

# Introduction to networks and protocols

*Ass. Prof. J.Y. Tigli,*  
*Polytech Nice Sophia,*  
**University of Nice Sophia Antipolis**

*Reference : Dr. John Sum*

Institute of Electronic Commerce

National Chung Hsing University

John Sum Homepage ([digi-pack.io](http://digi-pack.io))

Jean-Yves Tigli - [Jean-Yves.Tigli@univ-cotedazur.fr](mailto:Jean-Yves.Tigli@univ-cotedazur.fr)

9/26/2022

# Network, Internet, Web and Services

*Ex. Webservices*

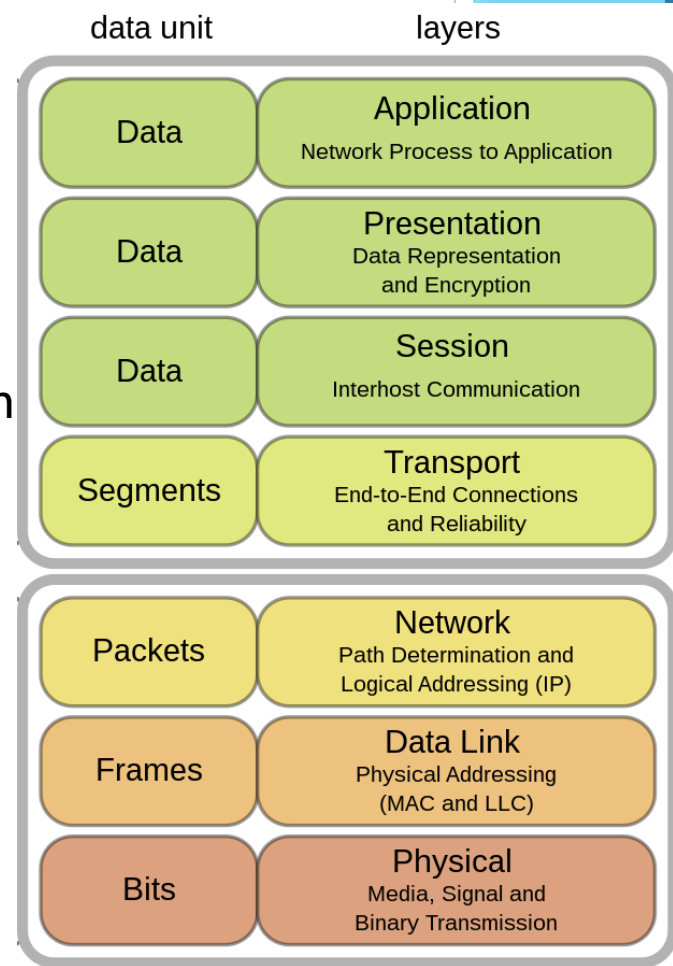
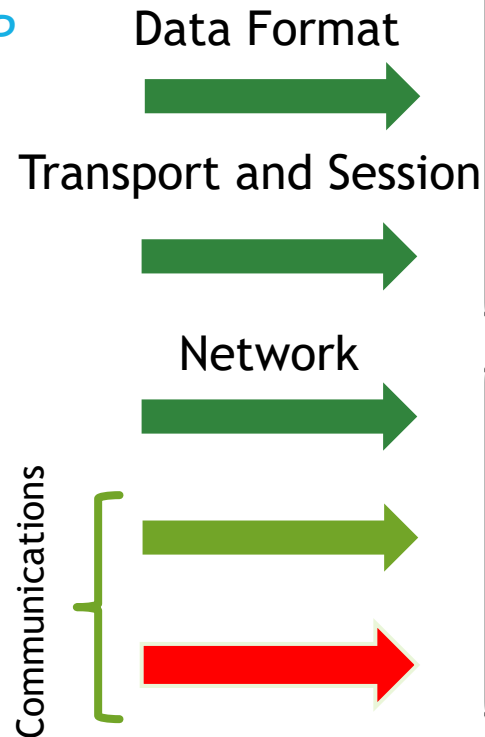
*Ex. HTML, XML, SOAP*

*Ex. HTTP (Web)*

*Ex. TCP*

*Ex. Internet Protocol*

*Ex. Ethernet / Wifi*



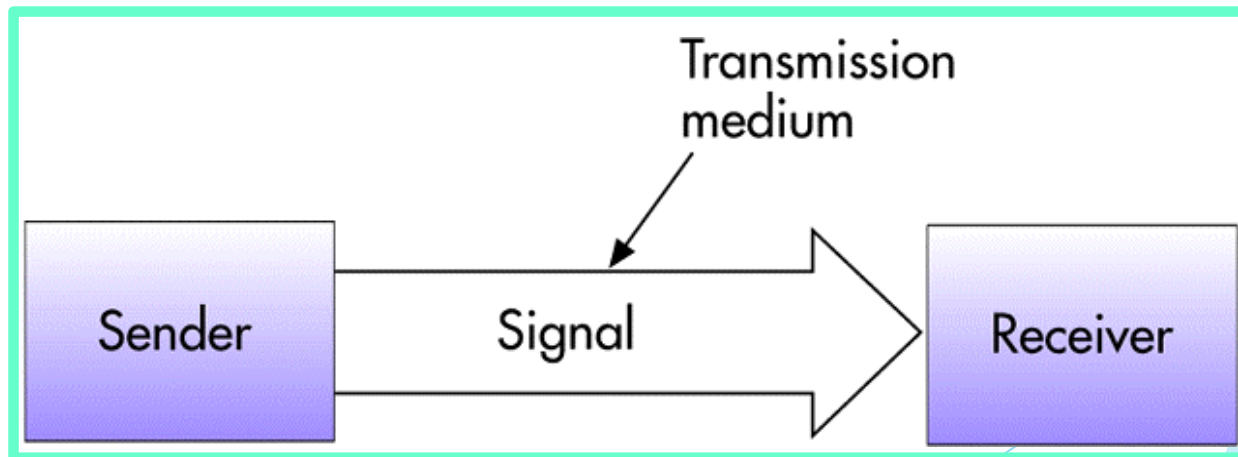
*Ex. radio, electrical cable, light, infrared,*

# communications

First step for all ... even for humans

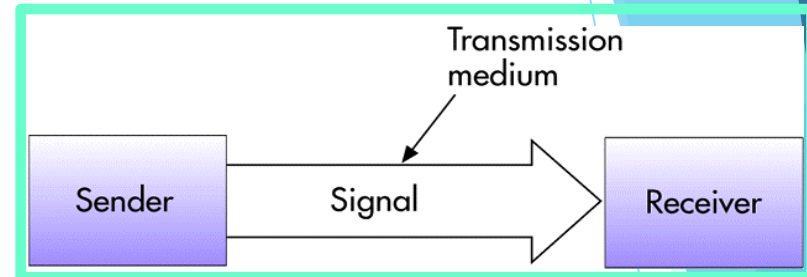
# Communications

- ▶ Communications
- ▶ The message (data and information) is communicated via the signal. The transmission medium “carries” the signal.

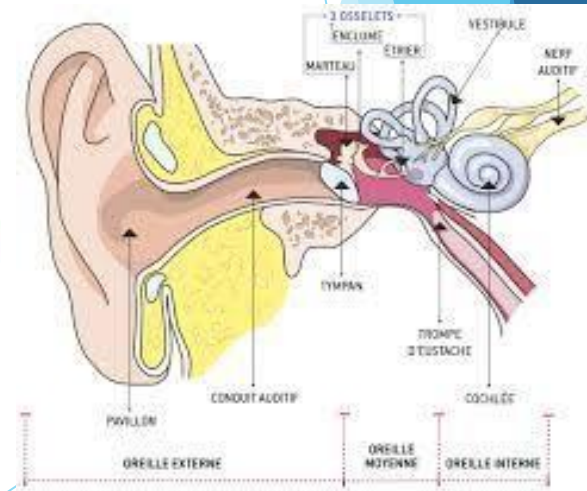
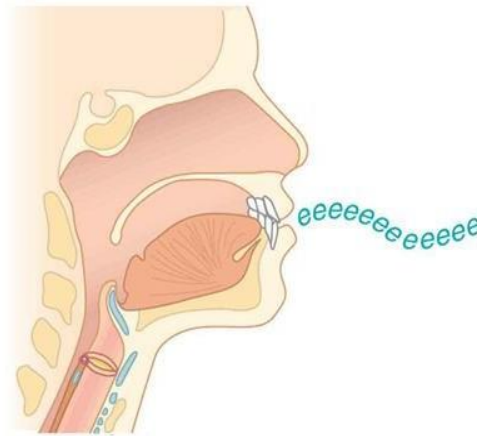
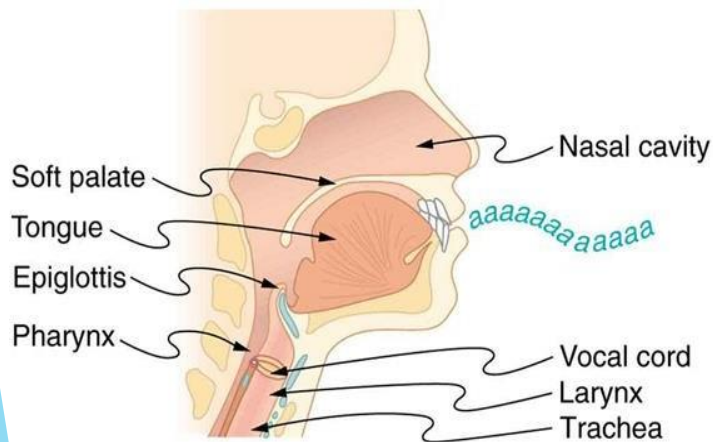


# Analog Communication

## ► Analog signal

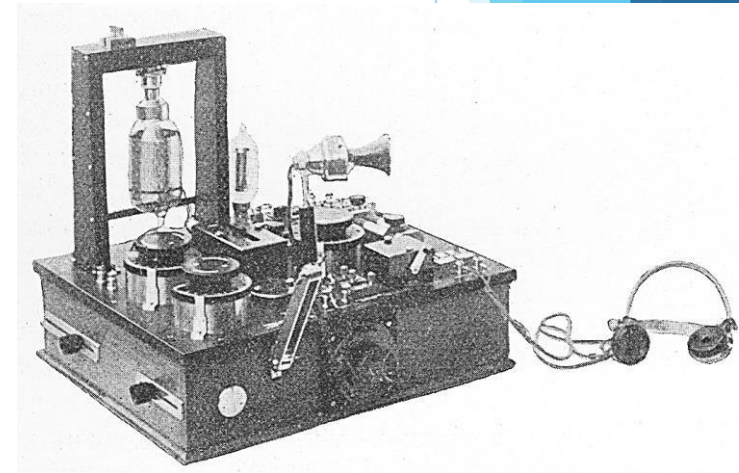
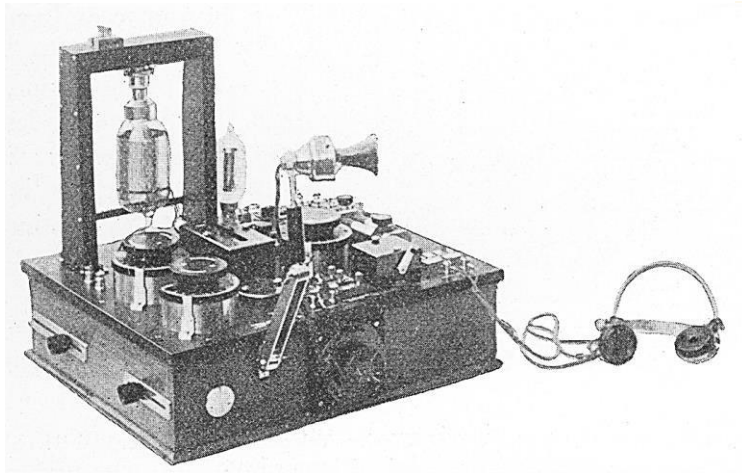
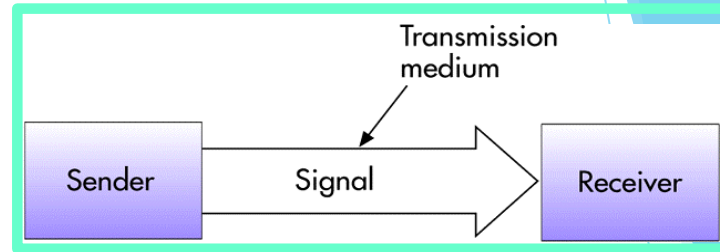


## ► Air pressure variation = sound waves



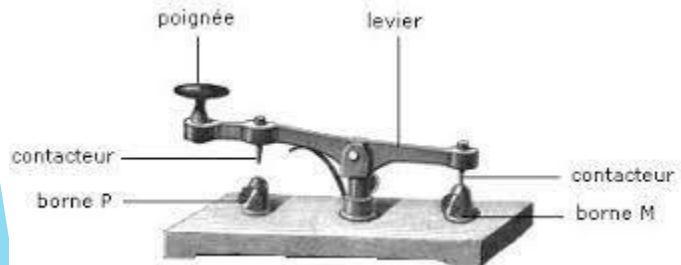
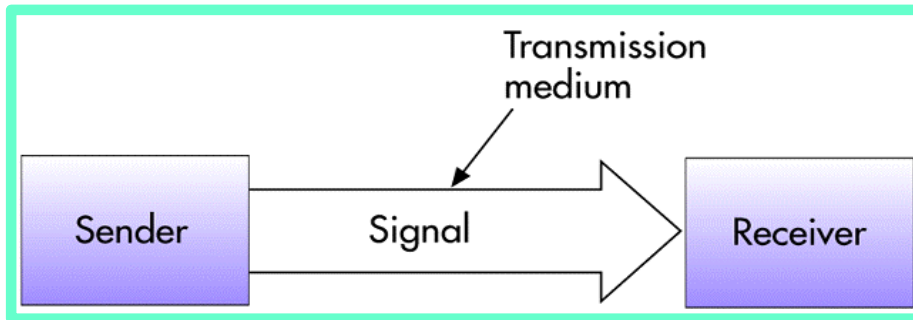
# Analog Communication

## ► Phone





# Discrete Communication

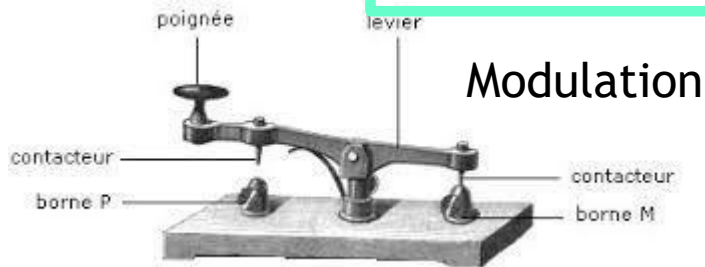
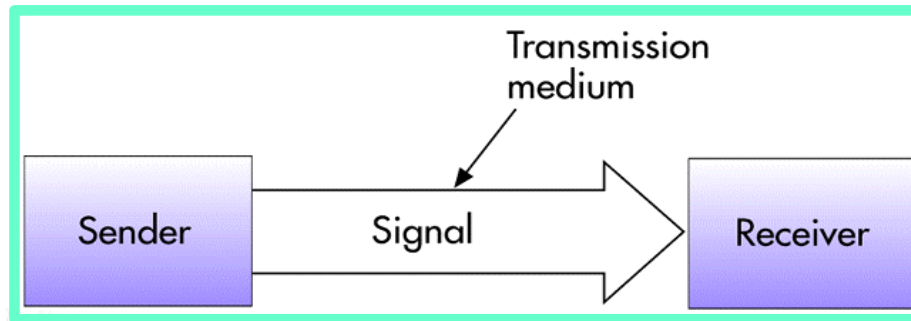


## Code morse international

1. Un tiret est égal à trois points.
2. L'espacement entre deux éléments d'une même lettre est égal à un point
3. L'espacement entre deux lettres est égal à trois points.
4. L'espacement entre deux mots est égal à sept points.

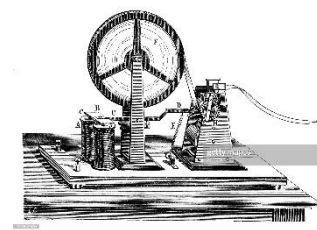
A	• —	U	• • —
B	— • •	V	• • — —
C	— • — •	W	• — — —
D	— • •	X	— • • —
E	•	Y	— • — —
F	• • — •	Z	— — • •
G	— — • •		
H	• • • •		
I	• •		
J	• — — —		
K	— • —		
L	• — • •		
M	— — •		
N	— •		
O	— — —		
P	• — — •		
Q	— — • —		
R	• — • •		
S	• • •		
T	—		
		1	• — — —
		2	• • — —
		3	• • • —
		4	• • • •
		5	• • • •
		6	— • • •
		7	— • • •
		8	— • • •
		9	— • • •
		0	— • • •

# Discrete Communication



Modulation

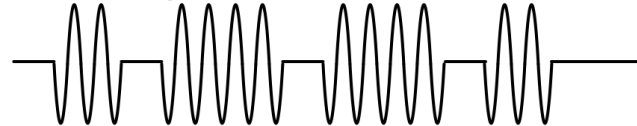
Demodulation



Bits: dots and dashes



Transmitted Signal



Bits: dots and dashes



## Code morse international

1. Un tiret est égal à trois points.
2. L'espacement entre deux éléments d'une même lettre est égal à un point.
3. L'espacement entre deux lettres est égal à trois points.
4. L'espacement entre deux mots est égal à sept points.

