple examples for illustration of the concepts
    - 1-bit Detection: parity scheme
    - 1-bit Correction: two-dimensional parity
  - In practice, more sophisticated codes are used

- **Classical algorithms**
  - Detection
    - Checksum, Cyclic redundancy check
  - Correction
    - Reed Solomon codes, How to correct error bursts

Error Detection: 1-bit parity

- **Parity scheme: simplest form of error detection**
  - Suppose you want to send a d-bits long data word D
  - `101100`

- **At sender side, add a parity bit and transmit (d+1) bits:**
  - Odd parity: the number of 1s in the (d+1) bits is odd
  - Even parity: the number of 1s in the (d+1) bits is even
  - `1011001`

- **At receiver side, count the number of 1s:**
  - Suppose an odd number of 1s is found with an even parity scheme
  - Receiver can conclude that at least one error happened
  - More precisely, any odd number of errors can be detected
    - `1010001, 1100001`
  - An even number of errors occurring in burst would go unnoticed
    - `1000001, 1101001`

Error Correction: 2D-parity

- **Two-dimensional parity**
  - Rearrange data as a matrix
  - Add parity for each row, col

101100 =>

1	0	1	1
1	0	0	0
1	1	0	

=> 10110010110

- **Able to correct single bit errors (also on parity bits)**

10110010110 => 10100010110 =>

1	0	1	1
0	0	0	0
1	1	0	

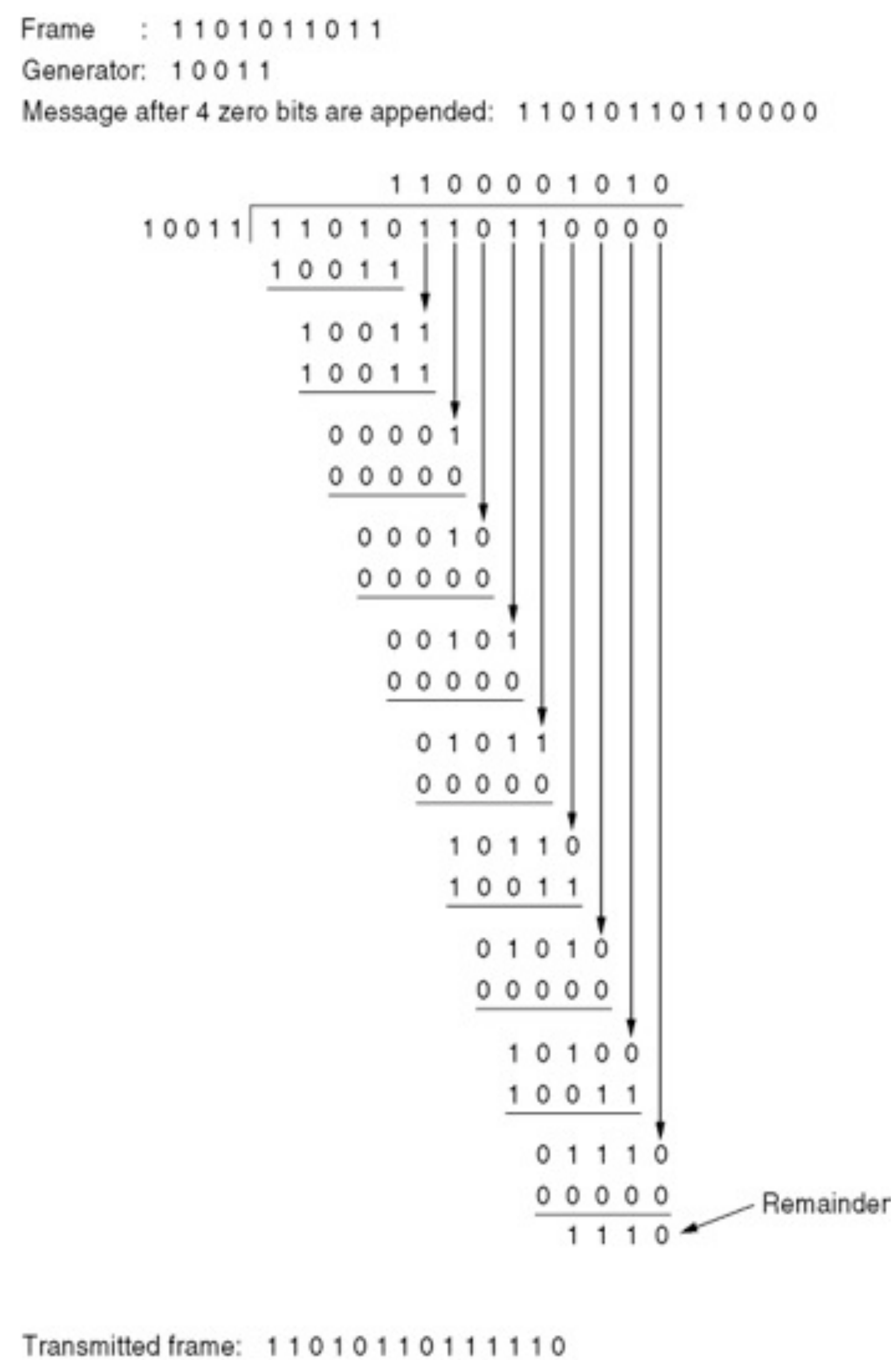
- **Able to detect (but not correct) two-bit errors**

