Principles of Networks and the Internet

Bloc Réseaux – Session 1

OUTLINE

- What is the Internet?
- Internet Architecture
- How the Internet Works
- Making Communicating Applications

OUTLINE

- What is the Internet?
  a) The **User**'s Point of View
  b) Technical Point of View
  c) Services & **Applications** Point of View
- Internet Architecture
- How the Internet Works
- Making Communicating Applications

What is the Internet?
a) the User's Point of View

- Name your favourite apps
  - Messaging, entertainment, calendar, information, …
  - Do these apps need the network?
- Sometimes without even realizing it
  - Do you know where your photos are stored (local vs. cloud)?
  - What about your private messages, where do they go, in whose hands, relayed or stored on servers?

What is the Internet?
b) Technical Point of View

Physical hardware

- **Equipment**
  - Switches, Routers, Access Points (Wi-Fi).
- **Machines**
  - Terminals (your laptop, smartphone, printer), servers.
  - “there is no cloud, just someone else computer” 😉
- **Link**
  - Cables (ethernet), fiber-optic (light), radio (wireless communication).

What is the Internet?
b) Technical Point of View

Addresses, Protocols, and Standards

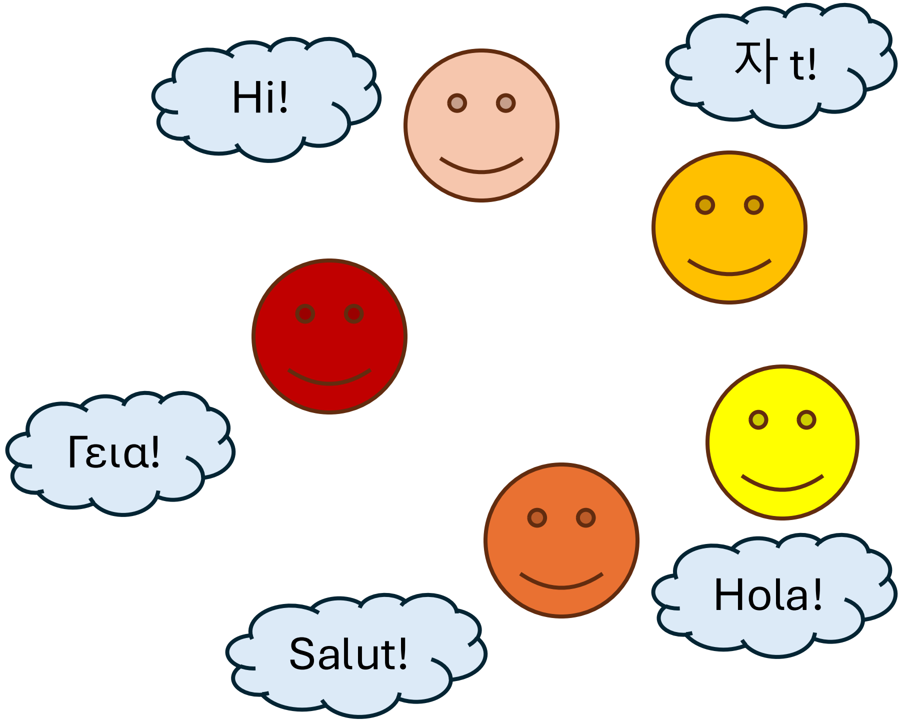
- **Addresses**
  - are employed to find each other.
  - designate your correspondent in an unambiguous way.

WHAT IS THE INTERNET?

b) Technical Point of View

Addresses, Protocols, and Standards

- **Addresses**
  - are employed to find each other.
  - designate your correspondent in an unambiguous way.
- **Protocols**
  - find ways to solve problems.
  - speak the same language, understand each other.



WHAT IS THE INTERNET?

b) Technical Point of View

Addresses, Protocols, and Standards

- **Addresses**
  - are employed to find each other.
  - designate your correspondent in an unambiguous way.
- **Protocols**
  - find ways to solve problems.
  - speak the same language, understand each other.
- **Standards**
  - write everything down in black and white, as precisely as possible!

```
Network Working Group                                    T. Berners-Lee
Request for Comments: 1945                                      MIT/LCS
Category: Informational                                      R. Fielding
                                                               UC Irvine
                                                              H. Frystyk
                                                                 MIT/LCS
                                                                May 1996


                Hypertext Transfer Protocol -- HTTP/1.0

Status of This Memo

   This memo provides information for the Internet community.  This memo
   does not specify an Internet standard of any kind.  Distribution of
   this memo is unlimited.

IESG Note:

   The IESG has concerns about this protocol, and expects this document
   to be replaced relatively soon by a standards track document.

Abstract

   The Hypertext Transfer Protocol (HTTP) is an application-level
   protocol with the lightness and speed necessary for distributed,
   collaborative, hypermedia information systems. It is a generic,
   stateless, object-oriented protocol which can be used for many tasks,
   such as name servers and distributed object management systems,
   through extension of its request methods (commands). A feature of
   HTTP is the typing of data representation, allowing systems to be
   built independently of the data being transferred.

   HTTP has been in use by the World-Wide Web global information
   initiative since 1990. This specification reflects common usage of
   the protocol referred to as "HTTP/1.0".

Table of Contents

   1.  Introduction ............................................  4
       1.1  Purpose ............................................  4
       1.2  Terminology ........................................  4
       1.3  Overall Operation ..................................  6
       1.4  HTTP and MIME ......................................  8
   2.  Notational Conventions and Generic Grammar ..............  8
       2.1  Augmented BNF ......................................  8
       2.2  Basic Rules ........................................ 10
   3.  Protocol Parameters ..................................... 12


Berners-Lee, et al           Informational                      [Page 1]
```

WHAT IS THE INTERNET?
b) Technical Point of View

Interconnection Principle

- Inter-net (inter networking):
  - **Interconnection of computer networks**.
  - Agreement on the same address space, protocols, and standards.
- Different organizational levels:
  - corporate, operator, national, international, private networks.
- Vocabulary:
  - "**i**nternet" = any interconnected network of computers (e.g., intranet).
  - "**I**nternet" = the global public network (e.g., browsing, emailing, streaming).
  - (and Internet ≠ Web, see chapter on applications)

WHAT IS THE INTERNET?

c) Services & Applications Point of View

Code!