A	••• —
B	— ••• ••
C	— •• — ••
D	— •• ••
E	•••
F	••• — ••
G	— •• — ••
H	••• •• — ••
I	••• ••
J	— •• — •• — ••
K	— •• — ••
L	•• — •• — ••
M	— •• — ••
N	— •• ••
O	— •• — •• — ••
P	•• — •• — ••
Q	— •• — •• — ••
R	•• — •• — ••
S	••• — ••
T	— •• — ••

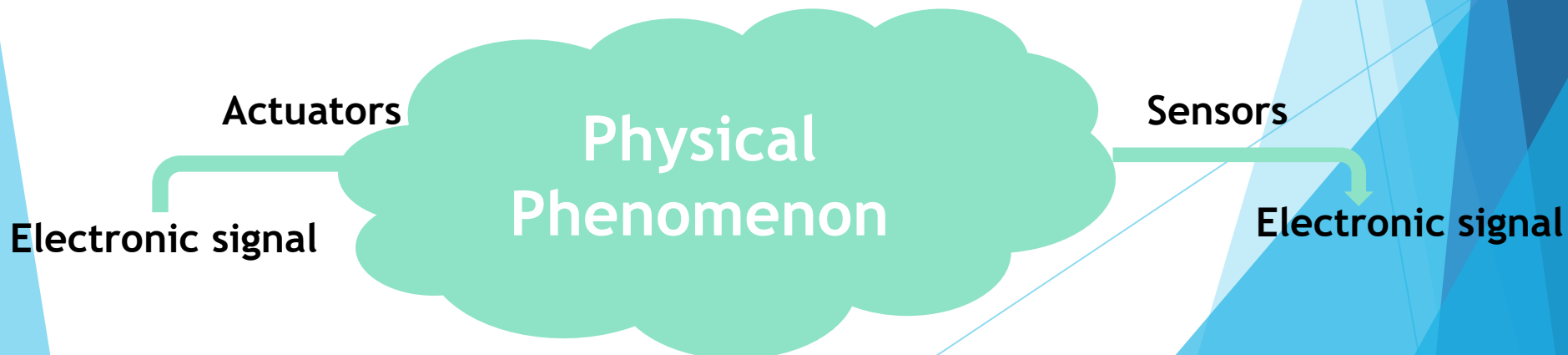
U	••• —
V	••• — ••
W	— •• — ••
X	— •• — •• — ••
Y	— •• — •• — ••
Z	— •• — •• — ••

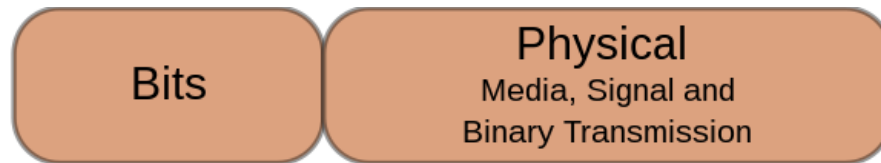
1	— •• — •• — ••
2	•• — •• — ••
3	•• — •• — ••
4	•• — •• — ••
5	•• — •• — ••
6	•• — •• — ••
7	•• — •• — ••
8	•• — •• — ••
9	•• — •• — ••
0	•• — •• — ••



# So telecommunications are ...

- ▶ Telecommunications
- ▶ The electronic transmission of signals for communications, including such means as:
- ▶ Telephone
- ▶ Radio
- ▶ Television

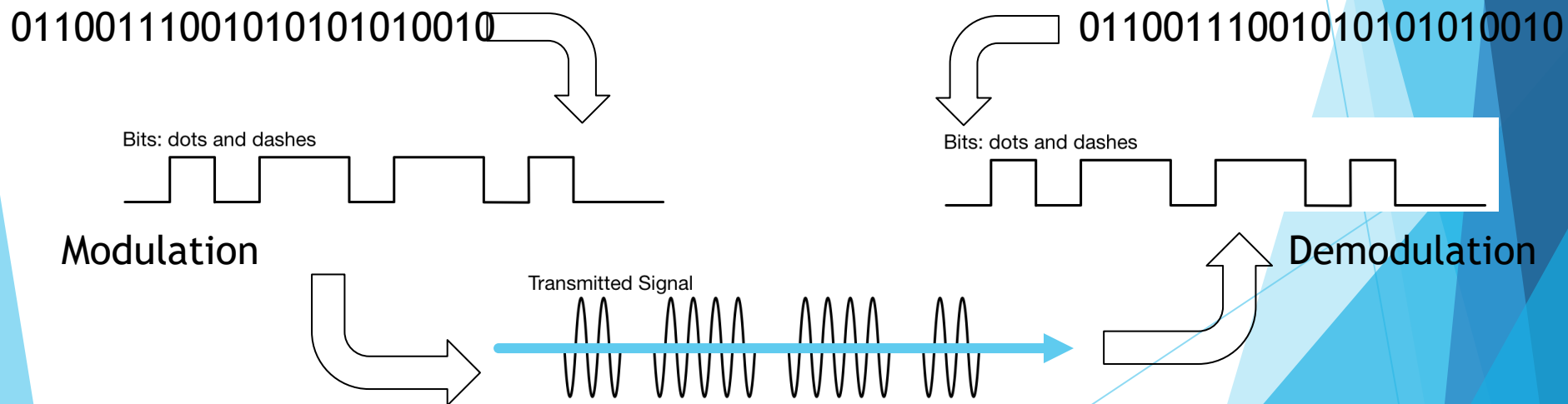




# First layer in OSI/ISO model

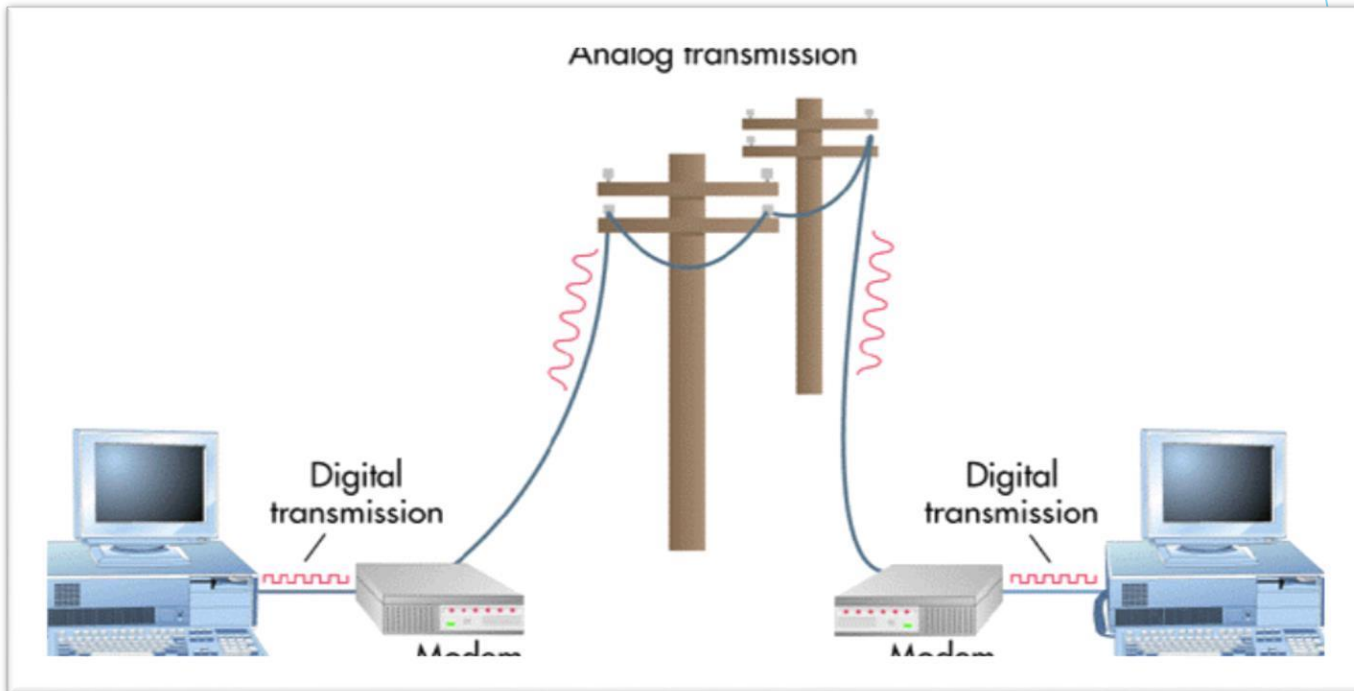
# Data Communications

- ▶ A specialized subset of telecommunications that refers to the electronic collection, processing, and distribution of data -- typically between computer system hardware devices.



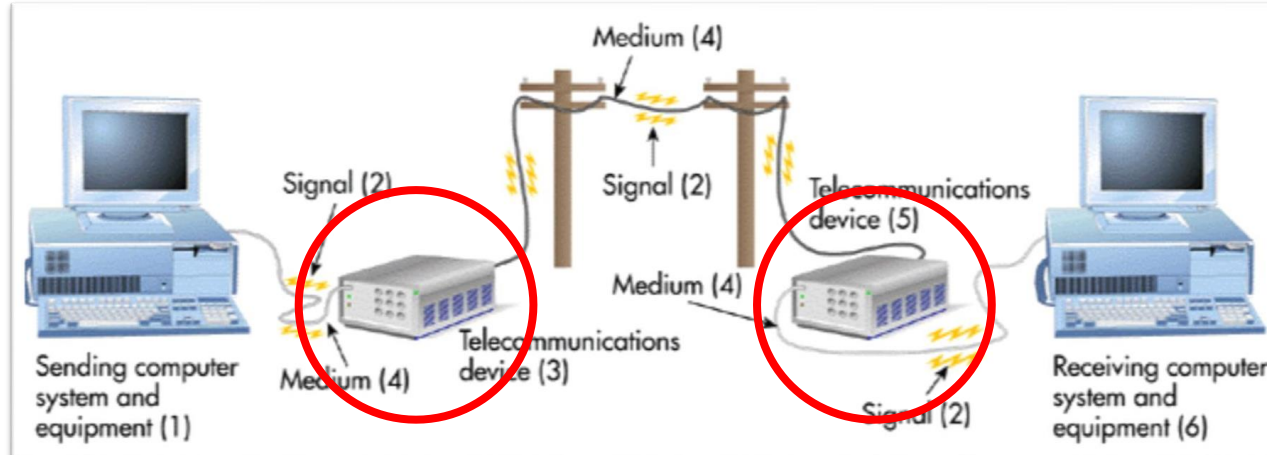
# Device: Modem

- ▶ Modem
- ▶ Modulates a digital signal into an analog signal for transmission via analog medium, then demodulates the signal into digital for receiving.



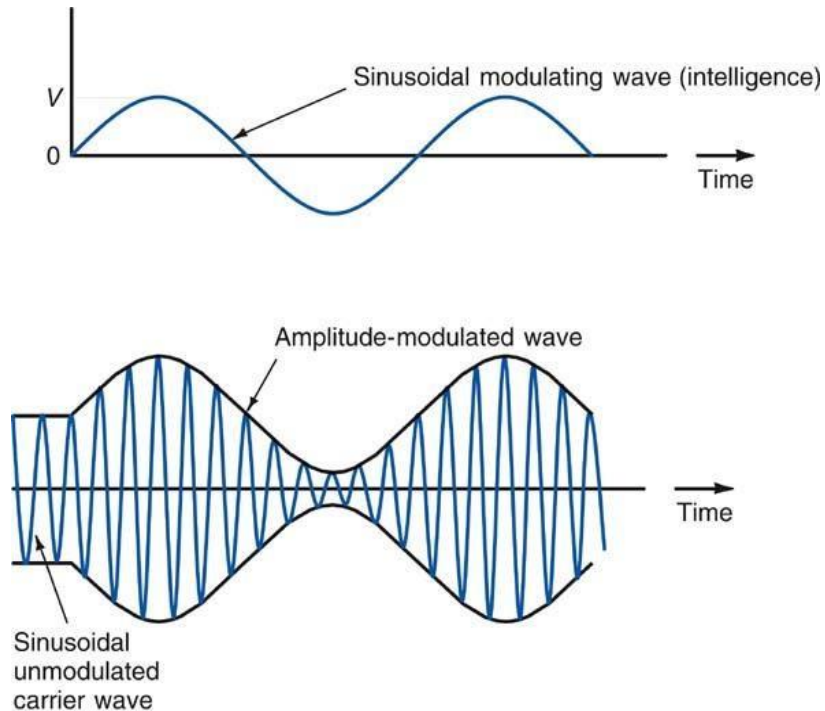
# Telecommunications System

- ▶ Telecommunication Devices
- ▶ Relay signals between computer systems and
- ▶ transmission media.



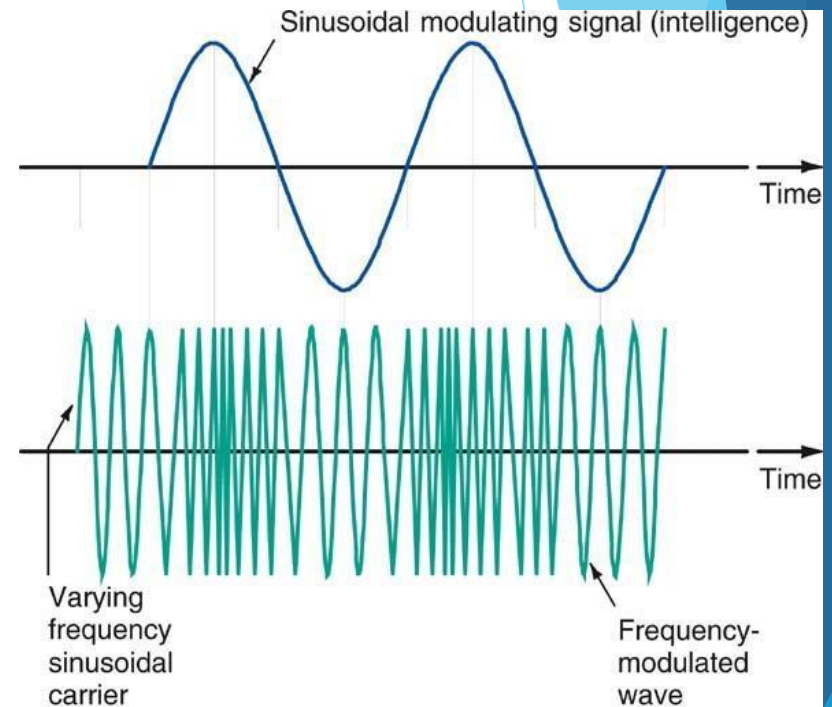
# Example of Type Modulation Techniques

## Amplitude modulation.



(a)

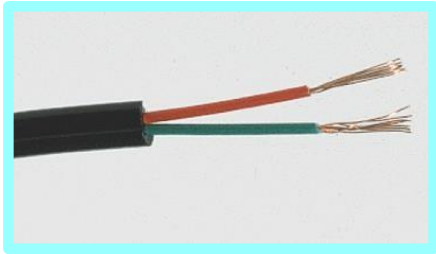
## Frequency modulation



(b)

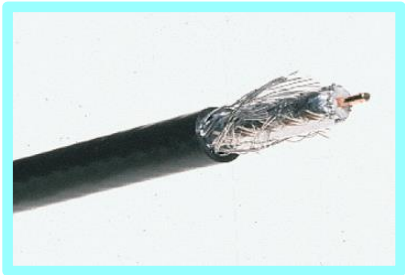


# Physical Layer and Types of Telecommunication Media



## ▶ Twisted Pair Wire Cable

- ▶ Insulated pairs of wires historically used in telephone service and to connect computer devices.



## ▶ Coaxial Cable

- ▶ Consists of an inner conductor wire surrounded by insulation, called the dielectric. The dielectric is surrounded by a conductive shield, which is surrounded by a non-conductive jacket. Coaxial cable has better data transmission rate than twisted pair.

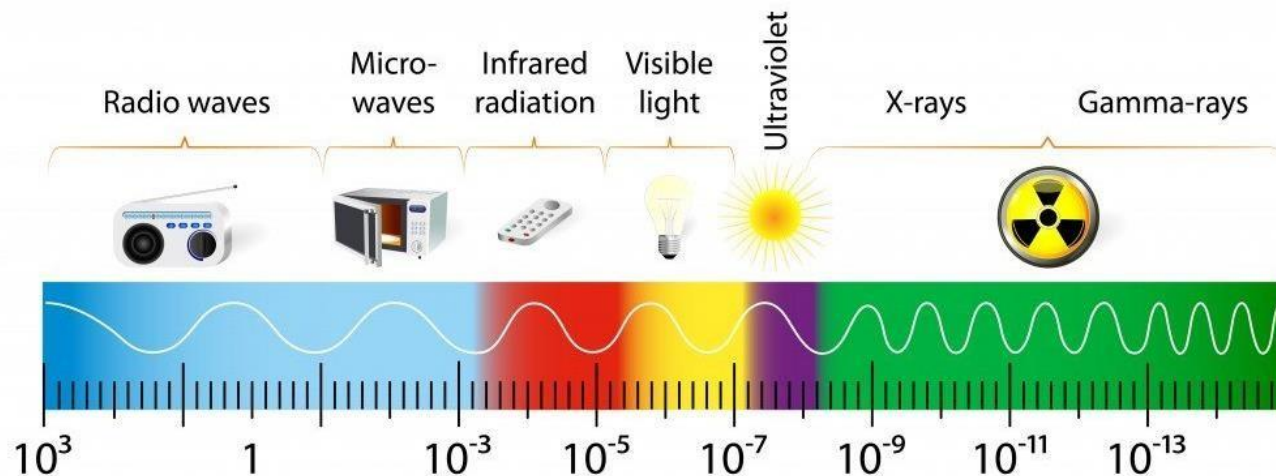
# Types of Telecommunication Media

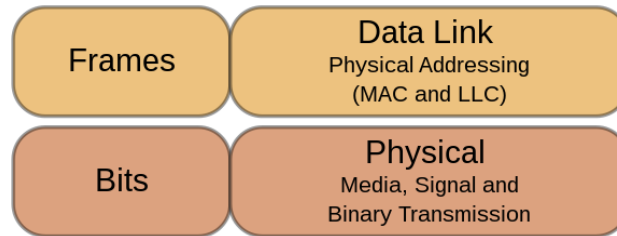
- ▶ **Fiber-optic Cable**
- ▶ Many extremely thin strands of glass or plastic bound together in a sheathing which transmits signals with light beams. Can be used for voice, data, and video.