10110010110 => 10101010110 =>

1	0	1	1
0	1	0	0
1	1	0	

Error Detection: Quick remarks

- **Checksum**
  - Quick but not very reliable,
  - Used at the transport layer (thus, end-to-end) which is implemented in software
- **Cyclic redundancy check**
  - Implemented in hardware as shift register
    - Given by standards, e.g. IEEE 802: $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$
    - Detects all burst of 32 bits or less, and any burst affecting an odd-number of bits
- **Often tested assuming random input; however, a MAC address won't change randomly!**
  - So, performance are different from expected!
  - What to do when Checksum and CRC disagree?



Error Correction: Quick remarks

- **How to protect from bursts of error?**
  - Protect each codeword with FEC for single bit errors
  - Apply a columnar trick
  - Rearrange transmission order
  - Rearrange bit order at reception
  - In case of error burst, errors affect different codewords
  - With single bit error that FEC is able to recover
  - Hamming code in the ex.

Char.	ASCII	Check bits
H	1001000	00110010000
a	1100001	10111001001
m	1101101	11101010101
m	1101101	11101010101
i	1101001	01101011001
n	1101110	01101010110
g	1100111	01111001111
	0100000	10011000000
c	1100011	11111000011
o	1101111	10101011111
d	1100100	11111001100
e	1100101	00111000101

Order of bit transmission

- **FEC in practice**
  - Reed-Solomon codes, used e.g., in CDs and xDSL

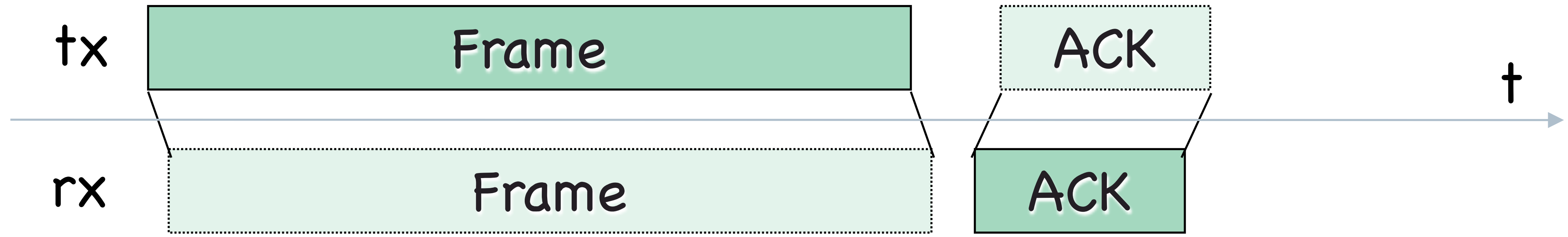
Error Handling

- **Now we can detect, receive, decode a frame:**
  - Has the frame been received correctly?
  - Has the frame been received at all?