- The code is executed on which machines?
  - on servers, on user terminals (smartphone).
- Where in the network?
  - at the edge, in your pocket, in data centers.
  - not in the heart of the network! (≠ ☎).
- Who runs (is responsible for) these applications?
  - the service provider who has rented a server in a datacenter.
  - the user who launches the application (or runs it on his own in the background).
- How do you program a communicating application?
  - using APIs (Application Programming Interface), which is a type of software interface.

OUTLINE

- What is the Internet?
- Internet Architecture
  a) The Actors
  b) Management and Coordination
  c) The Interconnection
- How the Internet Works
- Making Communicating Applications

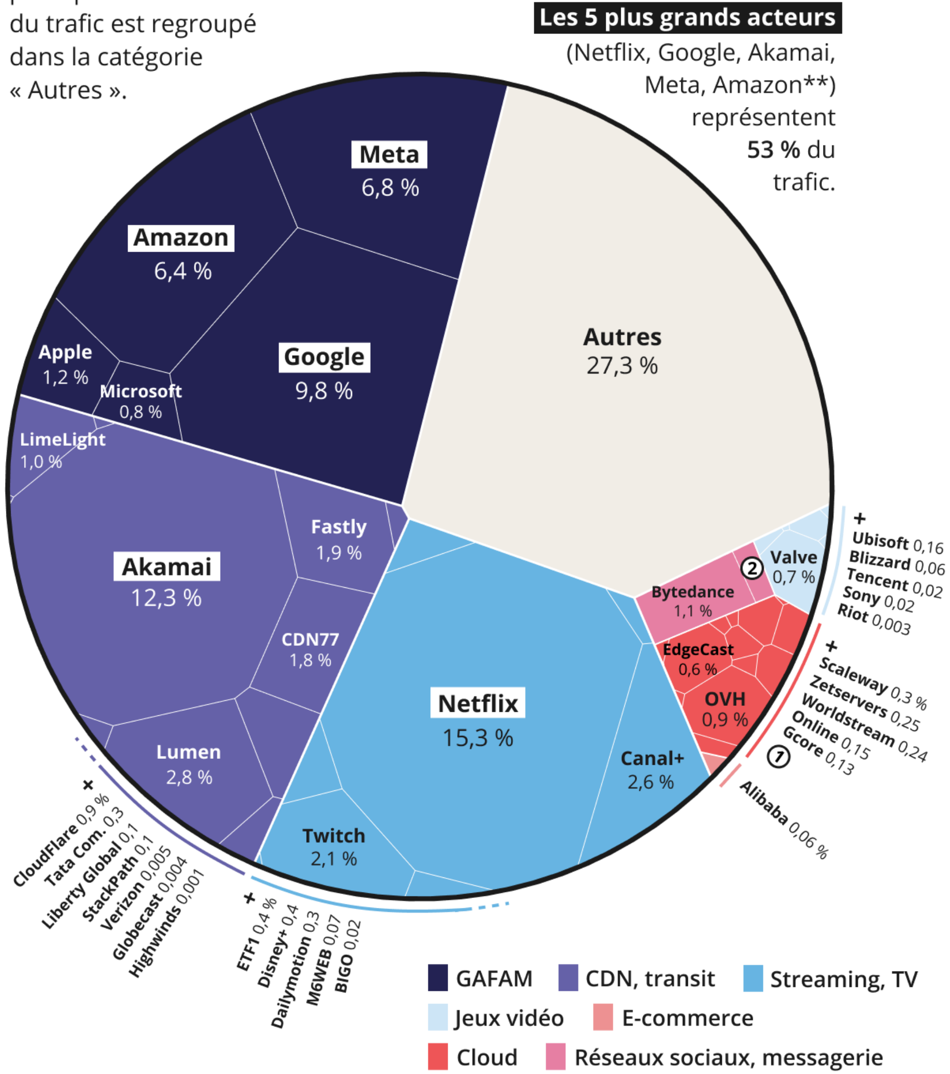
Internet Architecture

a) The Actors (1/2)

- **Content Providers**:
  - Produce, collect, deliver content, services. business applications.
  - Infrastructures: datacenters, CDN.
  - e.g., GAFAMs, among others.
- **Internet Service Providers** (ISPs):
  - Individuals, businesses.
  - e.g., Free, SFR, Orange, Bouygues.
- **Multi-role Players**:
  - Content + data centers + network + energy.
  - e.g., Amazon, Google, Akamai, Orange, among others.

DÉCOMPOSITION SELON L'ORIGINE DU TRAFIC VERS LES CLIENTS DES PRINCIPAUX FAI EN FRANCE (FIN 2023)

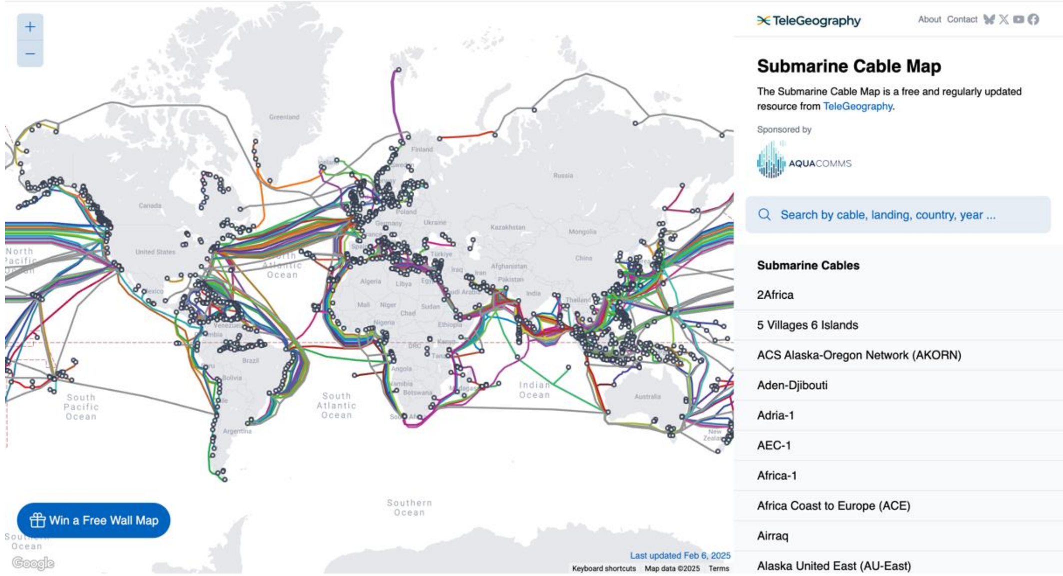
Pourcentage du trafic entrant au point d'interconnexion de 47 acteurs liés au transport ou à la production de contenu déclarés par les principaux FAI* fin 2023. Le reste du trafic est regroupé dans la catégorie « Autres ».