# Medium Electromagnetic Spectrum

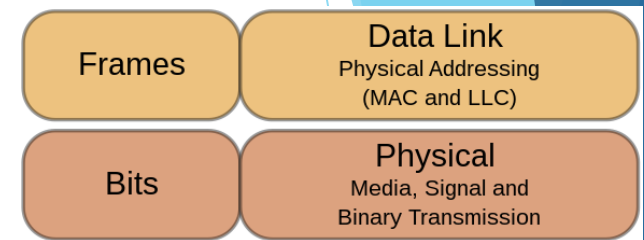
- ▶ Frequencies are more or less well-carried according to the medium type (that's key !)
- ▶ Give some examples of couples Frequencies / Network
- ▶ Technologies ?





# Second Layer in the OSI/ISO model

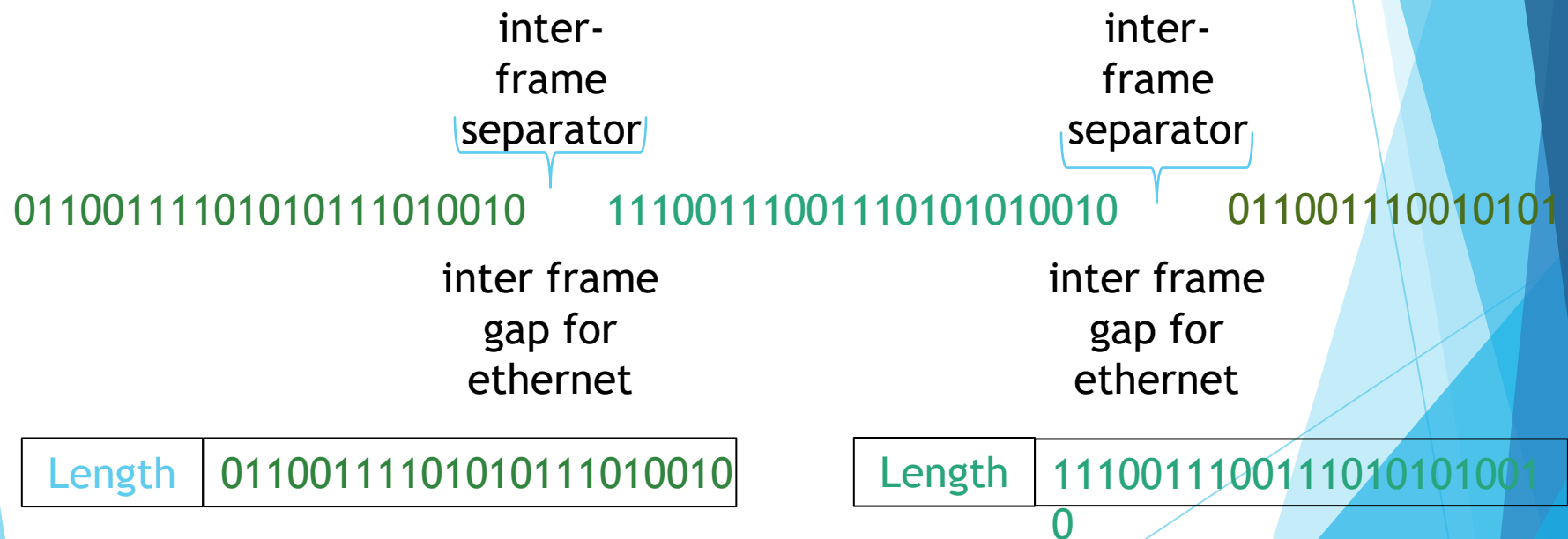
# Data Communication Management



- ▶ How to distinguish consecutive frames
- ▶ Shared Medium between various connected devices
- ▶ Who is sent the frame

# Distinguishing consecutive frames

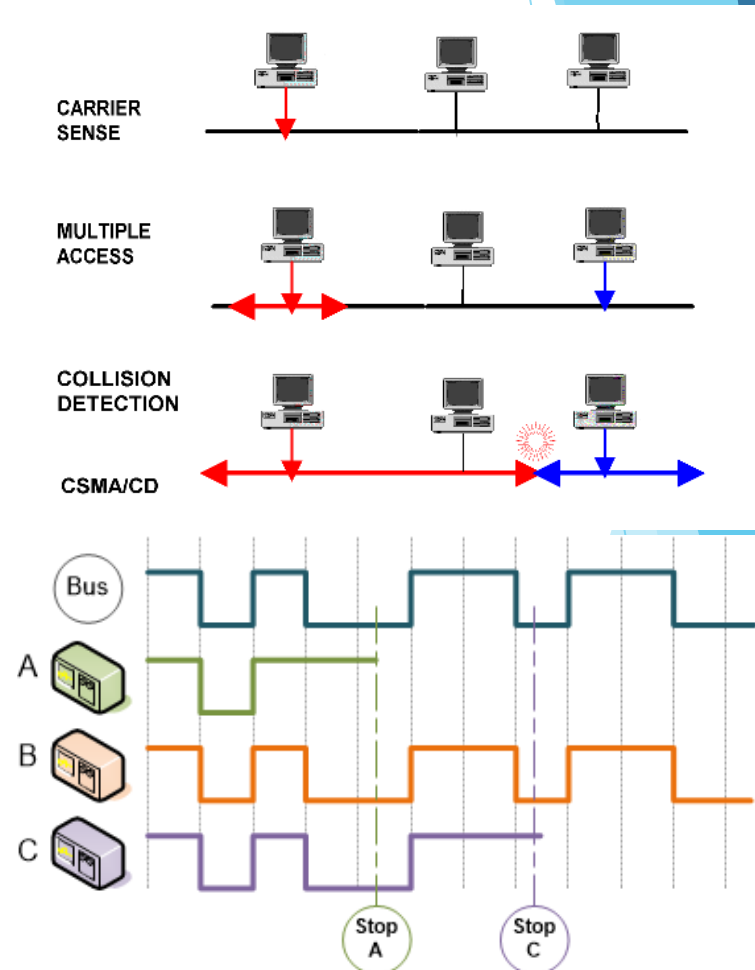
## ► Different kinds of strategies



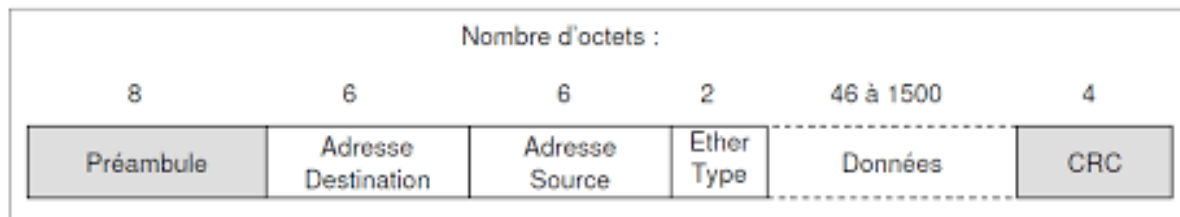
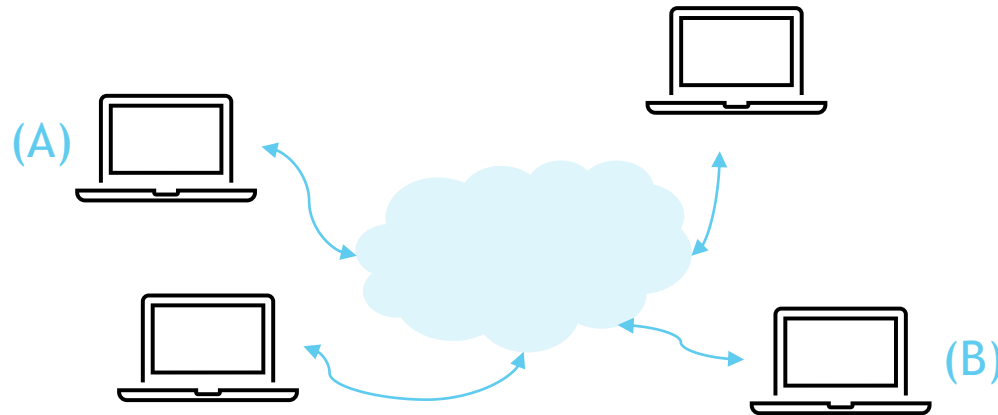


# Shared medium and multiple connected devices

- ▶ CSMA/CD (Ethernet)
- ▶ CSMA/CR (CAN)
- ▶ CSMA/CA (Wifi)
- ▶ Different kinds of strategies



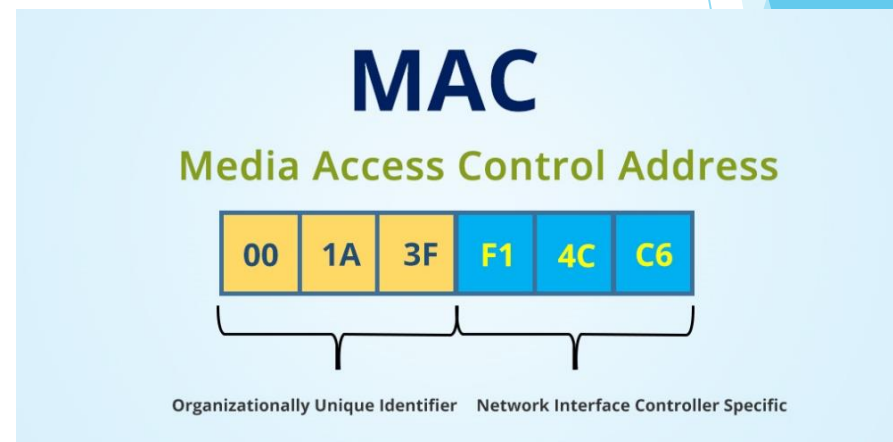
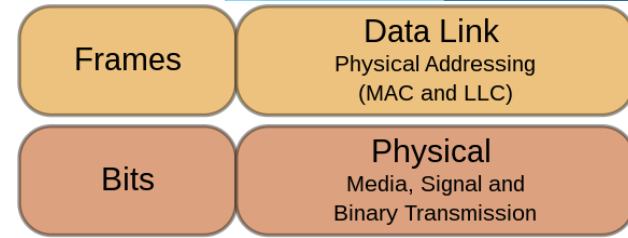
# Who is sent the frame ?



**Format de la trame Ethernet V2**

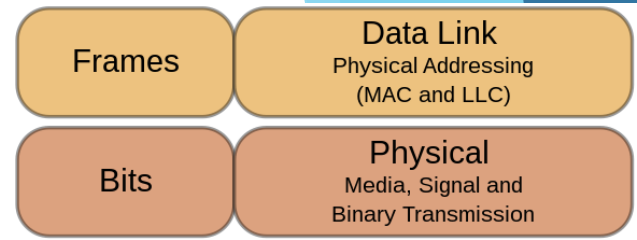
# MAC Address (IEEE 802.1)

- For
  - Local Area Network (LAN, ex. Ethernet)
  - Wireless Local Area Network (WLAN, ex. Wifi, Lifi)
  - Personal Area Network (PAN, ex. Bluetooth)
  - Low Power Wide Area Network (LPWAN, ex. LoRa, Sigfox, Zigbee)



- See *ifconfig* / *ipconfig* commands

# LAN devices



*Ethernet Switch*



*Bluetooth hotspot*

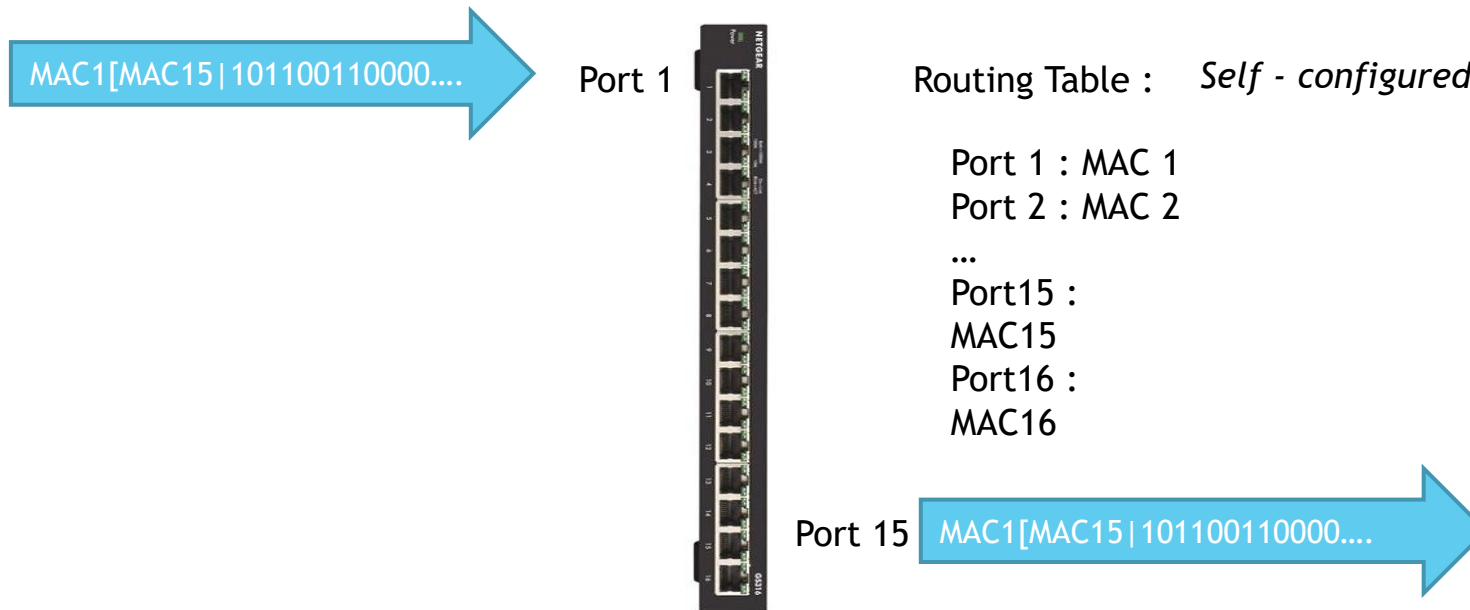


*Lifi hotspot*



*Wifi hotspot*

# Ethernet and Switch devices



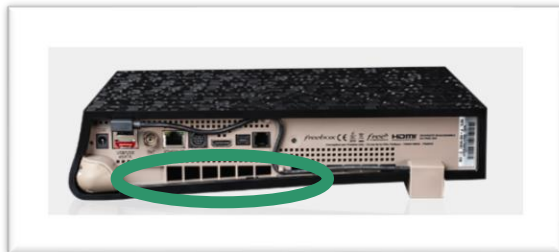
Be carefull, switches are have a max bandwith or speed in Bytes / s (ex. 100Mb/s, 1Gb/s, 10GB/s ...)