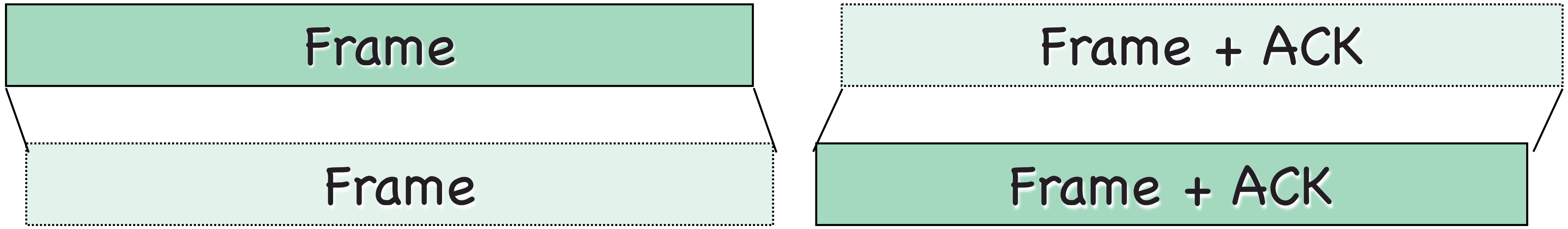
- **Now, let's focus on the both kinds of errors**
  - Receiver provides feedback to the transmitter
    - Positive feedback: frame correctly received
    - Negative feedback: frame with non-correctable errors
  - What if transmitted frame is lost?
    - Implicit negative feedback: after a timer, retransmit
  - What if feedback is lost?
    - Implicit negative feedback (again)

- **Strategy:**
  - Reliable channels: detect and retransmit
  - Unreliable channels: correct rather than retransmit