Source : Arcep.
* Bouygues, Free, Orange, SFR. ** 53 % en incluant Twitch, qui appartient à Amazon.
① **M247** 0,07 % ; **Dropbox** 0,04 ; **Dstorage** 0,03 ; **LeaseWeb** 0,03 ; **SdV Plurimédia** 0,003 ; **Mediactive** 0,002.
② **Telegram** 0,2 %.

Internet Architecture

a) The Actors (2/2)

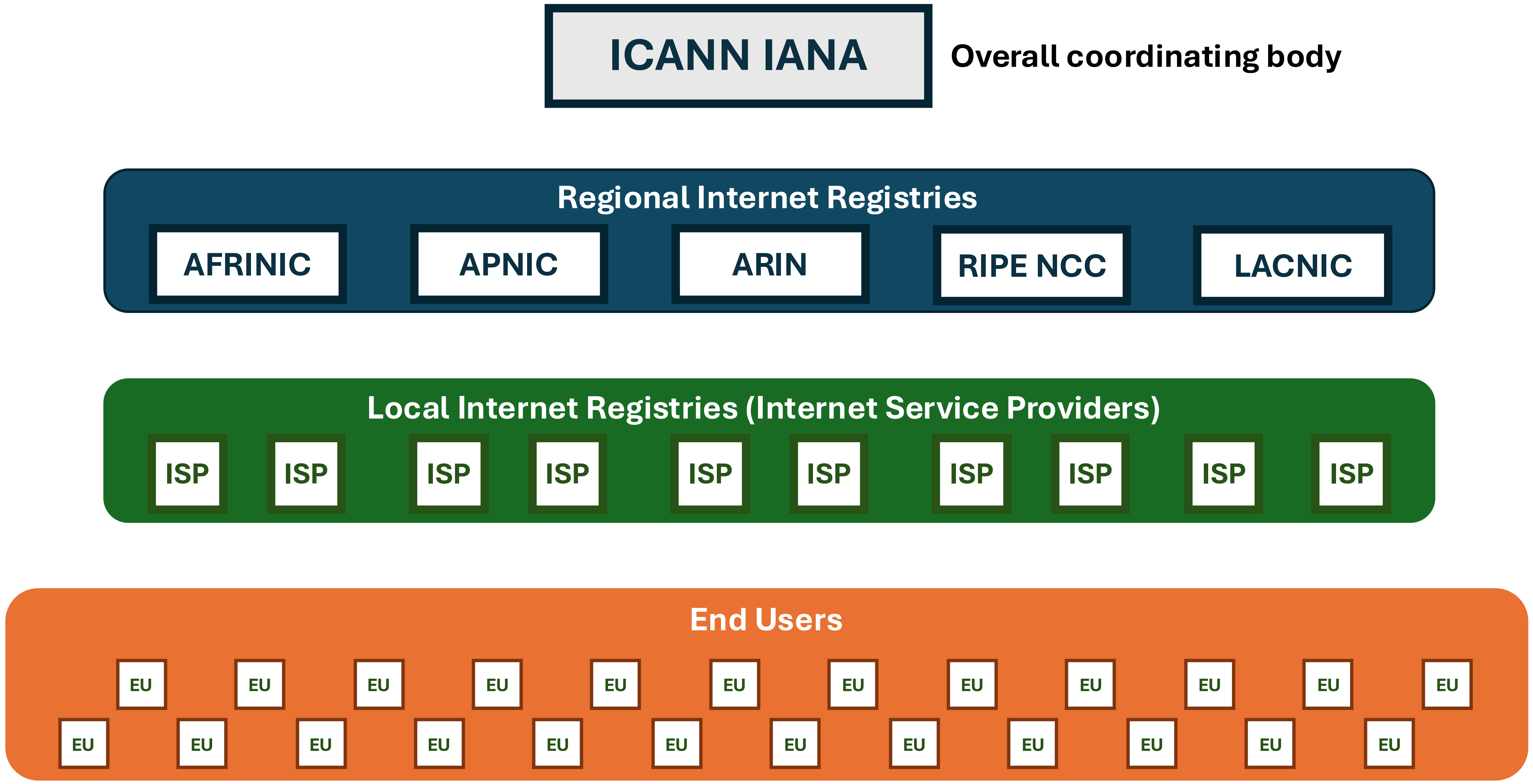


- **Core Network Operators**:
  - Pulling cables, infrastructures (e.g., data centers).
  - Metropolitan, regional, national, intercontinental.
  - **Tier 1 ISPs** (e.g., Lumen, Orange, NTT), **IXPs** (e.g., France-IX), **submarine cable operators** (e.g., Alcatel Submarine Networks) and **CDN providers** (e.g., Akamai, OVHcloud, AWS).
- **Standardization Bodies**:
  - **IEEE**: industrial consortium, electronics.
  - **IETF**: working groups, protocols, open, RFCs.
- **Management and Coordination** ICANN IANA:
  - See next slide!

Photo taken during a visit of Orange 5G Lab Châtillon at Orange Gardens.