# Ethernet infrastructure

simple switch



Rackable switch



Integrated switch (ex. freebox)

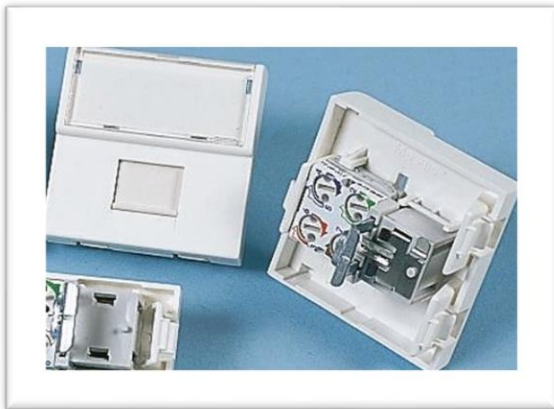
Rack



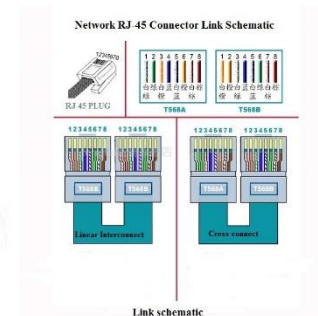
patch bay



# Ethernet Plugs



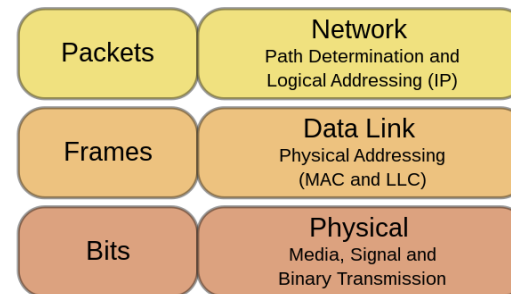
Plugs



Plug and cable Connexion

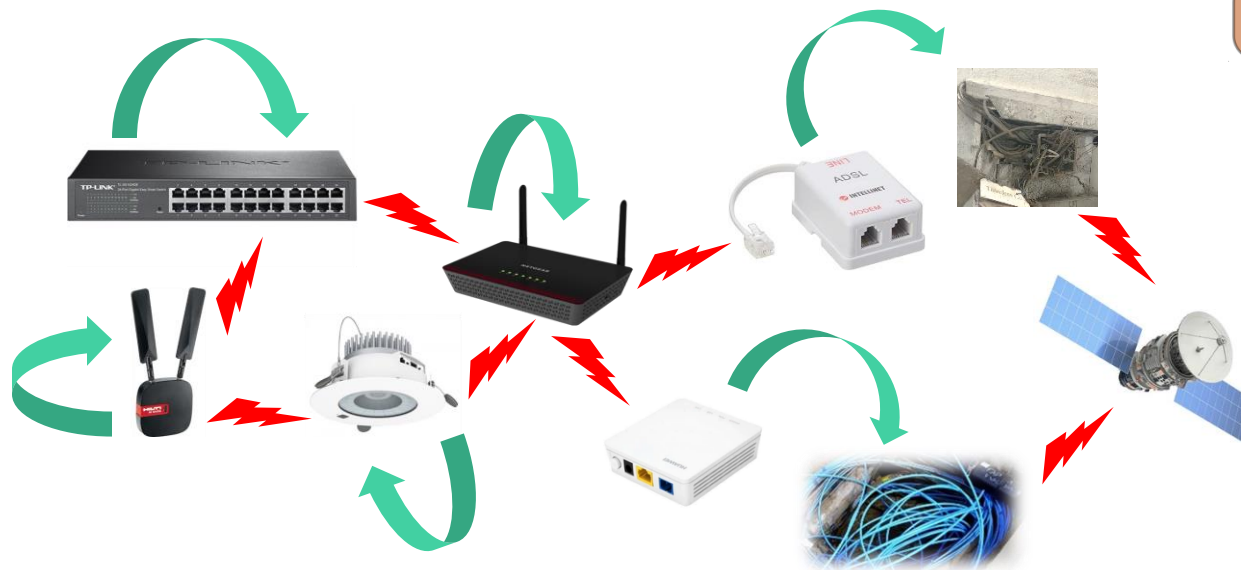
Plug must be compatible with cables !

# Third layer in OSI/ISO model



# Internet : the network of networks

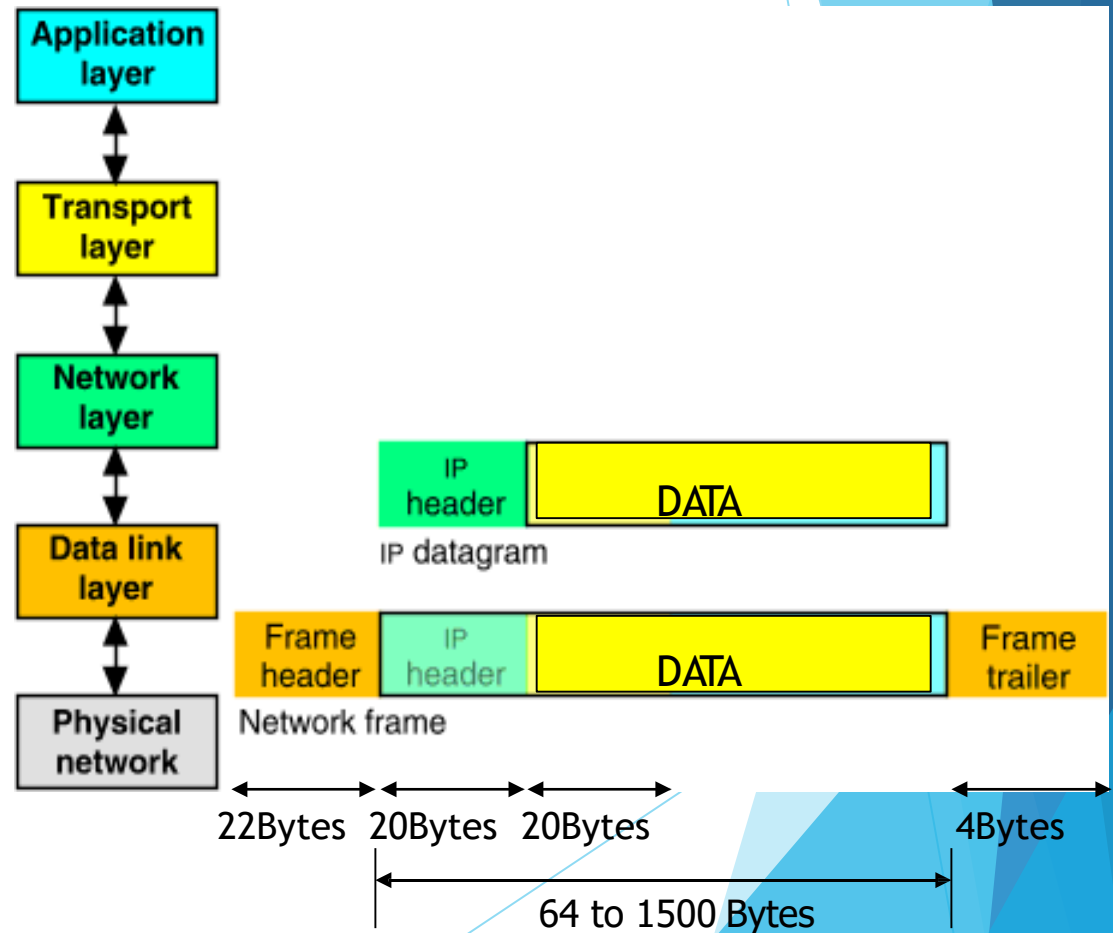
► From a network to another ...



Packets	Network Path Determination and Logical Addressing (IP)
Frames	Data Link Physical Addressing (MAC and LLC)
Bits	Physical Media, Signal and Binary Transmission

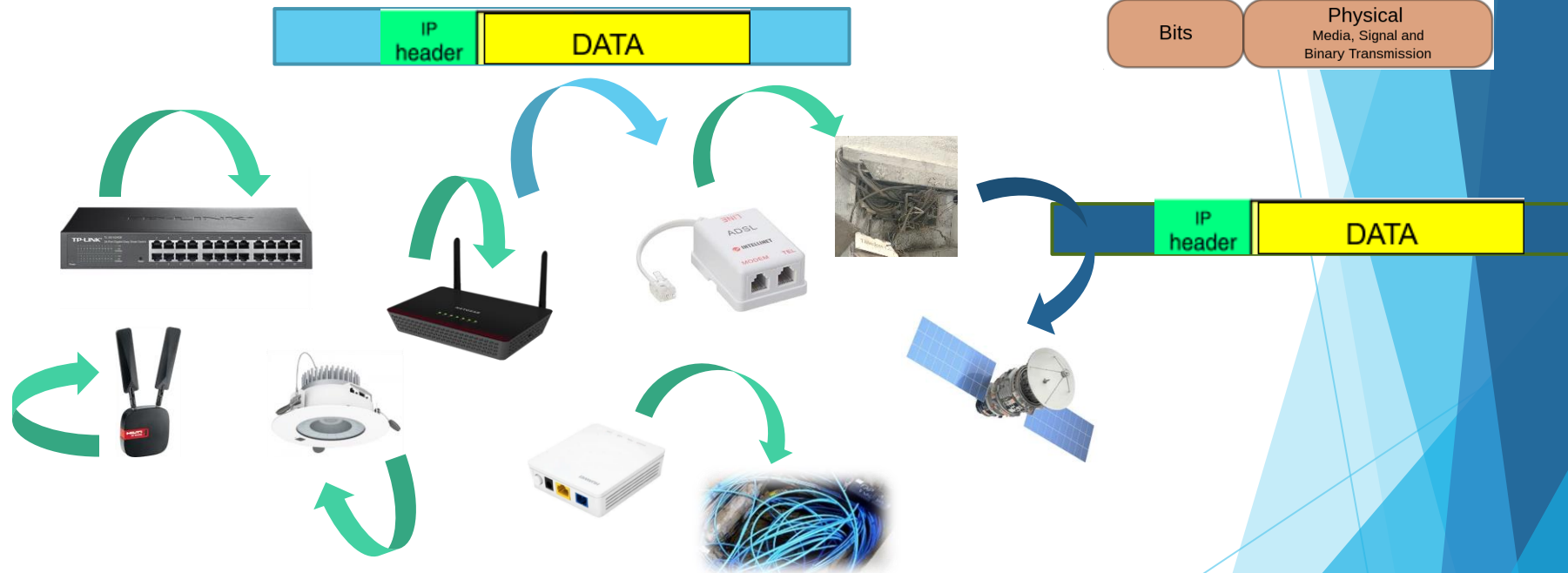
# Packet Encapsulation in OSI/ISO model

- ▶ The data is sent down the protocol stack
- ▶ Each layer adds to the data by prepending headers



# Internet : the network of networks

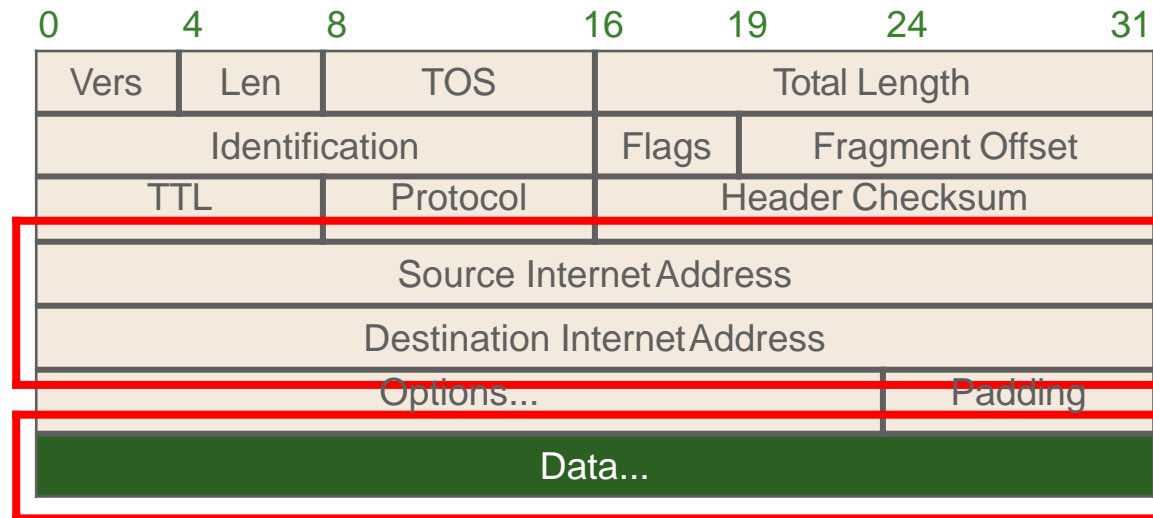
► From a network to another ...



# IP Datagram



Simplification



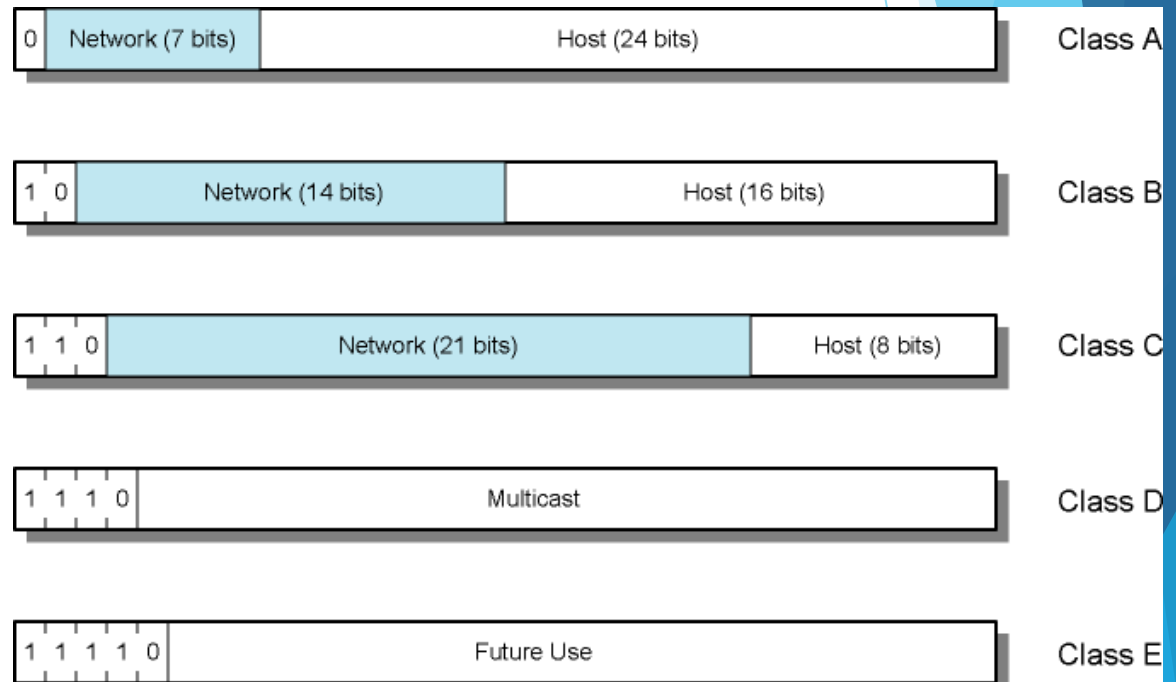
Field	Purpose
Vers	IP version number
Len	Length of IP header (4 octet units)
TOS	Type of Service
T. Length	Length of entire datagram (octets)
Ident.	IP datagram ID (for frag/reassembly)
Flags	Don't/More fragments
Frag Off	Fragment Offset

Field	Purpose
TTL	Time To Live - Max # of hops
Protocol	Higher level protocol (1=ICMP, 6=TCP, 17=UDP)
Checksum	Checksum for the IP header
Source IA	Originator's Internet Address
Dest. IA	Final Destination InternetAddress
Options	Source route, time stamp, etc.
Data...	Higher level protocol data



# IPv4 Address Formats

- ▶ 32 bit global Internet address
- ▶ Network part and host part



# Network Masks

- ▶ Distinguishes which portion of the address identifies the network and which portion of the address identifies the node.
- ▶ Default masks:
- ▶ Class A: 255.0.0.0
- ▶ Class B: 255.255.0.0
- ▶ Class C: 255.255.255.0

# Netid, Hostid, Mask

Mask

255.255.0.0

Netid

Hostid

11111111	11111111	00000000	00000000
----------	----------	----------	----------

a. Without subnetting

Mask

255.255.240.0

Netid

Subnetid

Hostid

11111111	11111111	1111	0000	00000000
----------	----------	------	------	----------

b. With subnetting

# Some Special IP address forms

- ▶ All-zero host part identifies the network
- ▶ All-one host part means broadcast (limited to current network)

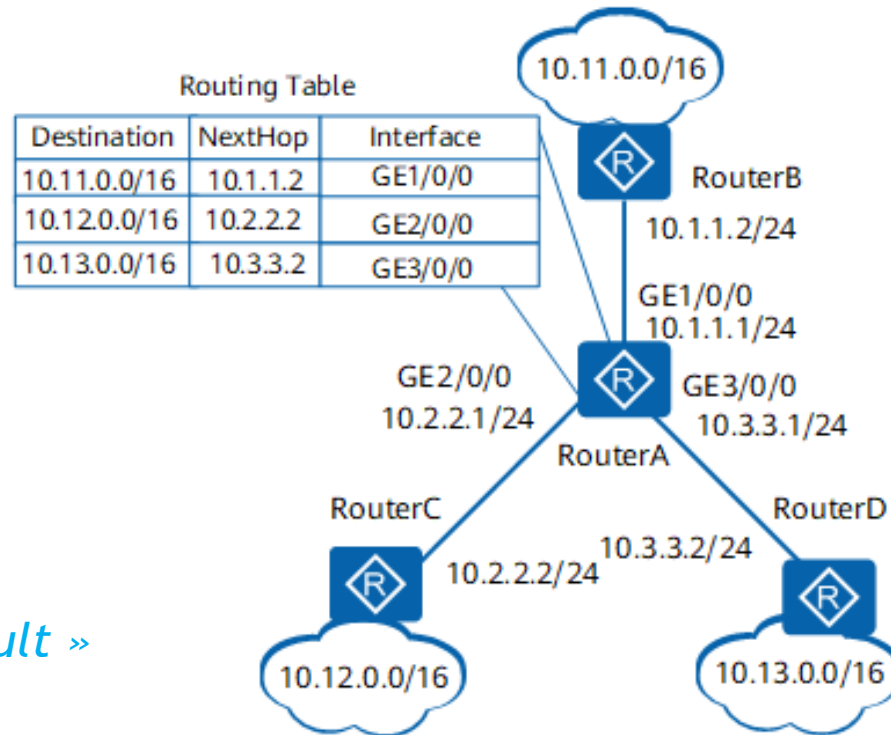
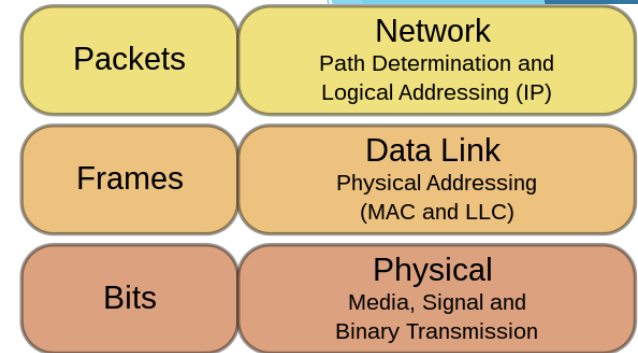
Prefix (network)	Suffix (host)	Type & Meaning
all zeros	all zeros	this computer (used during bootstrap)
network address	all zeros	identifies network
network address	all ones	broadcast on the specified network
all ones	all ones	broadcast on local network
127	any	loopback (for testing purposes)