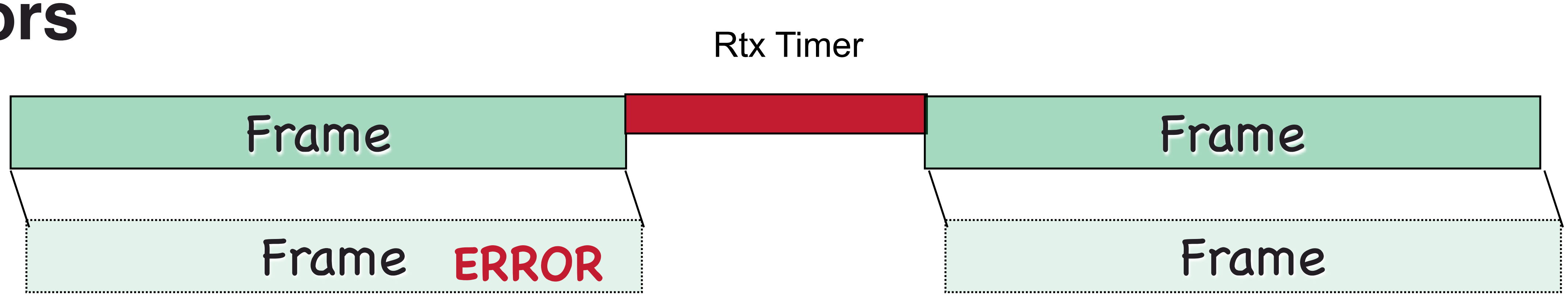
- **Correct transmission**



- **Piggybacking**



- **Errors**



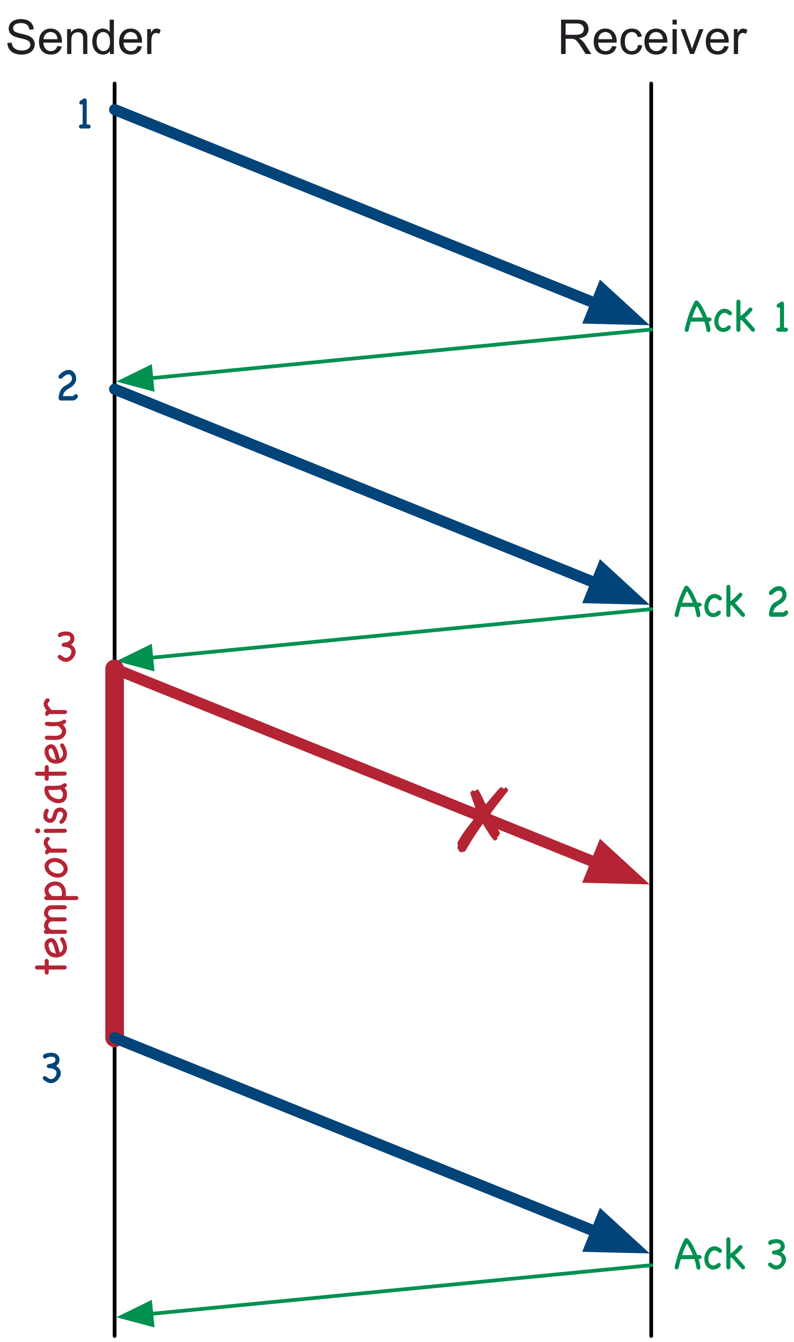
Regulating data flow

- **Aim:**
  - Prevent slow receivers to be swamped by fast senders, avoiding resource waste
  - Recover from detected errors that the FEC (if in use) is unable to correct

- **Automatic Repeat Request (ARQ)**
  - Stop-and-wait
  - Go-back-n
  - Selective repeat