Internet Architecture
b) Management and Coordination

- **ICANN** (Internet Corporation for Assigned Names and Numbers): a global organization that **coordinates** the maintenance and procedures of **databases related to the namespaces and numerical spaces of the Internet**.
- **IANA** (Internet Assigned Numbers Authority): a function of ICANN that oversees:
  - **Number Resources**: coordination of the **allocation** of the global pool of **IP addresses** and **Autonomous System (AS) numbers**, **delegating** them to **RIRs**.
  - **Domain Names**: root zone management in the Domain Name System (DNS), which includes all Top-Level Domains (TLDs) such as generic TLDs (gTLDs) like .com, .org, .net and country-code TLDs (ccTLDs) like .fr (France), and delegates the responsibility of managing each TLD to specific **registry operators**, e.g., **Afnic** is the registry operator for .fr, and thus it sets rules and policies.
  - **Protocol Assignments**: Internet protocols' numbering systems are managed in conjunction with standards bodies (e.g., IETF).
- **RIR** (Regional Internet Registry): organization that receive IPs and ASNs from IANA **and distribute them regionally** (e.g., RIPE NCC for Europe).
- **LIR** (Local Internet Registry): **ISPs** or **large organizations** (i.e., **RENATER**) that obtain IPs from an RIR and assign them to customers. Examples of LIR in France: www.ripe.net/membership/member-support/list-of-members/fr/
  - **ISP** (Internet Service Provider): A company that provides Internet access to users and businesses, often acting as an LIR.
- **EU** (End User): Individuals or businesses that receive Internet access and IP addresses from ISPs or LIRs.


ICANN IANA
Overall coordinating body
Regional Internet Registries
AFRINIC
APNIC
ARIN
RIPE NCC
LACNIC
Local Internet Registries (Internet Service Providers)
ISP
ISP
ISP
ISP
ISP
ISP
ISP
ISP
ISP
ISP
End Users
EU

INTERNET ARCHITECTURE

c) The Interconnection

An "Autonomous System"

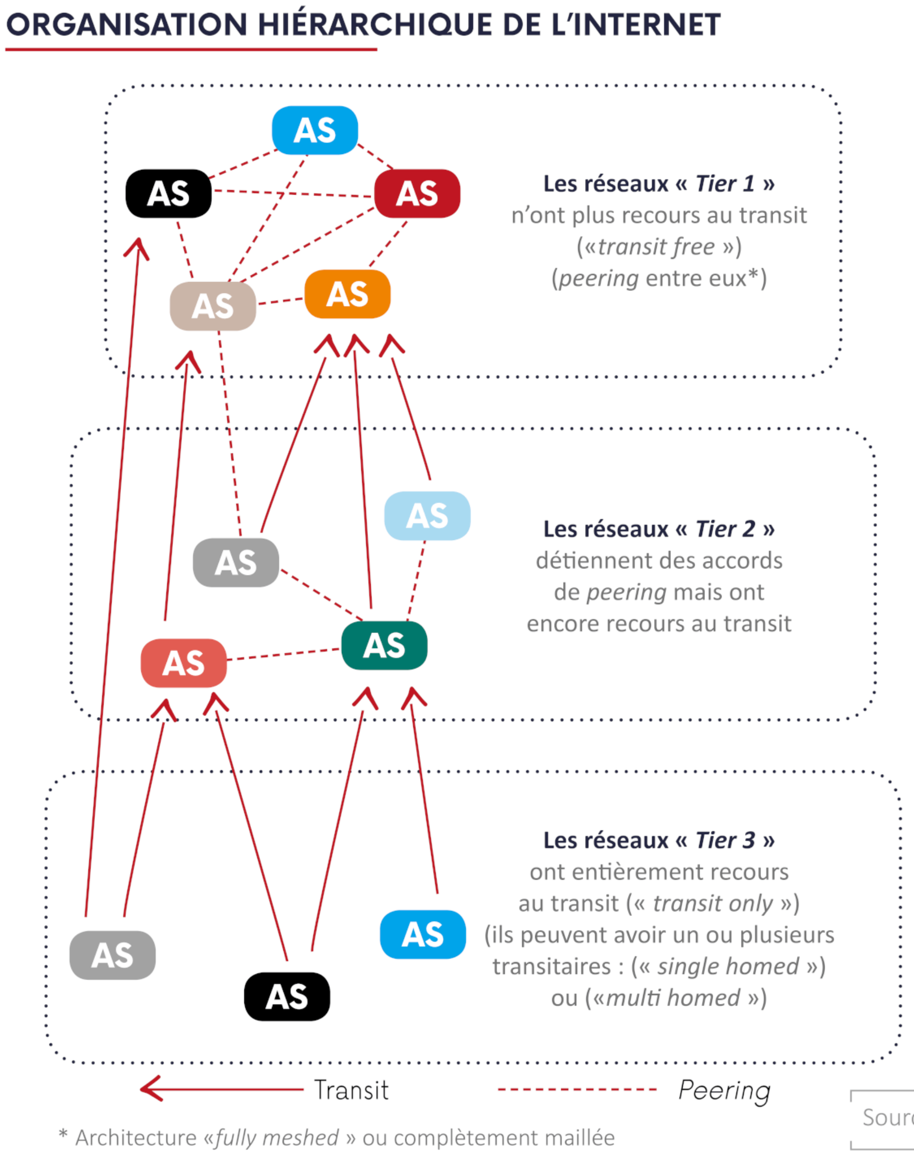
- A **self-governed (*autogéré*) large network or group of networks** under a single technical administration that follows a unified routing policy, i.e., **BGP**.
- An **AS has an AS number and a block of IP addresses** assigned by a **RIR**.
- Decides with whom it interconnects (through BGP routing protocol).
  - Announces to other ASes the routes to the networks and users it serves.
  - Note: Individuals and companies are not ASes (in general), but interconnect with them.
- **Internet = ∑ AS**
- For examples of ASes in France, visit: www.whatismyip.com/asn/country/fr/

Internet Architecture

c) The Interconnection

Summary

- **Peering:**
  - Direct AS-to-AS Peering (Private Peering).
  - On third-party premises (e.g., IXPs, Data center).
- **Transit:**
  - One AS pays another AS to provide a service.
- **AS Hierarchical Organisation:**
  - Tier-1: peering only.
  - Tier-2: peering + transit.
  - Tier-3: only transit, mainly with Tier-2.



Source : Arcep.