# Private Address Range

Address Class	Reserved Address Space
Class A	10.0.0.0 - 10.255.255.255
Class B	172.16.0.0 - 172.31.255.255
Class C	192.168.0.0 - 192.168.255.255

# Router Device

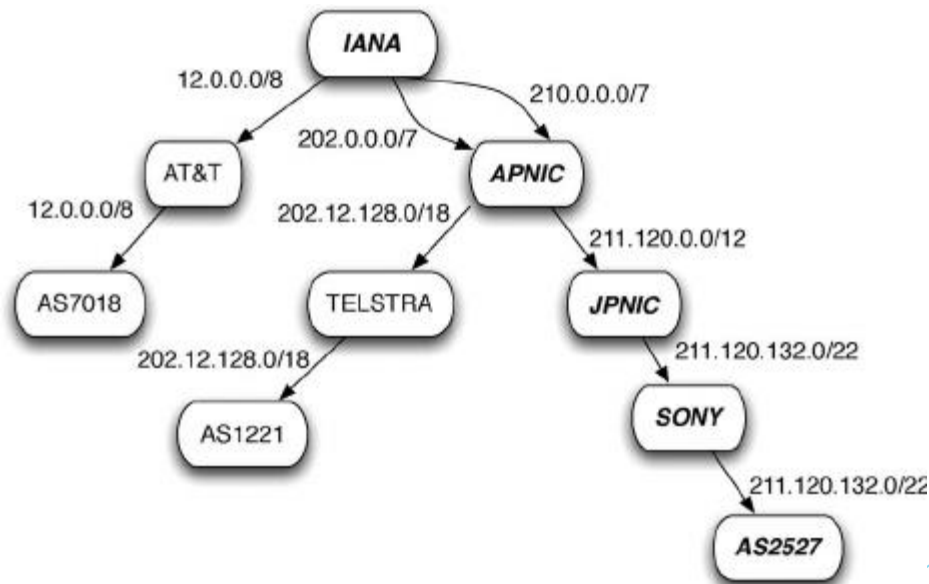
## ► And routing table ...



*Last entry is « default »*

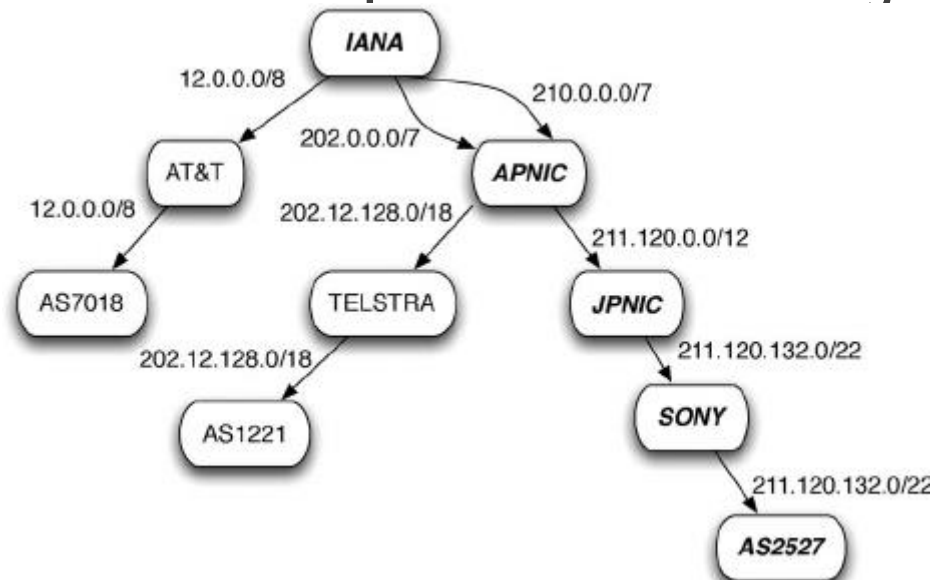
# Hierarchical organization of the IP addresses

- To avoid combinatorial explosion in the routing table ...



# Hierarchical organization of the IP addresses

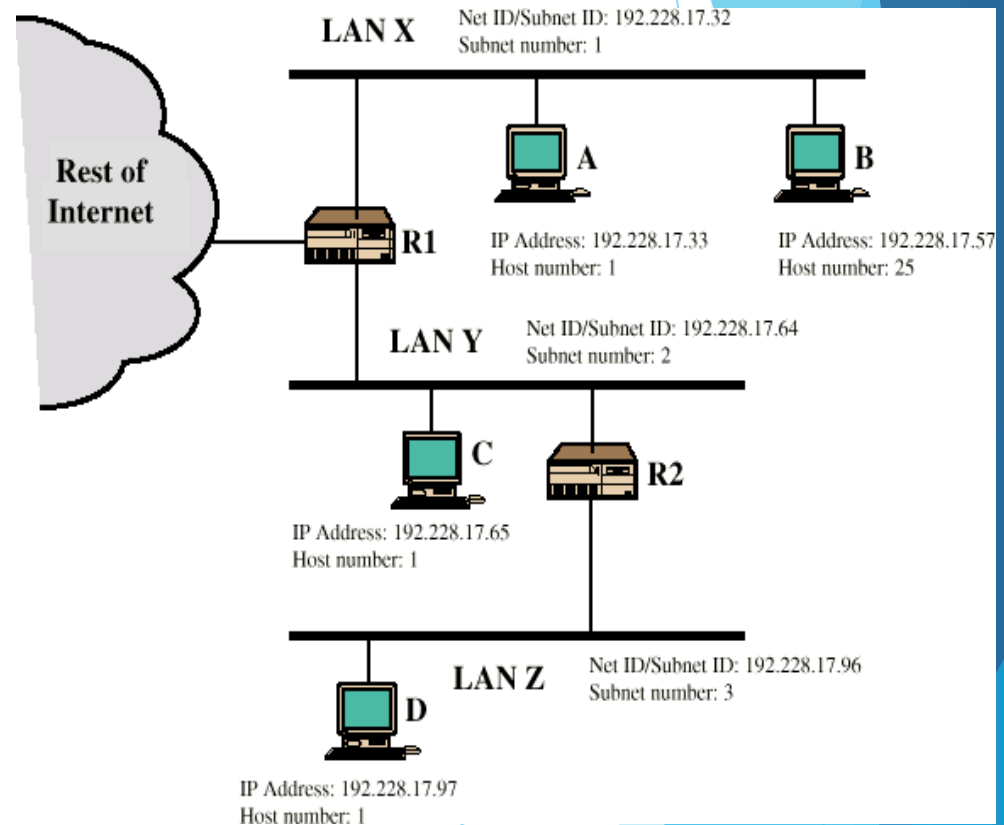
- To avoid combinatorial explosion in the routing table ...



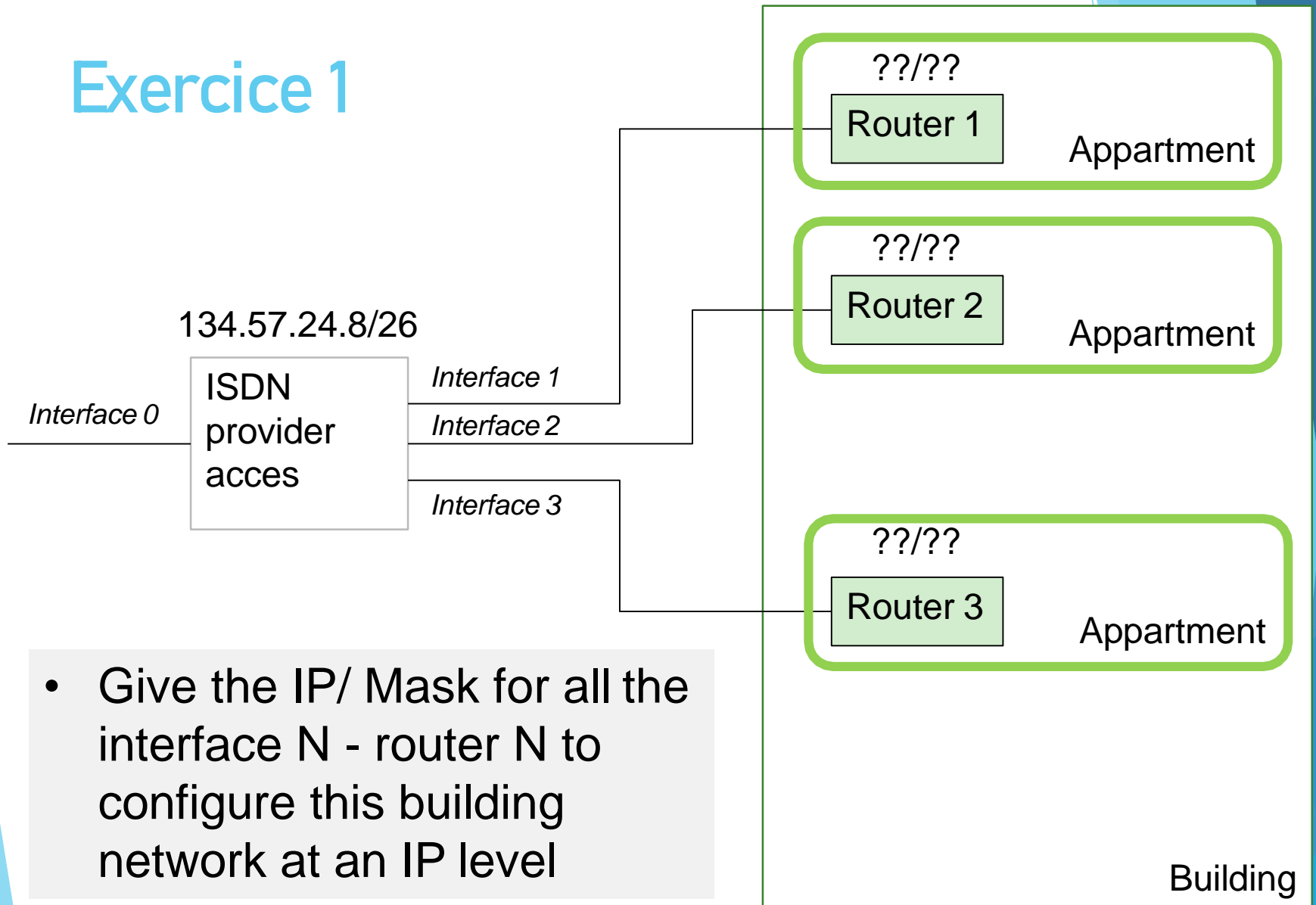


# Routing Using Subnets (Example)

- ▶ Subnet Mask:
- ▶ 255.255.255.224
- ▶ Addresses start with 192, so class C addresses. Last octet is for Subnet number and Host number
- ▶ 224 -> 11100000 in binary, last 5 bits are for Host number, previous 3 bits are for Subnet number
- ▶ Don't forget! All zero host number identifies the subnet and all ones is used for broadcast

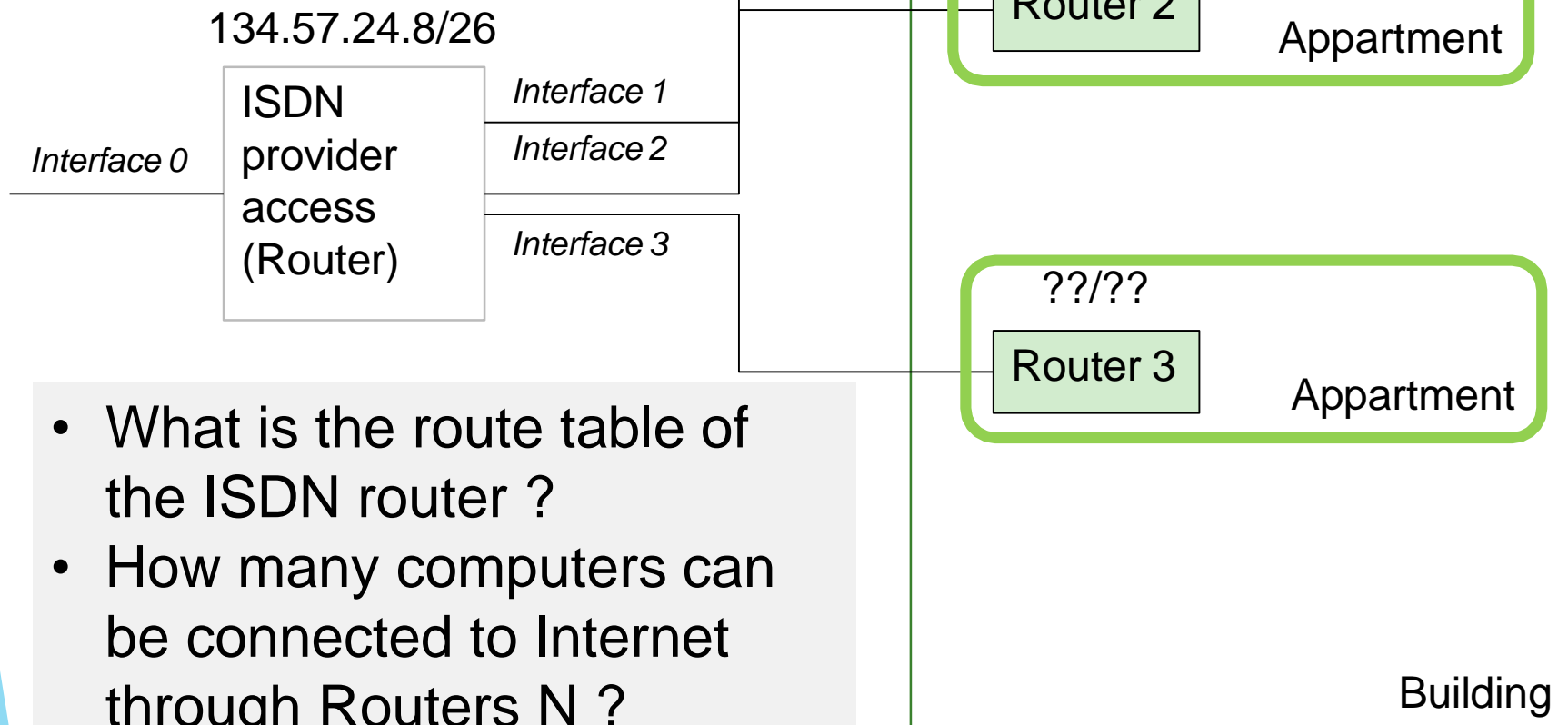


# Exercice 1



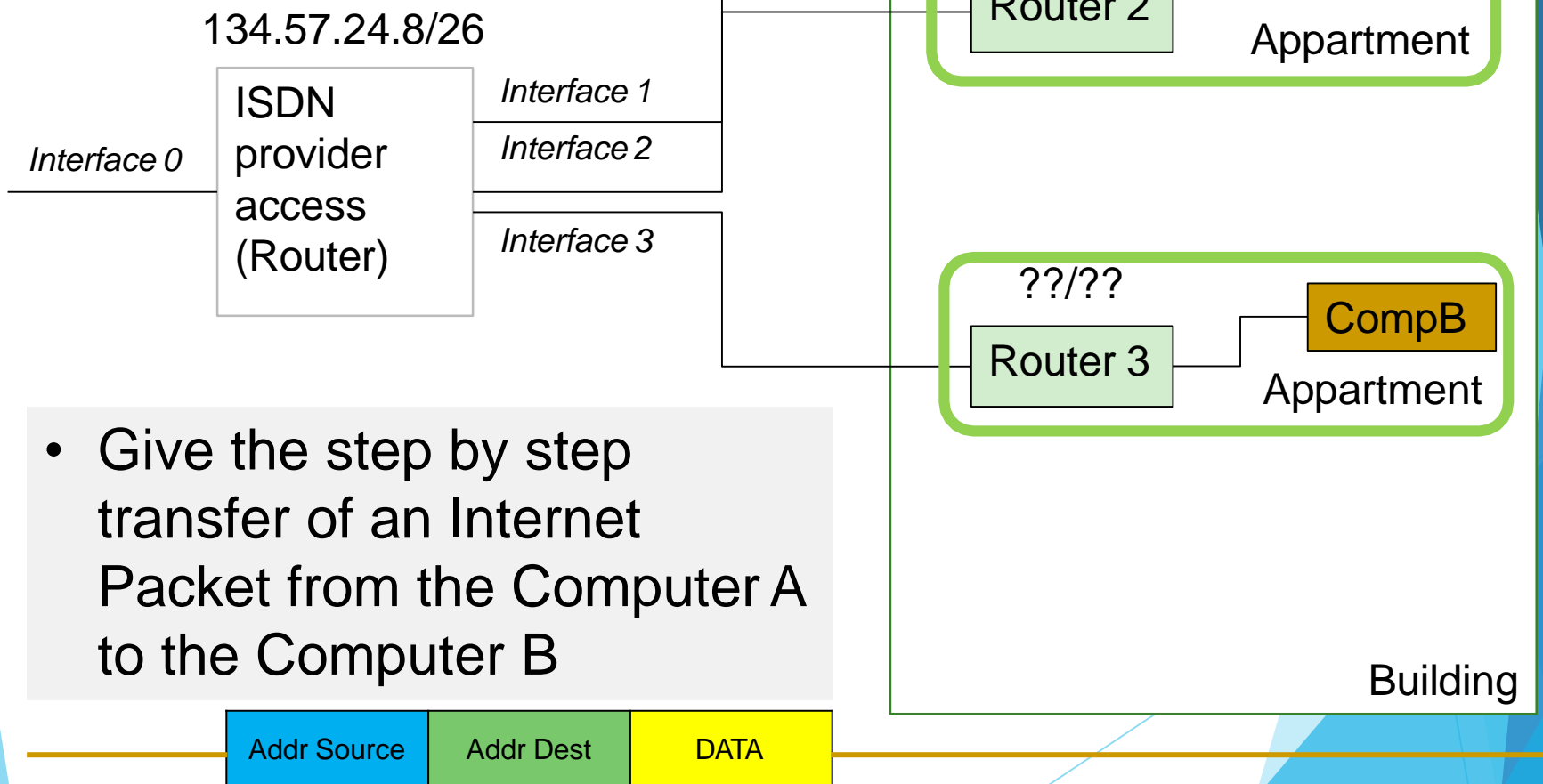
- Give the IP/ Mask for all the interface N - router N to configure this building network at an IP level

## Exercice 2



- What is the route table of the ISDN router ?
- How many computers can be connected to Internet through Routers N ?