ARQ : Stop and Wait

- **Acknowledge every frame**
  - need to wait for an ACK prior to send another frame
  - Needs at least 1-bit sequence no. (otherwise, what could happen?)
- **What if ACK is lost?**
  - Retransmission => duplicated frame
  - Discard duplicates (correctness preserved)
- **Not very efficient**
  - Acknowledgement
  - Time between frames
- **Buffer**
  - During wait for ACK, tx is idle, buffer may grow and packets get dropped
  - At receiver, no buffer space needed



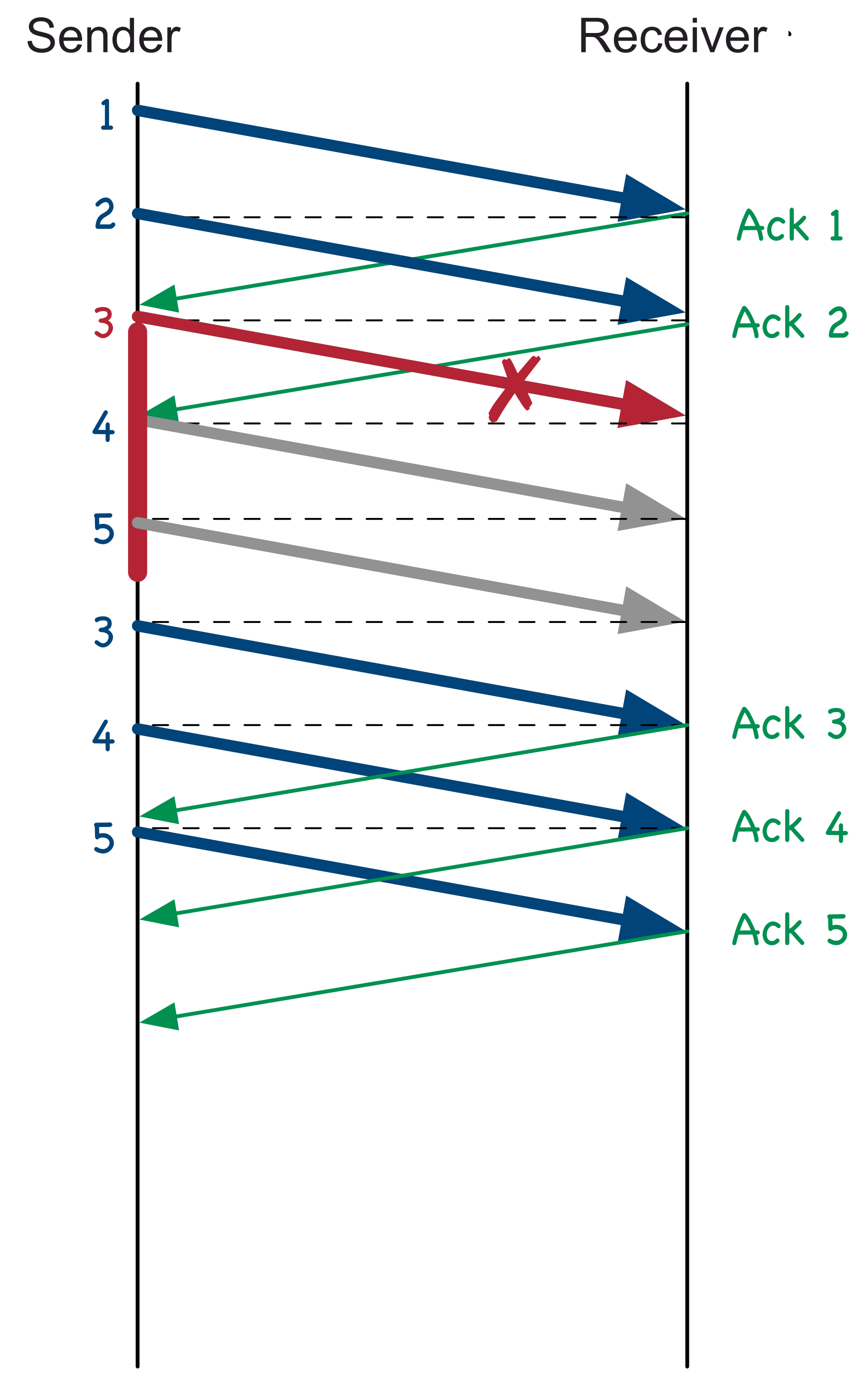
Pipelining

- **Stop-and-wait efficiency?**
  - Example: Satellite link, 50 kb/s 500-ms round trip delay
    - T=0 ms, sender start sending a frame of 1000 bits
    - T=20 ms, sender finished sending the frame
    - T=270 ms, frame entirely arrived at the receiver
    - T>520 ms, acknowledgement at the sender => Efficiency = 20/520 = 4%
- **Transmit more frames before blocking!**
  - In the example above, sender may transmit 26 frames before the first frame gets acknowledged !
  - This technique is known as *pipelining*
  - Necessary whenever bandwidth x round-trip delay is large
  - **Bandwidth x round-trip = capacity of the pipe**
- **Pipelining can raise serious issues on lossy channels!**
  - Two techniques: Go-back-N and Selective Repeat
  - Both techniques use sender window for pipelining