INTERNET ARCHITECTURE
c) The Interconnection

Interconnection between ASs

- **Transit:**
  - One AS (**customer**) pays another AS (**provider**) for access to the wider internet.
  - **Scenario**: A smaller local ISP (Tier 3) buys transit (connectivity) from larger ISP (Tier 2 or Tier 1) to reach the global internet.
  - **Example**: AS12322 (Free ISP - France) pays AS3215 (Orange France) for internet access, which in turn buys transit from AS3356 (Lumen, a Tier 1 ISP) to reach global networks.
  - **Analogy**: Transit is like paying a **toll** (*le péage*) to use a highway network owned by another company, , e.g., Rennes – Paris.

Internet Architecture
c) The Interconnection

Interconnection between ASs

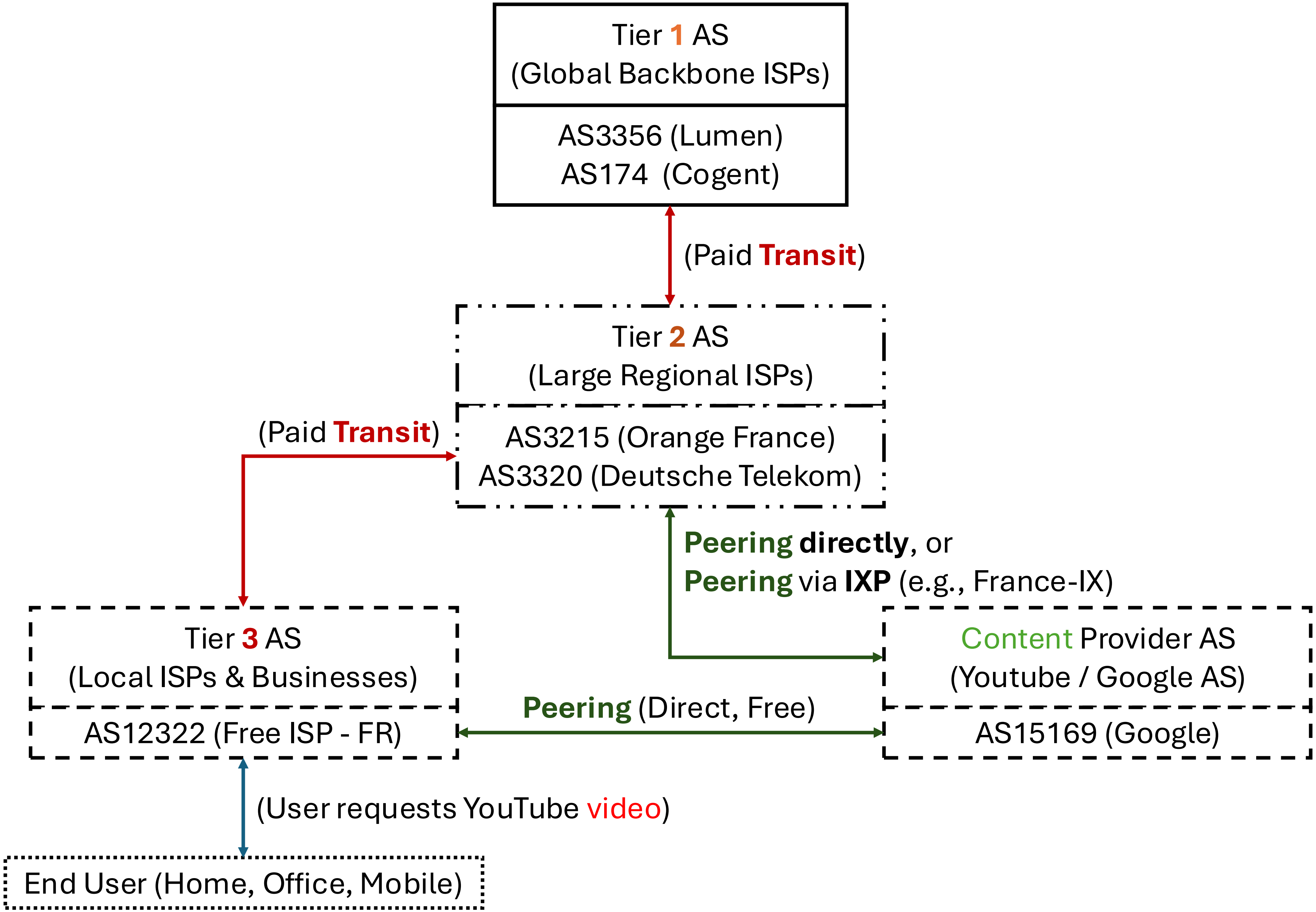
- **Peering:**
  - Two ASes agree to exchange traffic **directly between their own networks** for free **or** at a low cost.
  - Usually occurs at **Internet Exchange Points (IXPs)*** to optimize performance & reduce costs.
  - **Scenario**: Tier 2 ISPs, CDNs, and large companies peer with each other to improve speed and reduce transit costs.
  - **Example:** AS3215 (Orange France) peers with AS3320 (Deutsche Telekom) at an IXP.
  - **Analogy:** Peering is like two neighbouring towns building a free road between them to avoid paying a highway toll, e.g., Rennes – Nantes.

Internet Architecture
c) The Interconnection

Interconnection between ASs

- **Peering:**
  - Two ASes agree to exchange traffic **directly between their own networks** for free **or** at a low cost.
  - Usually occurs at **Internet Exchange Points (IXPs)*** to optimize performance & reduce costs.
  - **Scenario**: Tier 2 ISPs, CDNs, and large companies peer with each other to improve speed and reduce transit costs.
  - **Example:** AS3215 (Orange France) peers with AS3320 (Deutsche Telekom) at an IXP.
  - **Analogy:** Peering is like two neighbouring towns building a free road between them to avoid paying a highway toll, e.g., Rennes – Nantes.

- *An IXP (France-IX in France) acts as a **central hub** where multiple ASes connect to a shared switching fabric, allowing them to efficiently exchange traffic.