# Exercice 3

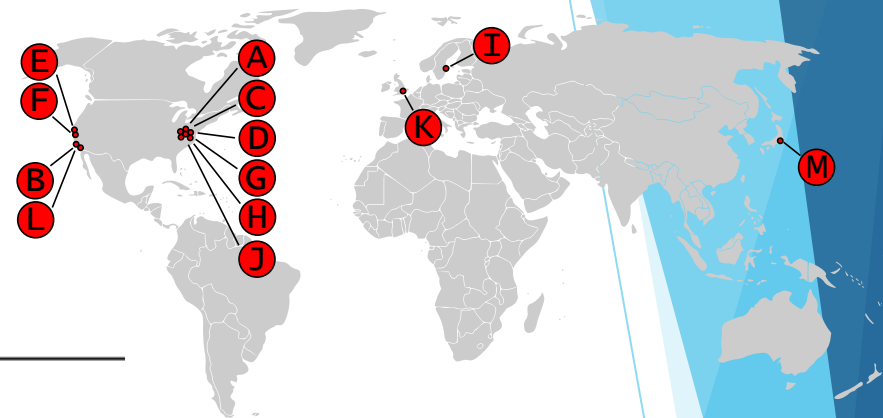


# IPv6 Enhancements

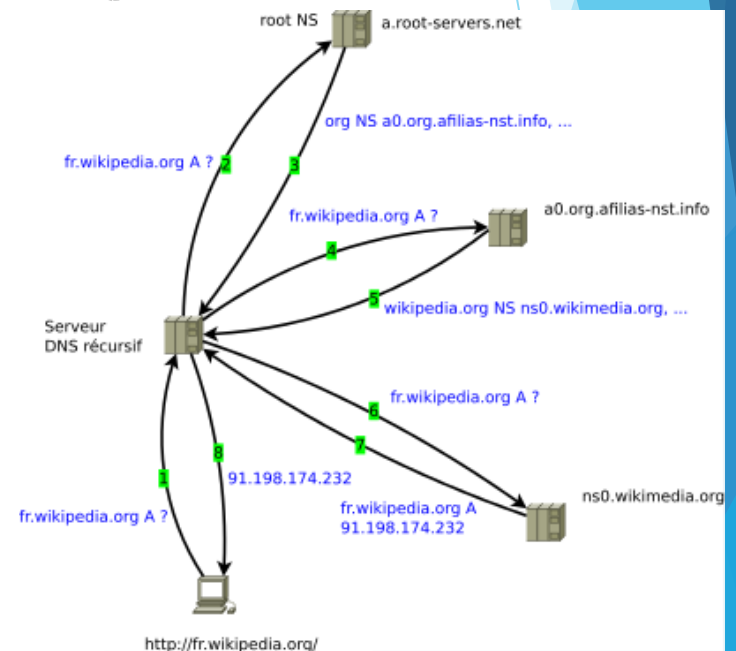
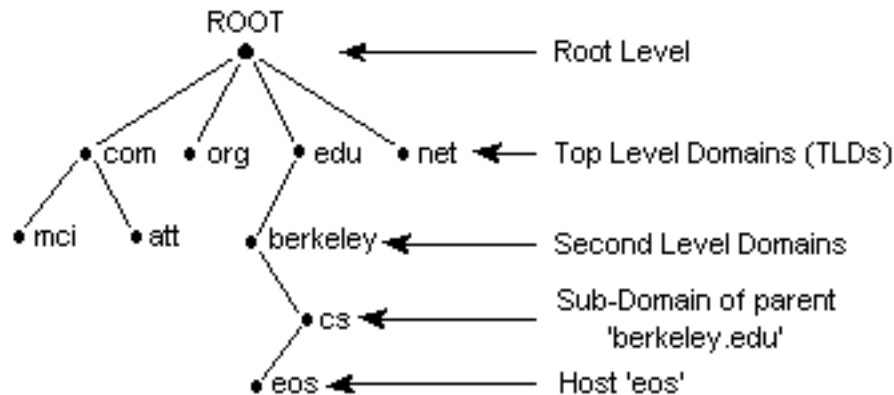
- ▶ Expanded address space
  - ▶ 128 bit
  - ▶  $6^{*}10^{23}$  addresses per square meter on earth!
- ▶ Improved option mechanism
  - ▶ Separate optional headers between IPv6 header and transport layer PDU
  - ▶ Some are not examined by intermediate routers
  - ▶ Improved speed and simplified router processing
  - ▶ Easier to extend with new options
  - ▶ Flexible protocol

# Help to associate names and IP addresses

## ► Domain Name Server : DNS

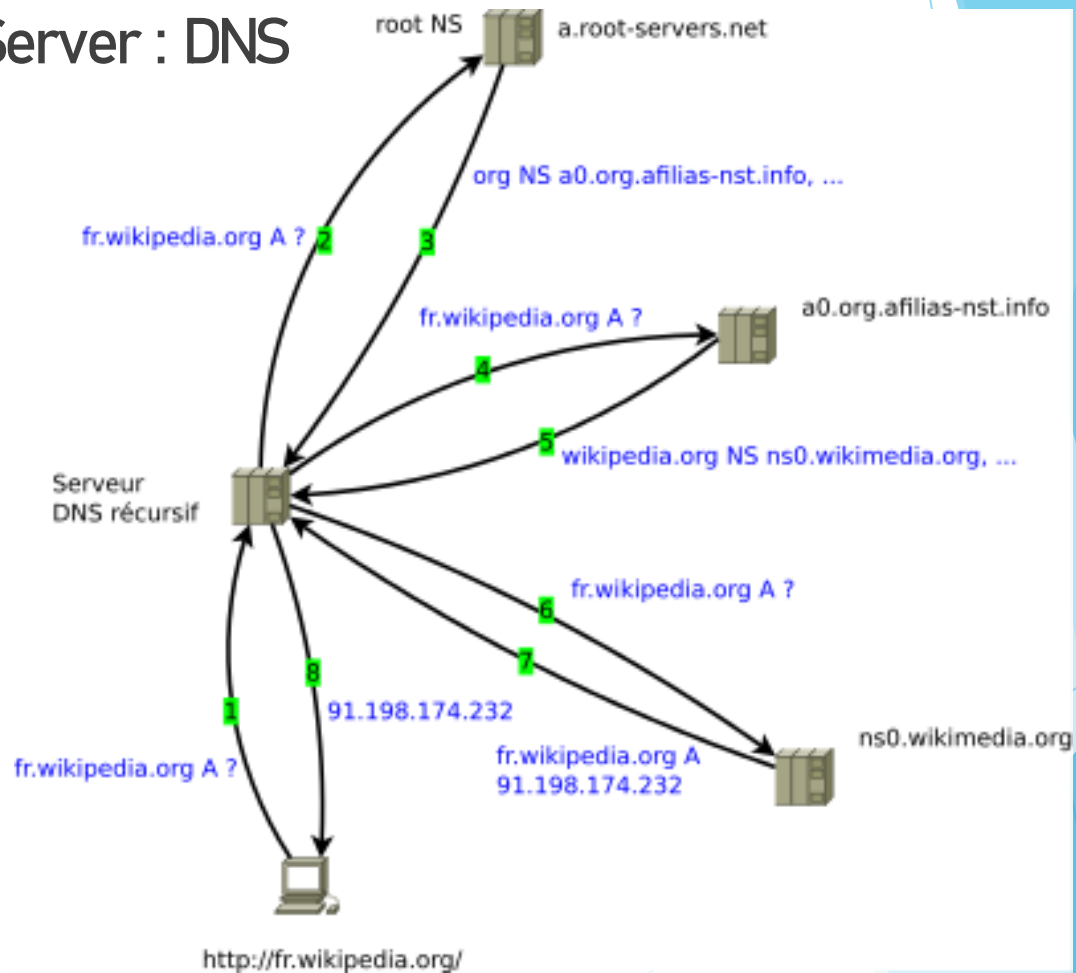


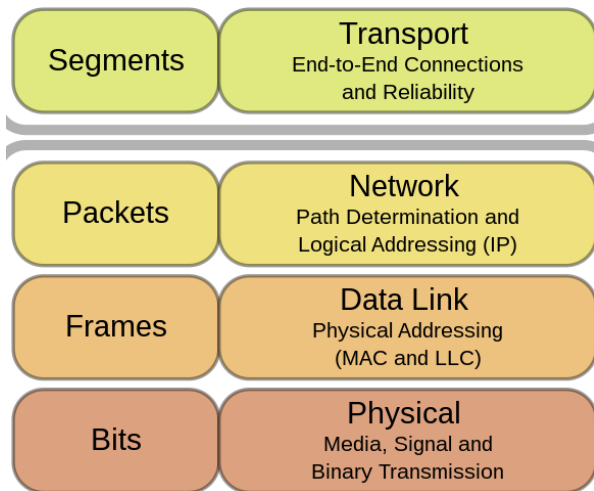
DNS Hierarchy



# How does it work ?

## ► Domain Name Server : DNS



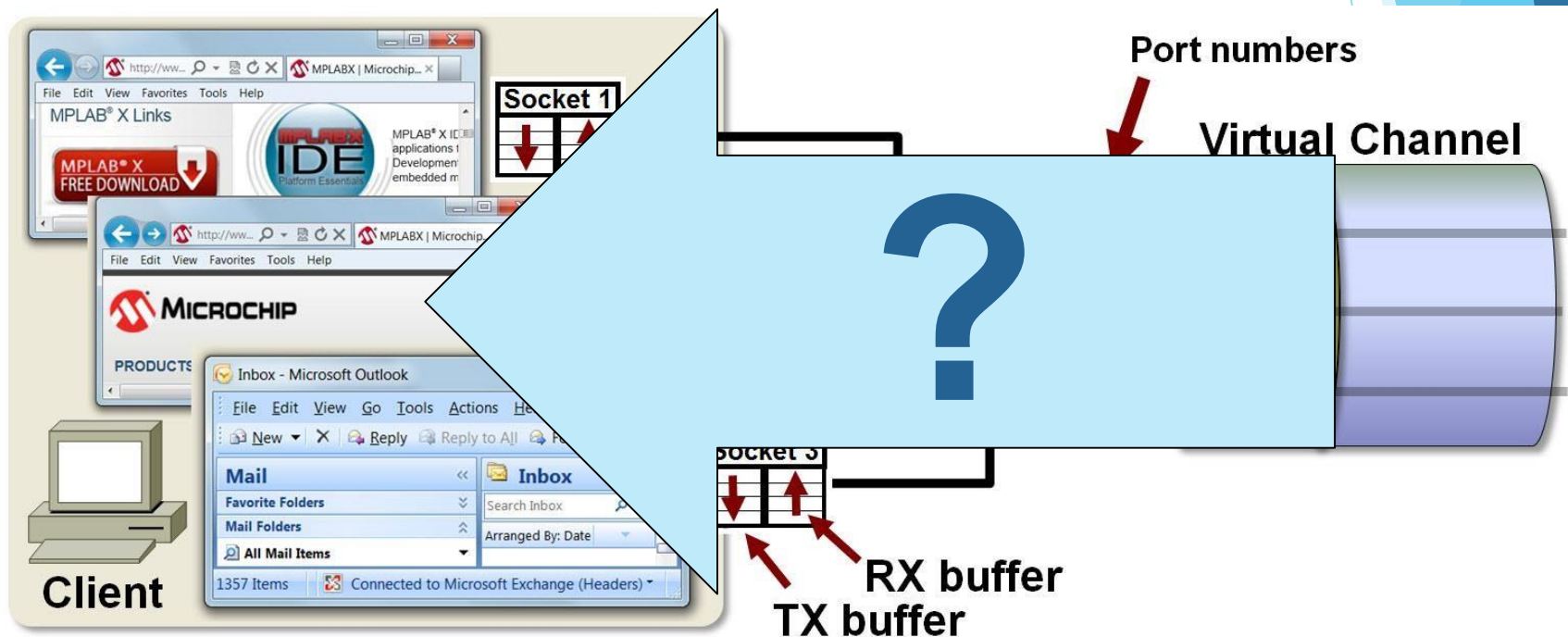


# Introduction to Transport Protocols over IP : UDP / TCP

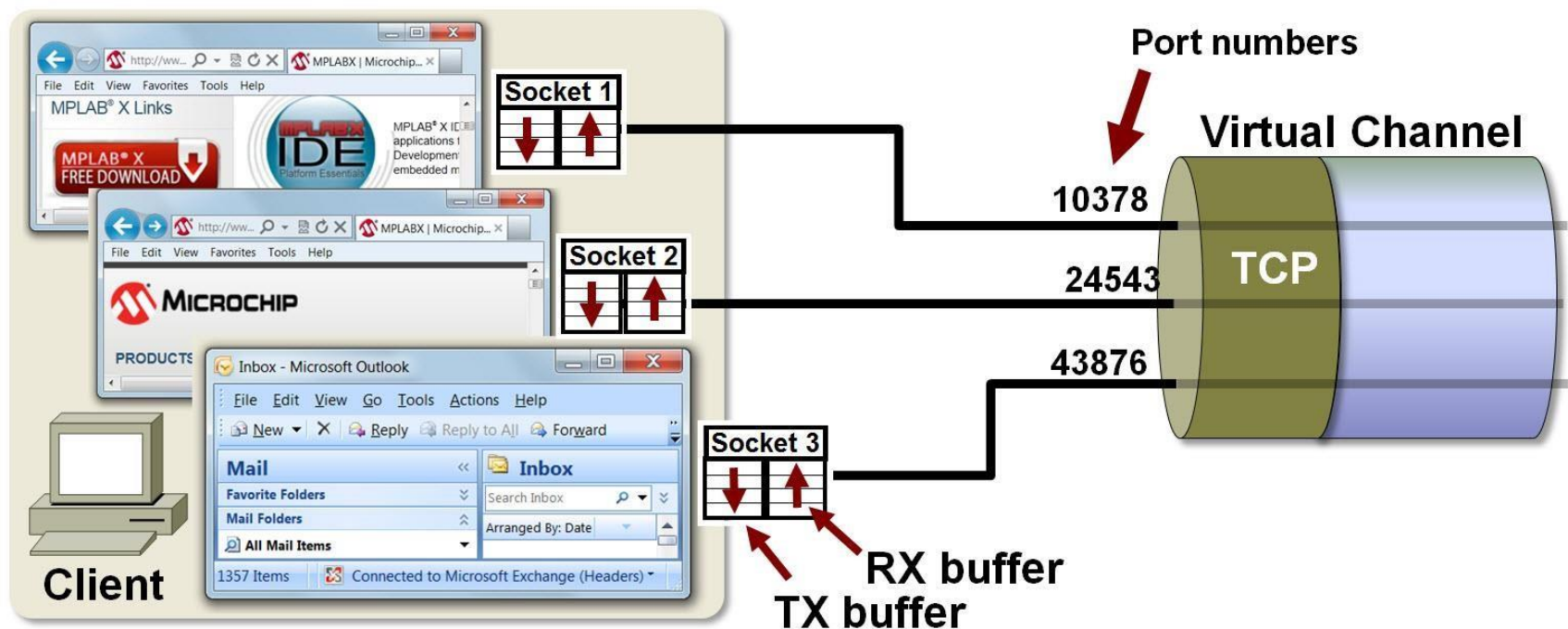


# How multiple programs can communicate over internet ?

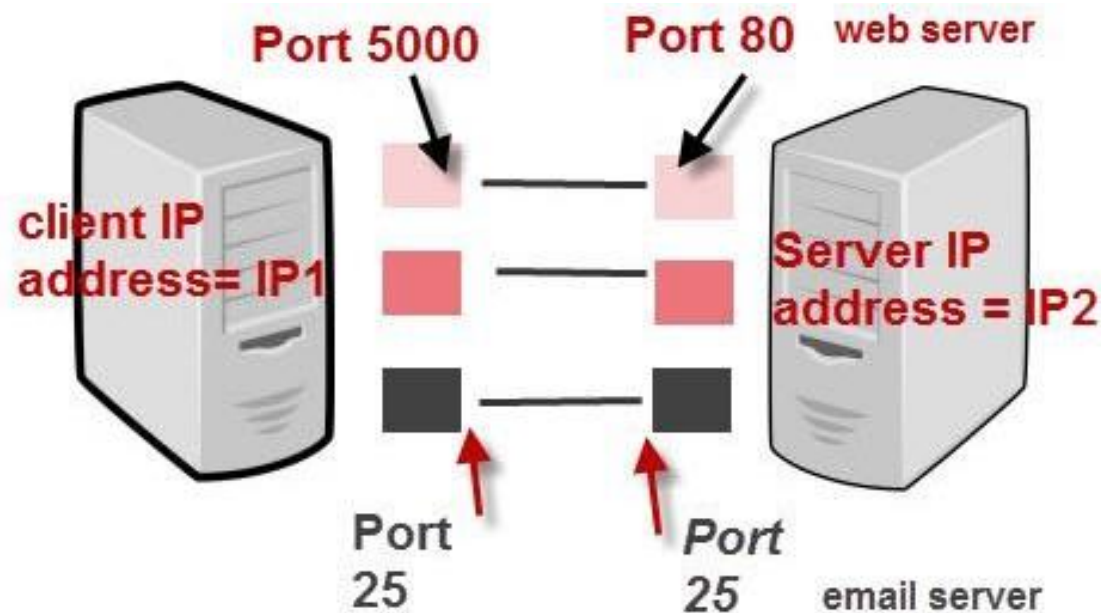
- How to reach an application with an IP packet ?