TELECOM ParisTech

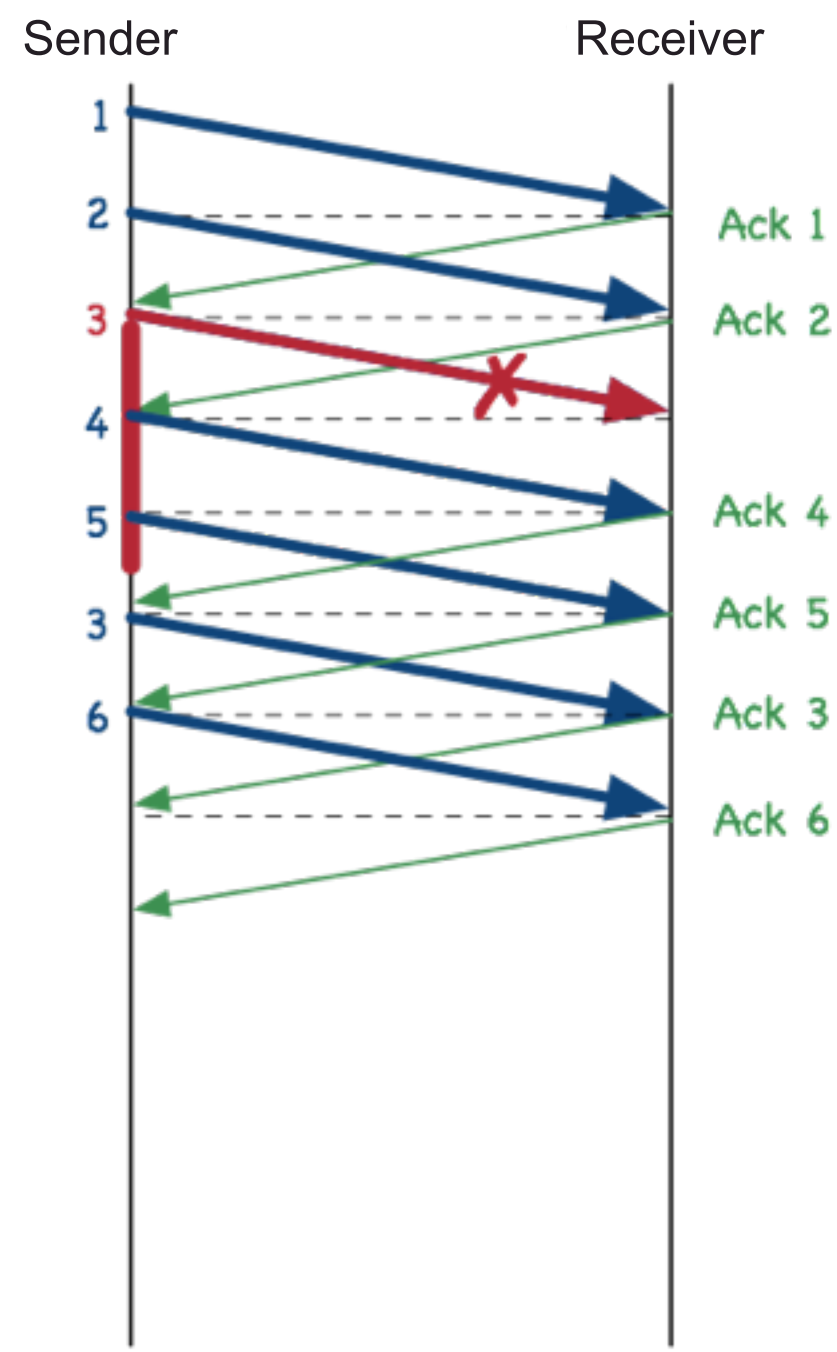
ARQ: Go-Back-N

- **Go-back-N**
  - Sliding window pipelining, receiver window size = 1
  - Doesn't have to wait for ack to transmit next frame
- **When loss happens**
  - Receiver examines seqno: in case of loss, nothing gets acknowledged anymore
  - After timeout, sender retransmit everything since the last acknowledged frame
- **Therefore**
  - Memory efficient, simple receiver
  - Many retransmissions and duplicated frames
  - Can waste a lot of bandwidth, works well when errors are extremely rare



ARQ: Selective Repeat

- **Selective repeat**
  - Pipelining, receiver window size > 1
  - Less retransmissions: only lost frames
  - Need a bigger receiver memory
- **Peformance tradeoff**
  - More bandwidth efficient, but more complex receiver
  - Normally coupled to the use of negative acknowledgement
- **Need for flow control**
  - Carefully dimensioning the sender window to avoid swamping the receiver



Services provided to the network layer

- **Increasing level of reliability**
  - Unacknowledged Connectionless service
  - Acknowledged Connectionless service
  - Acknowledged Connection-oriented service

- **Transport layer (TCP) does end-to-end reliability, LLC offers single link reliability**
  - is this redundancy really necessary?
  - on faulty links, local retransmission of a frame may avoid end-to-end retransmission of a segment

Types of services

- **Unacknowledged Connectionless service**
  - Appropriate for very reliable channels, such as optical fiber;
  - Appropriate for any type of traffic where a bad packet is better than a late packet (e.g., voice)

- **Acknowledged Connectionless service**
  - An upper layer packet may be broken in several (say, N) frames.
  - The loss of a single frame entails the retransmission of all N frames unless link-layer acknowledgement is used
  - Acknowledgment loss may imply data to be received more than once
  - Providing acknowledgment at data link layer is an optimization, never a requirement

- **Acknowledged Connection-oriented service**
  - Frames are numbered to guarantees that are received exactly once
  - Need to handle signaling of connection startup and tear-down