Tier 1 AS
(Global Backbone ISPs)
AS3356 (Lumen)
AS174 (Cogent)
(Paid Transit)
Tier 2 AS
(Large Regional ISPs)
AS3215 (Orange France)
AS3320 (Deutsche Telekom)
(Paid Transit)
Peering directly, or
Peering via IXP (e.g., France-IX)
Tier 3 AS
(Local ISPs & Businesses)
AS12322 (Free ISP - FR)
Peering (Direct, Free)
Content Provider AS
(Youtube / Google AS)
AS15169 (Google)
(User requests YouTube video)
End User (Home, Office, Mobile)

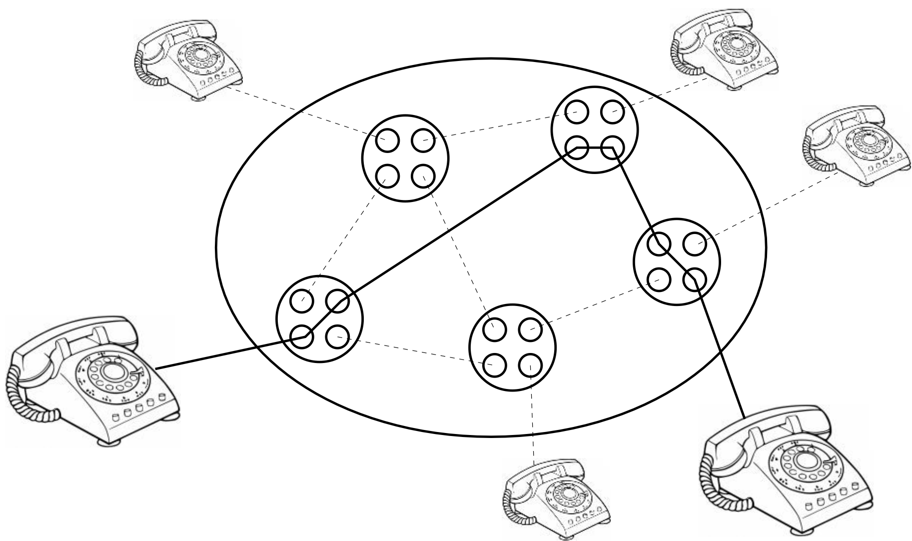
OUTLINE

- What is the Internet?
- Internet Architecture
- How the Internet Works
  a) Circuit vs Packet Switching
  b) The Protocols
  c) Layered Organization
  d) The Addresses
  e) Routing (Layer 3. Network)
  f) Transport of Information (Layer 4. Transport)
- Making Communicating Applications

How the Internet Works
a) Circuit vs Packet Switching (1/2)

Circuit Switching

- **Definition**: circuit switching establishes a dedicated communication path between two devices before data transmission begins.
- **Example**: traditional landline telephone (networks) calls.
- **Advantages**:
  - Guaranteed bandwidth and quality of service (QoS).
  - No data loss since the path is dedicated.
- **Disadvantages**:
  - Inefficient for bursty data traffic since the circuit remains reserved even when no data is being sent.
  - Higher setup time before transmission starts.



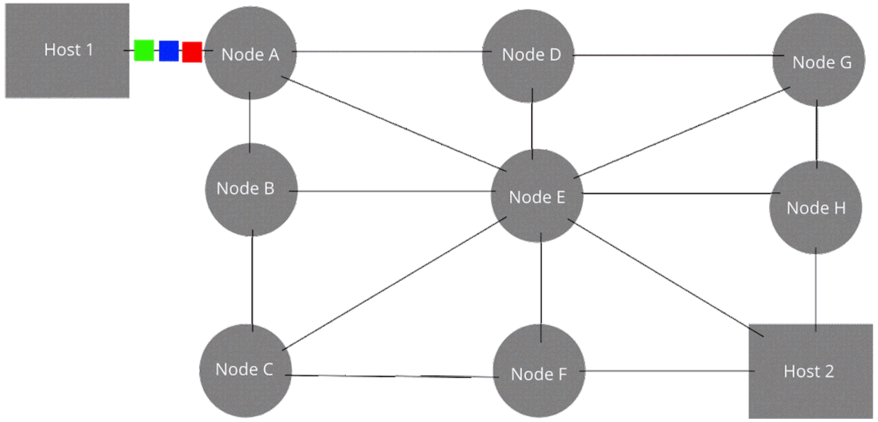
How the Internet Works

a) Circuit vs Packet Switching (2/2)

Packet Switching

- **Definition**: packet switching divides data into small packets and transmits them independently over a shared network. Each packet may take a different route to reach the destination, where they are reassembled.
- **Example**: Internet, where emails and web traffic use packet-switched networks.
- **Advantages**:
  - Efficient use of network resources, as bandwidth is shared.
  - Faster and more scalable for large amounts of data.
- **Disadvantages**:
  - Packets may arrive out of order or get lost, requiring reassembly and error correction.
  - No guaranteed bandwidth, which can lead to congestion and variable latency.

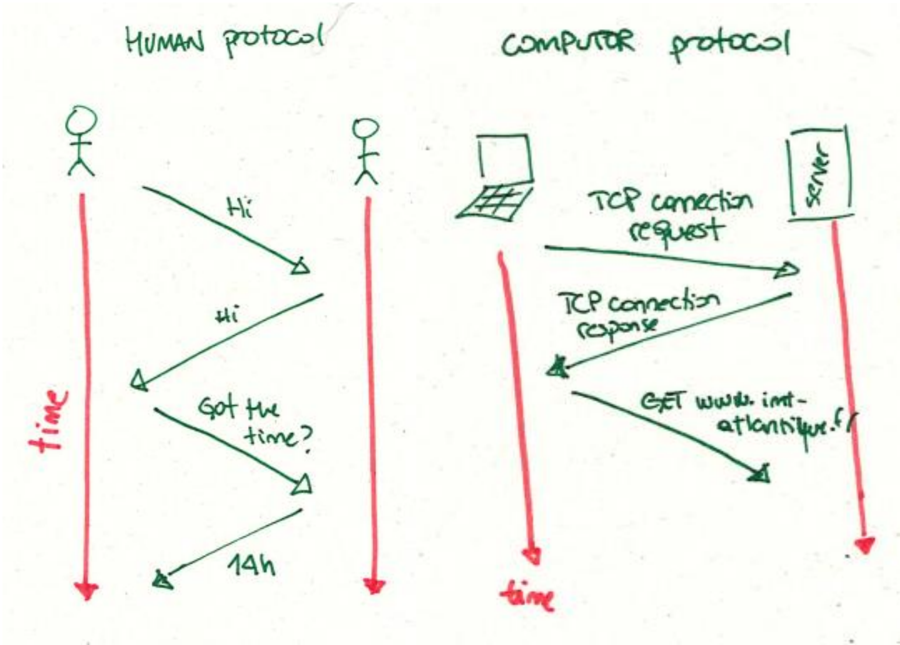
The original message is Green, Blue, Red.