# Adding buffers in the computers and corresponding port number to IP Address



Communication endpoints are  
(IP\_Src/Port Src, IP\_Dest/Port Dest )



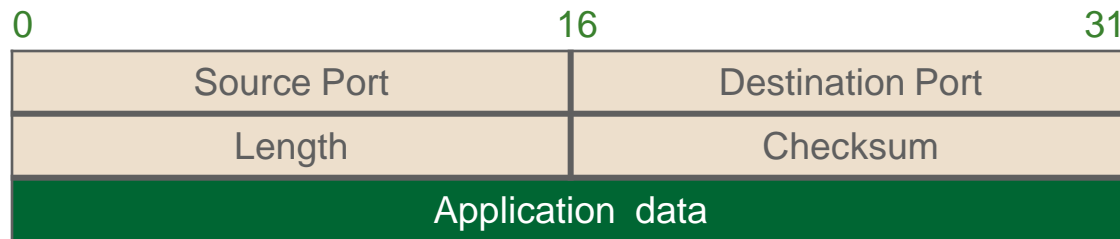
IP Address + Port number = Socket

## TCP/IP Ports And Sockets

# UDP

- ▶ Source and destination ports
  - ▶ Ports are used to associate a packet with a specific application at each end Thin layer on top of IP
- ▶ Adds packet length + checksum
  - ▶ Guard against corrupted packets
- ▶ Still unreliable:
  - ▶ Duplication, loss, out-of-orderness possible

# UDP datagram



Field	Purpose
Source Port	16-bit port number identifying originating application
Destination Port	16-bit port number identifying destination application
Length	Length of UDP datagram (UDP header + data)
Checksum	Checksum of IP pseudo header, UDP header, and data

# Typical applications of UDP

- ▶ Where packet loss etc is better handled by the application than the network stack
- ▶ Where the overhead of setting up a connection isn't wanted
- ▶ VOIP
- ▶ NFS – Network File System
- ▶ Most games

# TCP

- ▶ Reliable, full-duplex, connection-oriented, stream delivery
  - ▶ Interface presented to the application doesn't require data in individual packets
  - ▶ Data is guaranteed to arrive, and in the correct order without duplications
  - ▶ Or the connection will be dropped
  - ▶ Imposes significant overheads

# Applications of TCP

- ▶ Most applications !
- ▶ HTTP, FTP, ...
- ▶ Saves the application a lot of work, so used unless there's a good reason not to



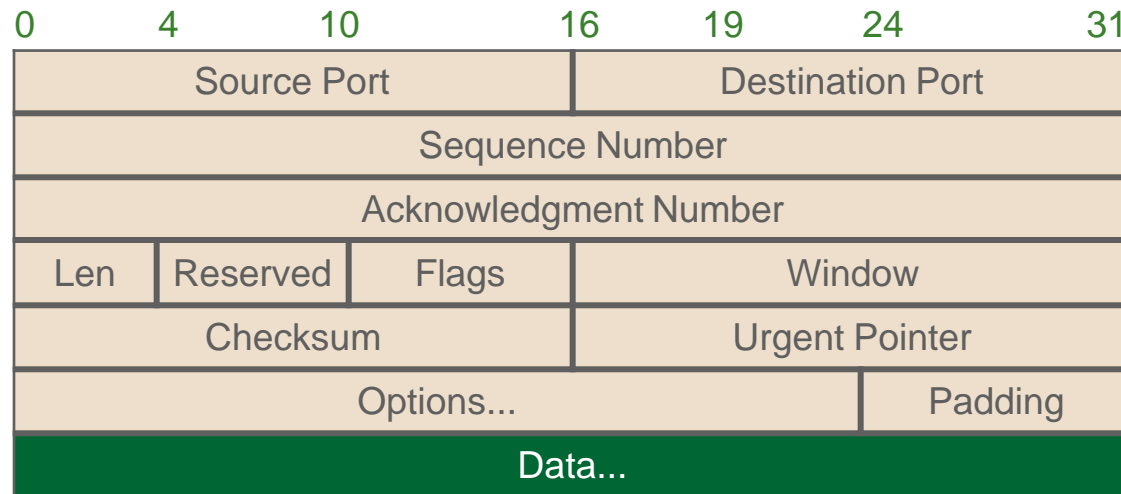
# TCP implementation

- ▶ Connections are established using a three- way handshake
- ▶ Data is divided up into packets by the operating system
- ▶ Packets are numbered, and received packets are acknowledged
- ▶ Connections are explicitly closed
- ▶ (or may abnormally terminate)

# TCP Packets

- ▶ Source + destination ports
- ▶ Sequence number (used to order packets)
- ▶ Acknowledgement number (used to verify packets are received)

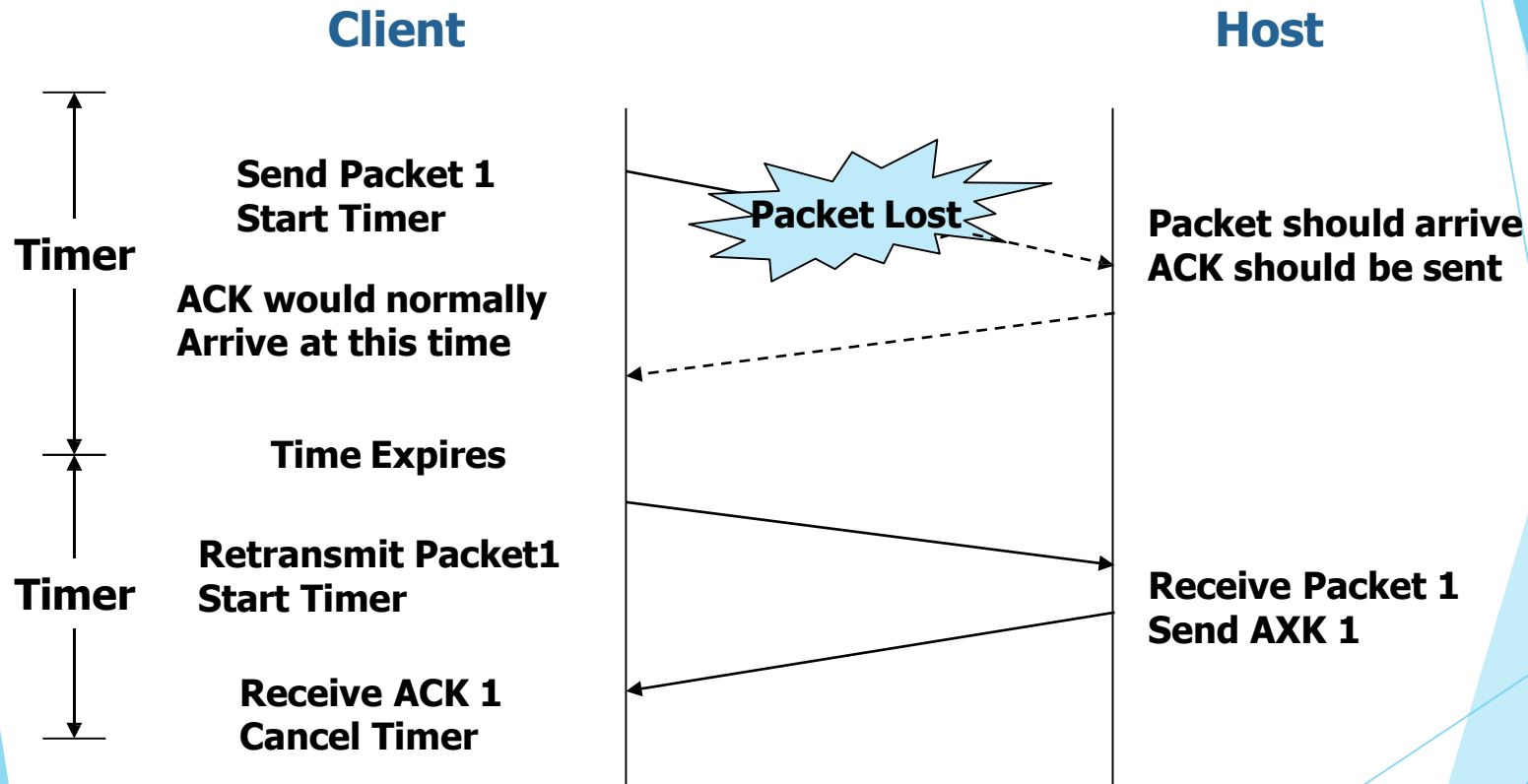
# TCP Segment



Field	Purpose
Source Port	Identifies originating application
Destination Port	Identifies destination application
Sequence Number	Sequence number of first octet in the segment
Acknowledgment #	Sequence number of the next expected octet (if ACK flag set)
Len	Length of TCP header in 4 octet units
Flags	TCP flags: SYN, FIN, RST, PSH, ACK, URG
Window	Number of octets from ACK that sender will accept
Checksum	Checksum of IP pseudo-header + TCP header + data
Urgent Pointer	Pointer to end of "urgent data"
Options	Special TCP options such as MSS and Window Scale

*You just need to know port numbers, seq and ack are added*

# TCP : Data transfer, robustness and lost packets



# Applications over TCP/IP



## Web Clients



## Mail Clients



## Instant Messaging



## Conferencing platforms

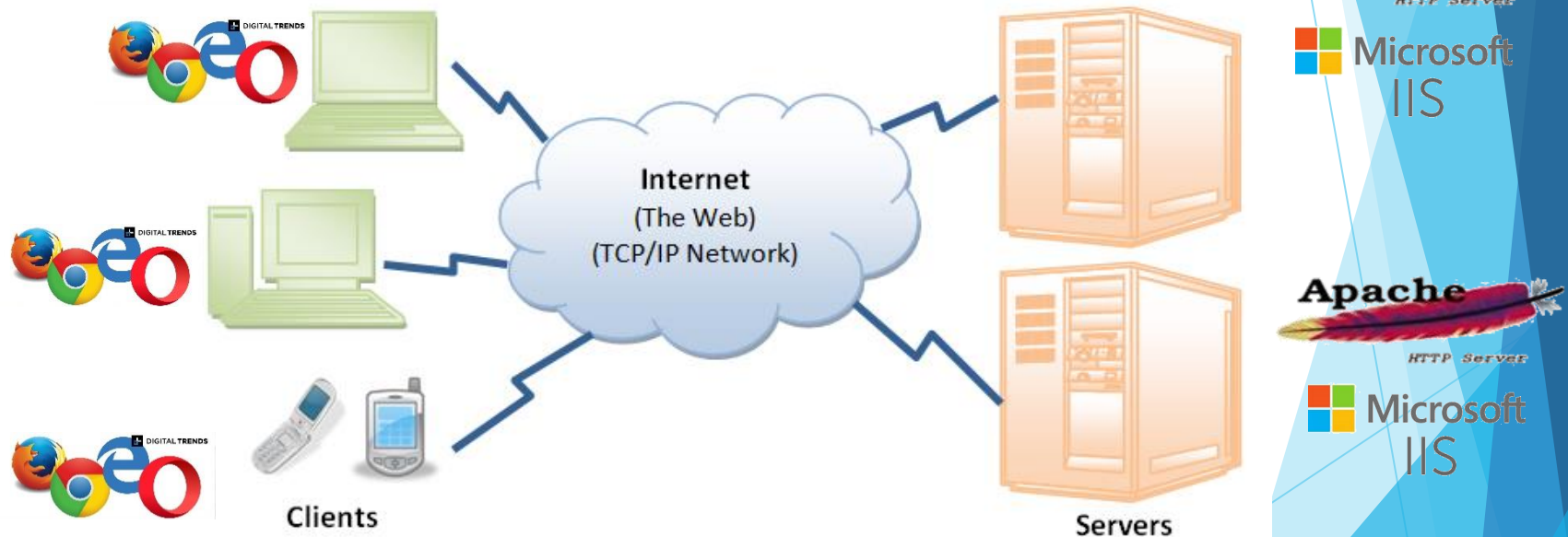


## Networked Games

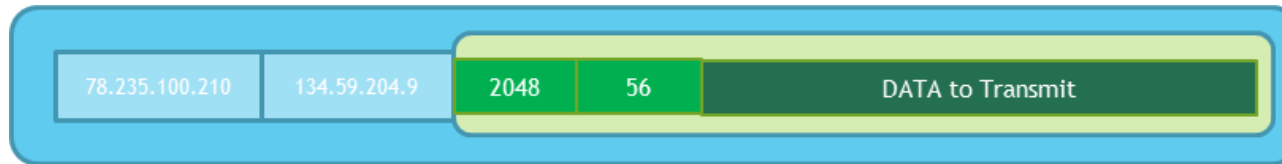
And more and more ....

# Web protocol, called HTTP

## ► HTTP : Hypertext Transfer Protocol



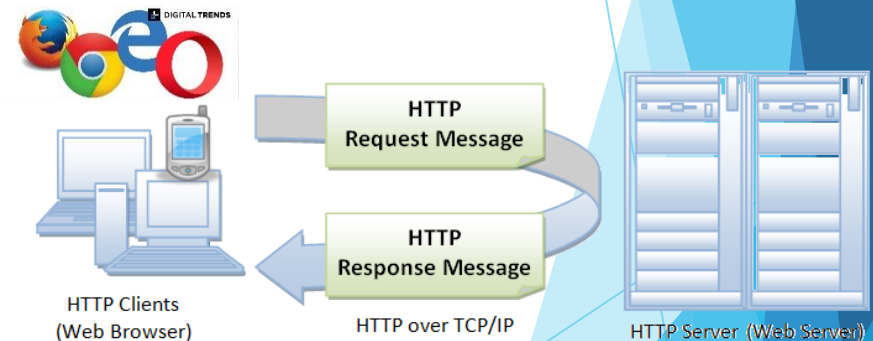
# HTTP: HyperText Transfer Protocol



- Over IP
- Over TCP
- Application Protocol HTTP
- Administrative Port for the official server : 80

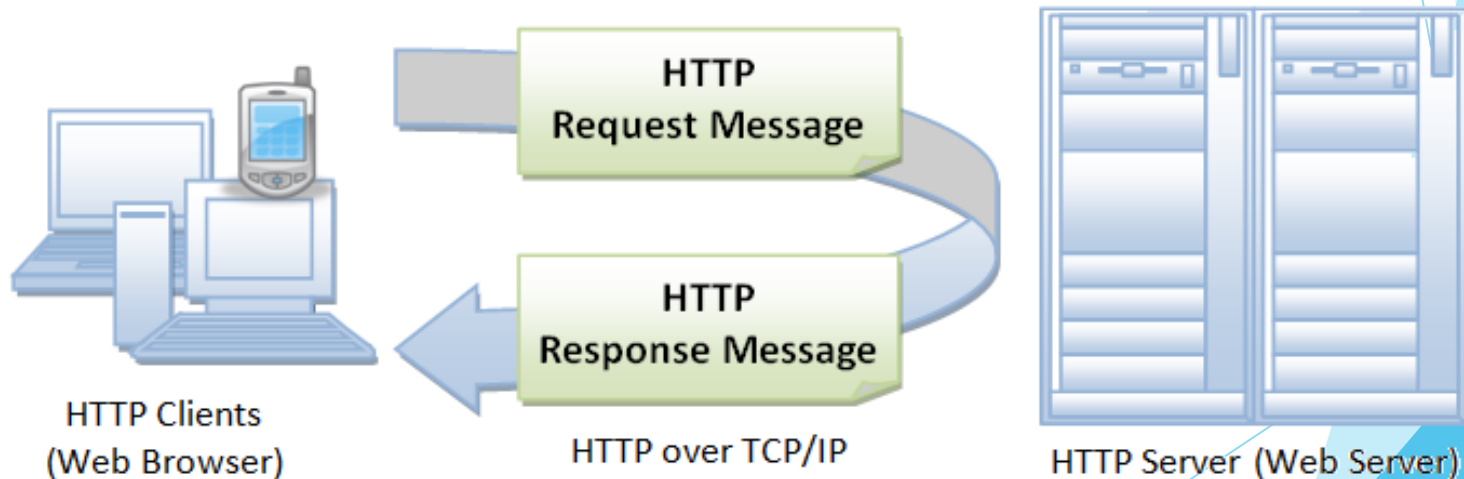
HTTP Protocol

**GET** is used to request data from a specified resource.  
**GET** is one of the most common **HTTP** methods.  
**POST** is used to send data to a server to create/update a resource.  
**POST** is one of the most common **HTTP** methods.  
**PUT** is used to send data to a server to create/update a resource.