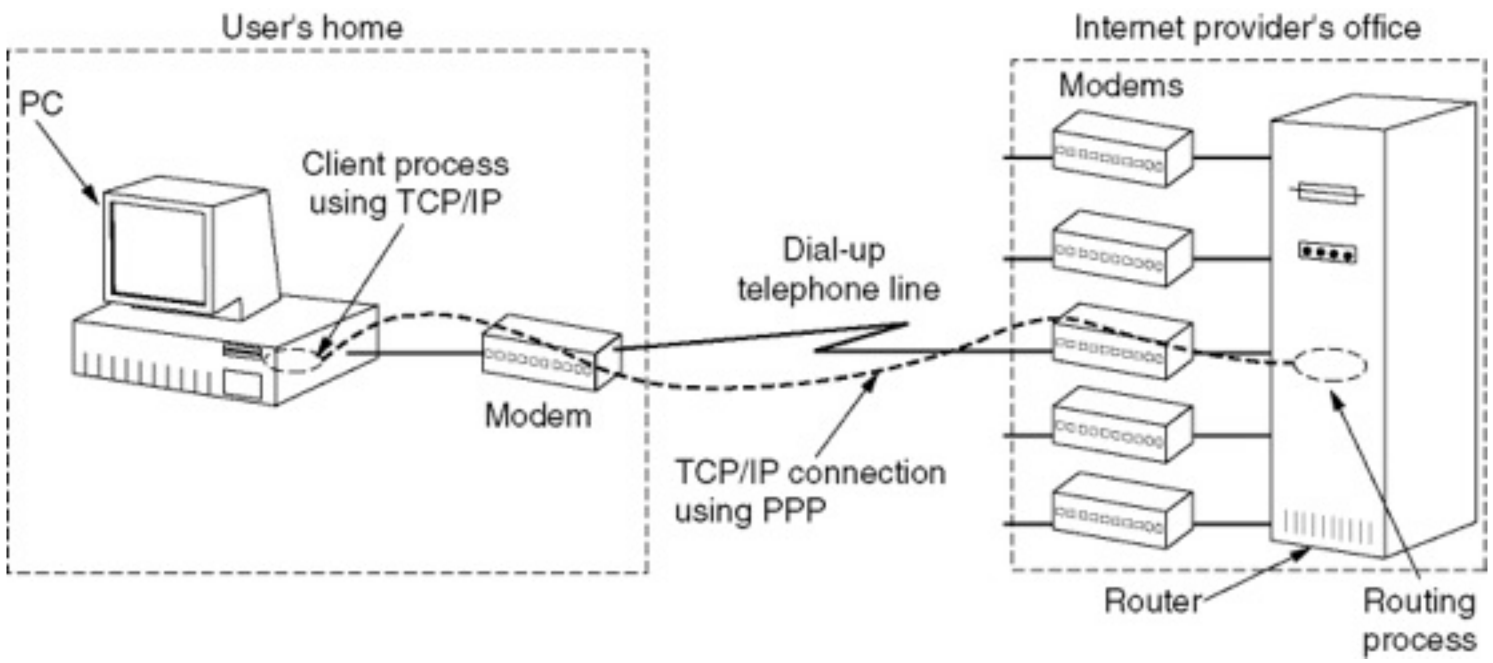
Data Link Layer: Protocols

- **Many examples**
  - HDLC and variants
    - based on IBM's SDLC protocol
    - basis for Point-to-Point Protocol (PPP)
    - LAPB for X.25
    - LAPM for V.42
    - LAPD for ISDN
    - LAPF for FrameRelay
  - Ethernet (IEEE 802.3)
  - WiFi (IEEE 802.11)
  - WiMAX (IEEE 802.16)

Example: PPP

Point-to-Point Protocol (PPP)

- **Widely used in the Internet**
  - Router to router, or home user to ISP



- **Properties**
  - Framing and error detection provided by PPP
  - Link control protocol (LCP)
    - bringing up/down lines, negotiating options,
    - supports asynchronous/synchronous lines and bit/byte encoding
  - Network control protocol (NCP)
    - Negociate network layer option and parameters (e.g., IP address) independently from the network protocol

Point-to-Point Protocol (PPP)

Framing

- Byte-stuffing / Address is constant / Unnumbered Unreliable by default
- Protocol defines type of Payload (LCP, NCP, IP, IPX, AppleTalk…)
- Payload size defaults 1500 / Possible padding / Polynomial checksum

Bytes	1	1	1	1 or 2	Variable	2 or 4	1
	Flag 01111110	Address 11111111	Control 00000011	Protocol	Payload	Checksum	Flag 01111110

Conclusion

- **Many different functions**
  - Framing, error handling, flow control

- **Some end-to-end features are replicated locally**
  - Flow control, error handling

- **Design choices depends on channel properties**
  - Large bandwidth•delay product: pipeline for efficient utilization
  - Wired: error detection more efficient than correction
  - Wireless: acknowledged service with forward error correction