How the Internet Works

b) The Protocols

- **Definition**: a protocol is a set of rules that enables two or more devices to transmit and receive data across a network. Moreover, it ensures **authentication**, **data integrity**, and **confidentiality**.
  - Some of the rules that a protocol defines are related to data format (e.g., XML, JSON), data encoding (UTF-8, ASCII), error detection and recovery, packet losses, packet re-ordering, network congestion, clocks synchronisation.
- Involves:
  - Information to be carried in the transmitted data.
  - Information to be memorised by the devices.
- A protocol may be **implemented** by **hardware**, **software**, or a combination of **both**.
- Separate the problems (see next slide)!

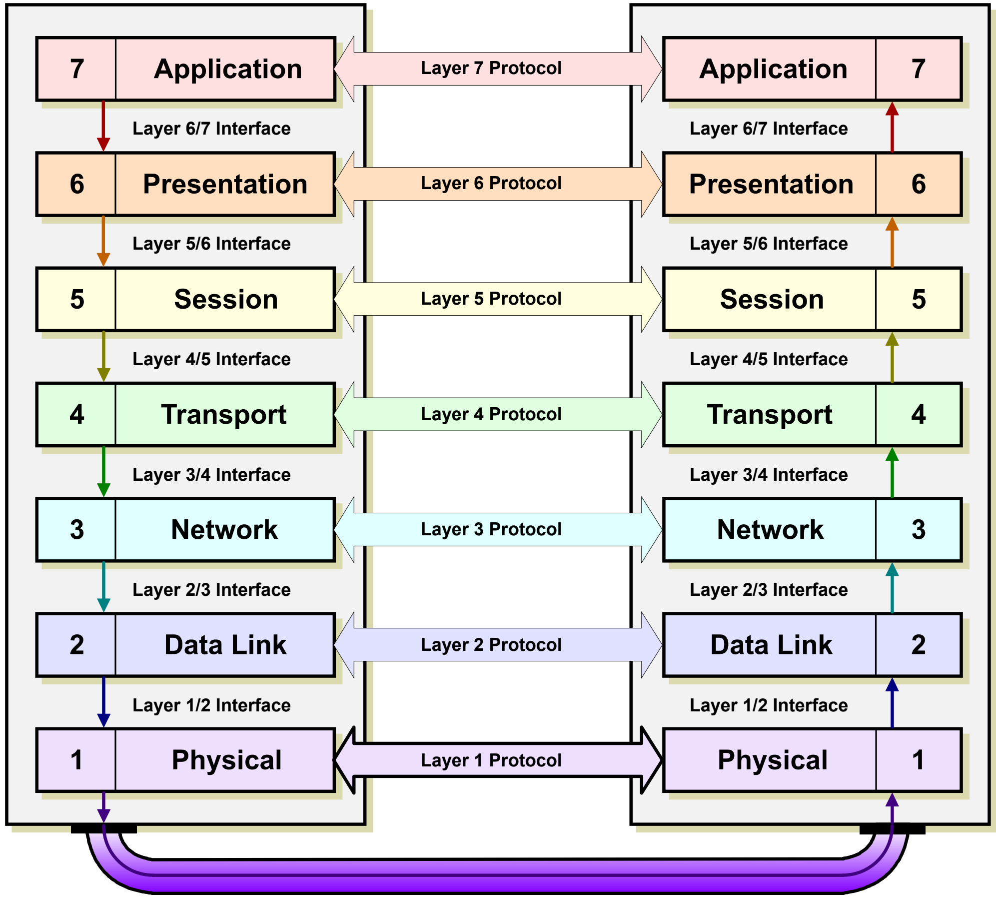


How the Internet Works

c) Layered Organization (1/5)

OSI (Open Systems Interconnection) conceptual Model

- Defines 7 layers with precise roles
  - 1. Physical, 2. Data Link, 3. Network, 4. Transport, 5. Session, Presentation, and 7. Application
- Layered abstraction – each:
  - solves problems at its level – protocols.
  - offers services to layer N+1.
  - uses services from layer N-1.
  - talks to its remote peer.



HOW THE INTERNET WORKS

c) Layered Organization (2/5)