# HTTP (Hypertext Transfer Protocol)

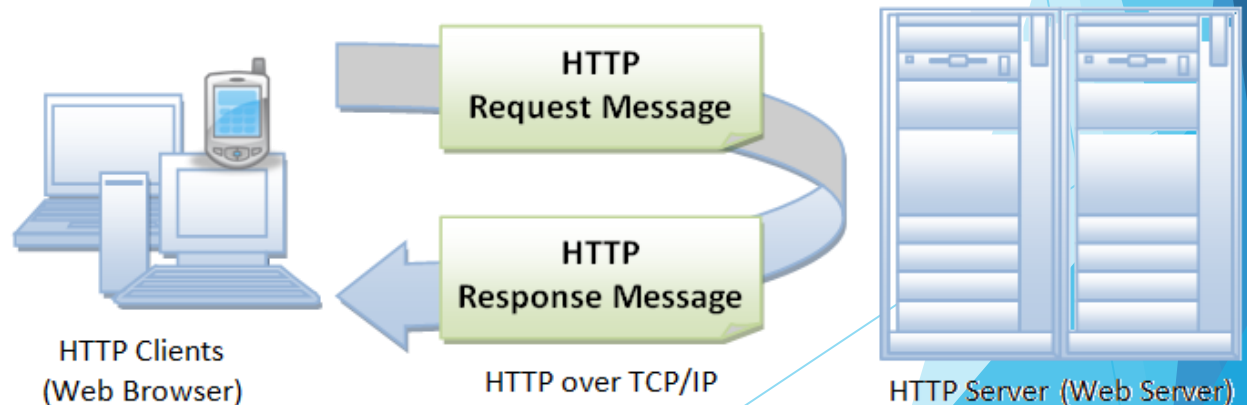
- ▶ HTTP (Hypertext Transfer Protocol) is perhaps the most popular application protocol used over TCP/IP (for the WEB).





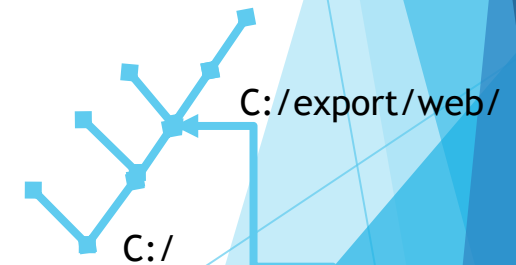
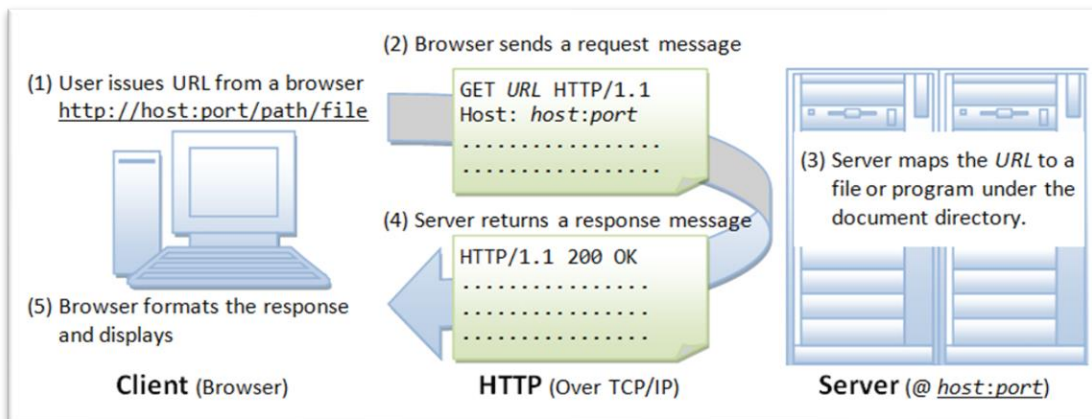
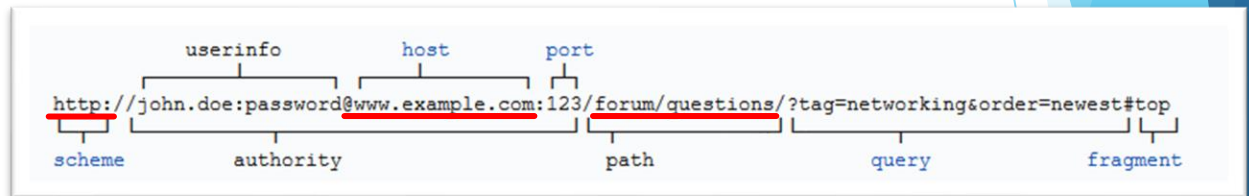
# HTTP (Hypertext Transfer Protocol)

- ▶ HTTP is an asymmetric request-response client-server protocol as illustrated.
  - ▶ An HTTP client sends a request message to an HTTP server.
  - ▶ The server, in turn, returns a response message.
- ▶ HTTP is a stateless protocol. In other words, the current request does not know what has been done in the previous requests.



# Browser Web is the HTTP Client

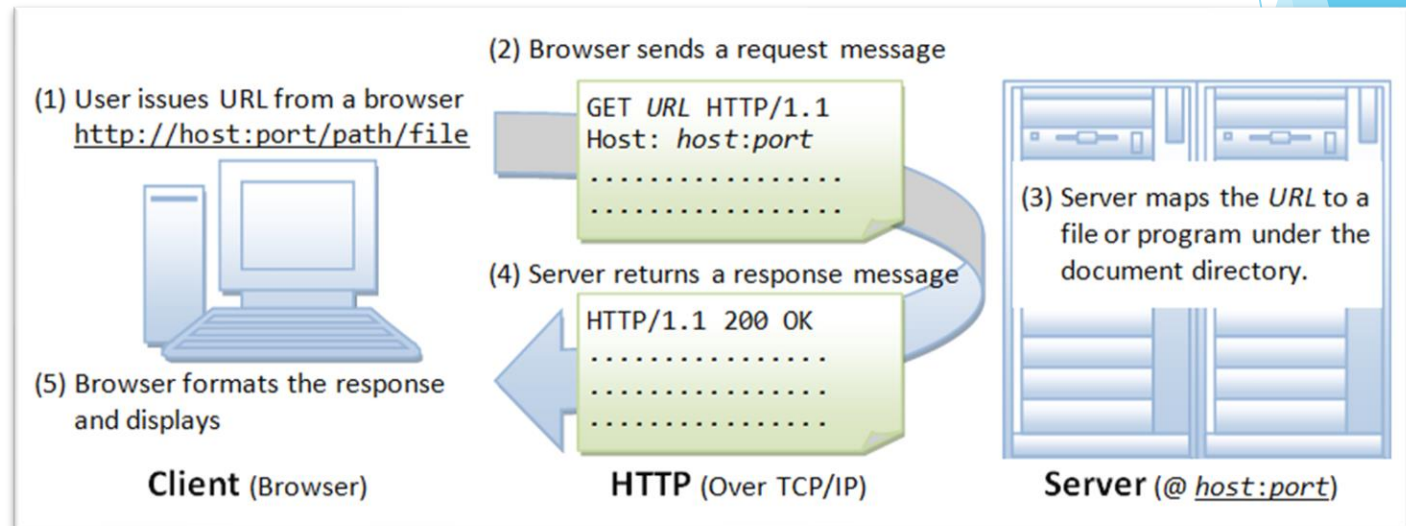
- ▶ Whenever you issue a URL from your browser to get a web resource using HTTP, e.g.  
<http://www.example.com/forum/questions/index.html>, the browser turns the URL into a request message and sends it to the HTTP server.



\$WebRoot ou \$HTTPRoot =  
 C:/export/web/  
 Et donc /forum/questions dans l'url, correspond à  
 C:/export/web /forum/questions sur le disque du  
 serveur

# Browser Web is the HTTP Client

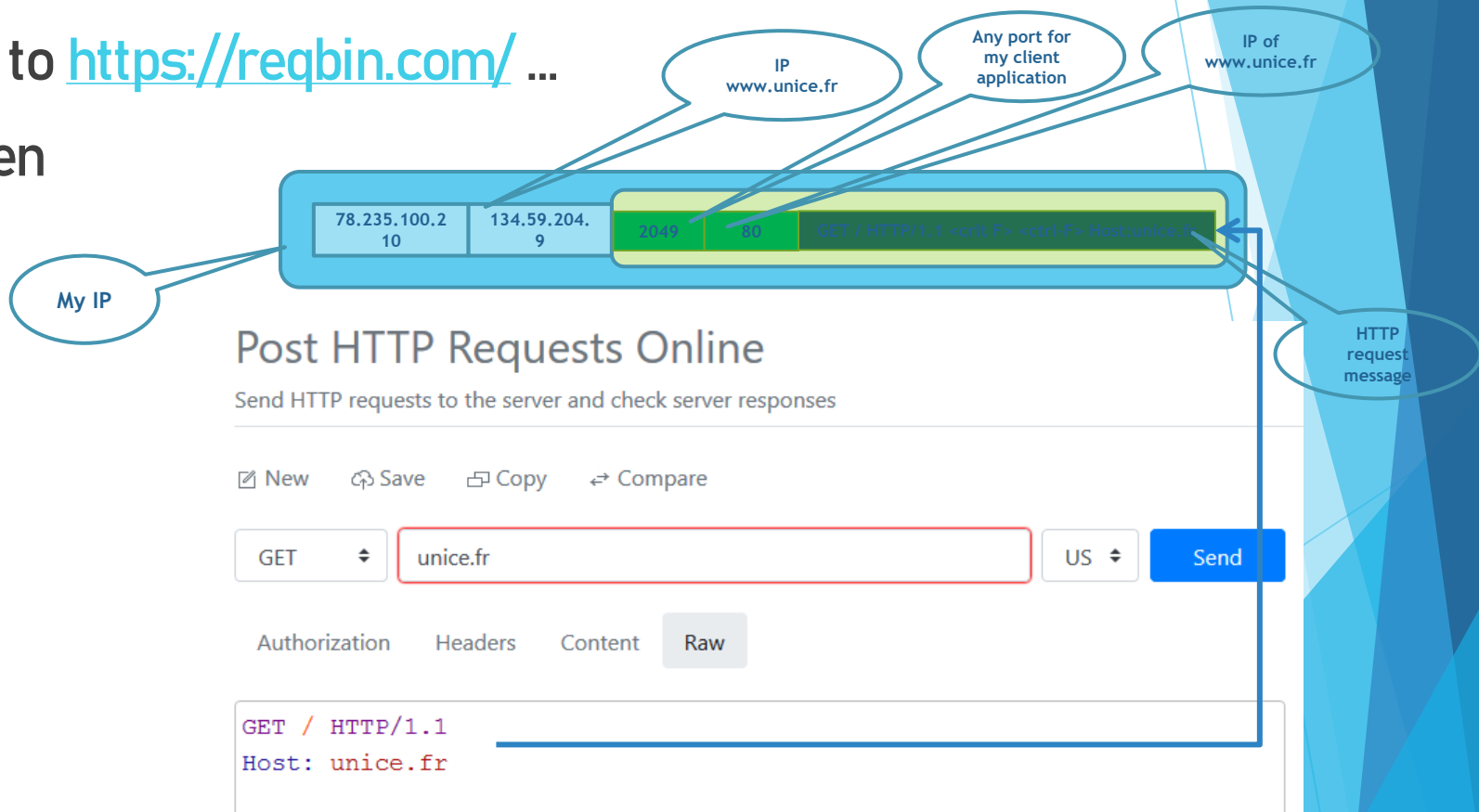
- ▶ The HTTP server interprets the request message, and returns you an appropriate response message, which is either the resource you requested or an error message.
- ▶ This process is illustrated below:



# GET Command sample

► Go to <https://reqbin.com/> ...

► Then



The screenshot shows the 'Post HTTP Requests Online' interface. Annotations explain the fields in the request builder:

- My IP:** Points to the 'From' field containing '78.235.100.2 10'.
- IP www.unice.fr:** Points to the 'To' field containing '134.59.204.9'.
- Any port for my client application:** Points to the 'Port' field containing '2049'.
- IP of www.unice.fr:** Points to the 'Host' field in the request message containing 'Host:unice.fr'.
- HTTP request message:** Points to the entire request text area.

The request text area shows the following command:

```
GET / HTTP/1.1 <ctrl-F> <ctrl-F> Host:unice.fr
```

Below the input fields, the 'Send' button is visible. The 'Raw' tab is selected, showing the raw HTTP request:

```
GET / HTTP/1.1
Host: unice.fr
```

# GET Command sample

- ▶ Answer from the HTTP server ([www.unice.fr](http://www.unice.fr))

- ▶ Content of this answer 

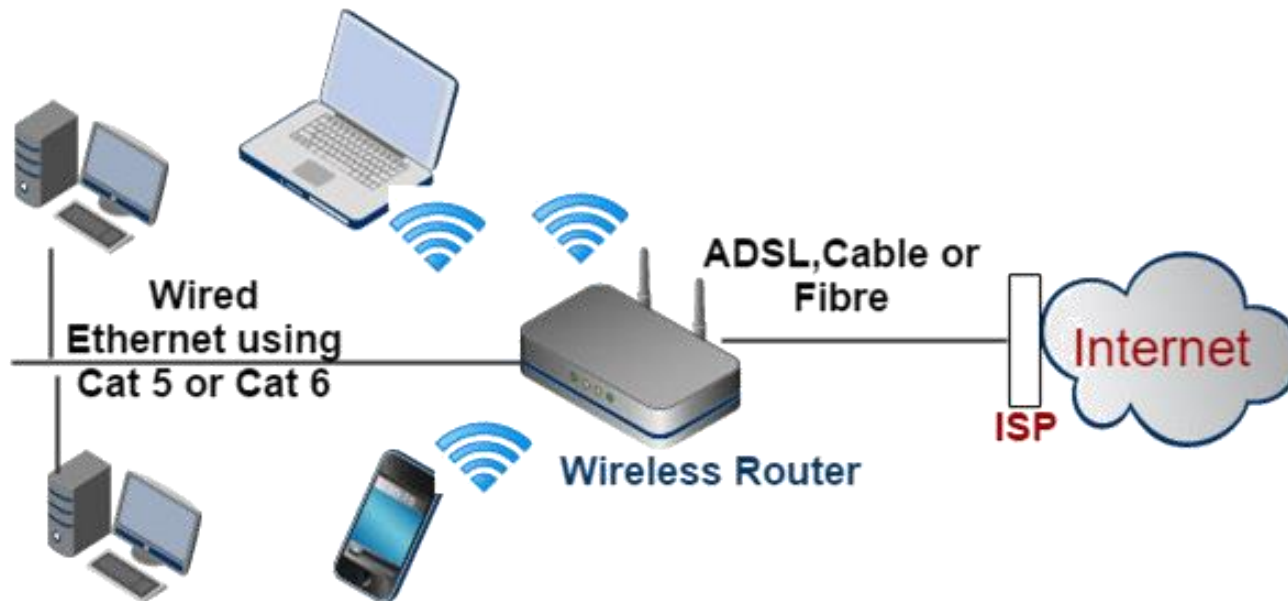
```
HTTP/1.1 200 OK
Server: nginx/1.6.3
Date: Mon, 23 Mar 2020 12:10:06 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Cache-Operation: plone.app.caching.noCaching
Content-Language: fr
Expires: Fri, 26 Mar 2010 12:10:06 GMT

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">
  <head><base href="http://unice.fr/front-page" /><base href="http
    <meta charset="utf-8" />
    <!--<meta name="viewport" content="width=device-width, initi
    <meta name="viewport" content="width=device-width, initial-s
    <meta name="description" content="" />
    <meta name="author" content="" />
    <link rel="shortcut icon" href="/++theme++ThemeUNS/assets/ic

    <title>Bienvenue à l'Université Nice Sophia Antipolis – Univ
    jQuery(function($){
      $.datepicker.setDefaults(
```

# Your Network at Home

- ▶ Box is composite device



# Some commands ...

- ▶ ipconfig / ifconfig
- ▶ traceroute
- ▶ route
- ▶ nslookup
- ▶ ... and more sophisticated like [Wireshark](#)