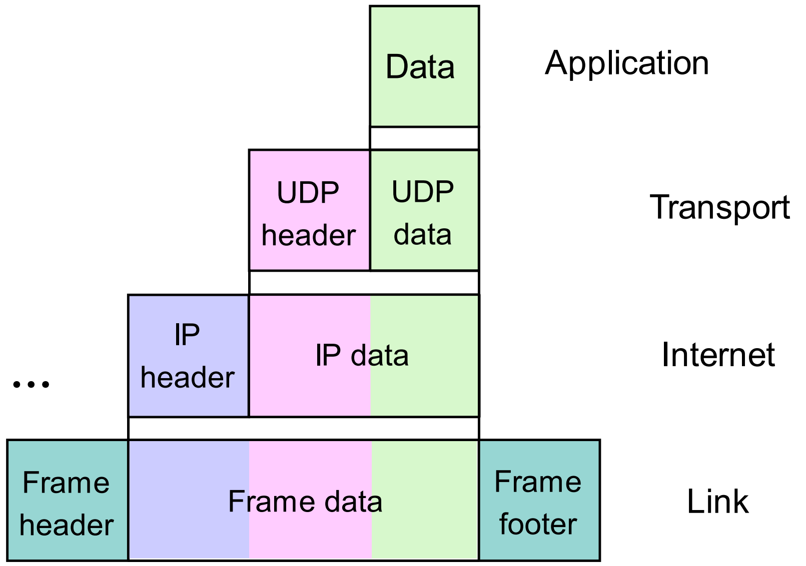
The 7 layers of the OSI model

- **7. Application**: enables user interaction with network services e.g., web, file transfer, email.
  - e.g., HTTP, IMAP, SMTP, DNS, SSH.
- **6. Presentation**: negotiate common data format and encoding between two ends.
  - e.g., XML, Json, CBOR, ASN.1.
- **5. Session**: establishes (synchronizes, authenticates), maintains, & ends communication sessions.
  - e.g., SIP, RTCP, PPTP.
- **4. Transport**: manages end-to-end communication, e.g., packet reordering, congestions control.
  - e.g., TCP, UDP.
- **3. Network**: determines the best path, forwards the packets from one relay node to another.
  - e.g., Addresses are required to identify devices (e.g., IPv4, IPv6), Routing Protocols (e.g., OSPF, BGP).
- **2. Data Link**: ensures error-free data transfer between adjacent nodes (i.e., point-to-point links).
  - Builds the Protocol Data Unit (PDU). Technologies: e.g., ethernet, Wi-Fi, FTTH. Protocols: e.g., ARQ, FEC.
- **1. Physical**: transmits raw bits over cables, fiber, or wireless signals. Clock synchronization.
  - Transform binary data (e.g., 11100111011010101) into physical signal e.g., electrical, light, radio.

How the Internet Works

c) Layered Organization (3/5)

From Wikipedia, by user Cburnett and Kbrose under the CC BY-SA 3.0
https://commons.wikimedia.org/w/index.php?curid=1546338



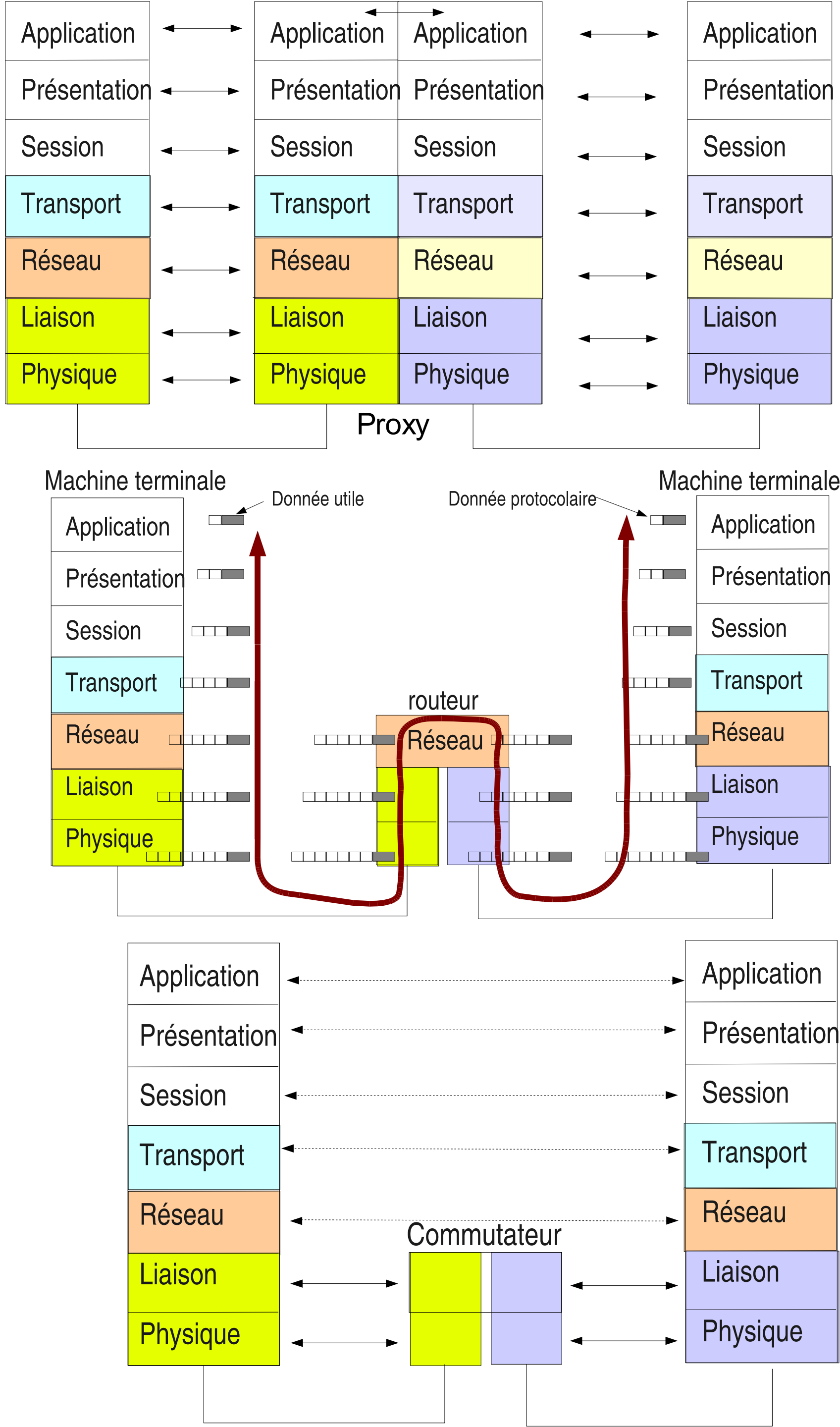
Encapsulation

- Each layer
  - adds *protocol-related* information …
  - … to the *useful* data before transmission over a network …
  - … and this is the **header**.
- **How it works**:
  - Application: data is generated by applications (e.g., emails, web requests).
  - Transport: adds a TCP or UDP header with port numbers and sequencing.
  - Network: adds an IP header with source and destination addresses.
  - Data Link: adds a MAC header and trailer to form a frame.
  - Physical: converts the frame into electrical, optical, or radio signals for transmission.
- On the receiving end, **decapsulation** occurs, where headers are removed layer by layer until the original data reaches the application.

How the Internet Works

c) Layered Organization (4/5)

- Terminal machines
  - The 7 layers
- Intermediate machines
  - Web, or Proxy server: Application
  - Routers: Network
  - Bridge/Switch: Data Link
  - AP Wi-Fi: Data Link
  - Hub: Physical
  - Repeater (Wi-Fi): Physical

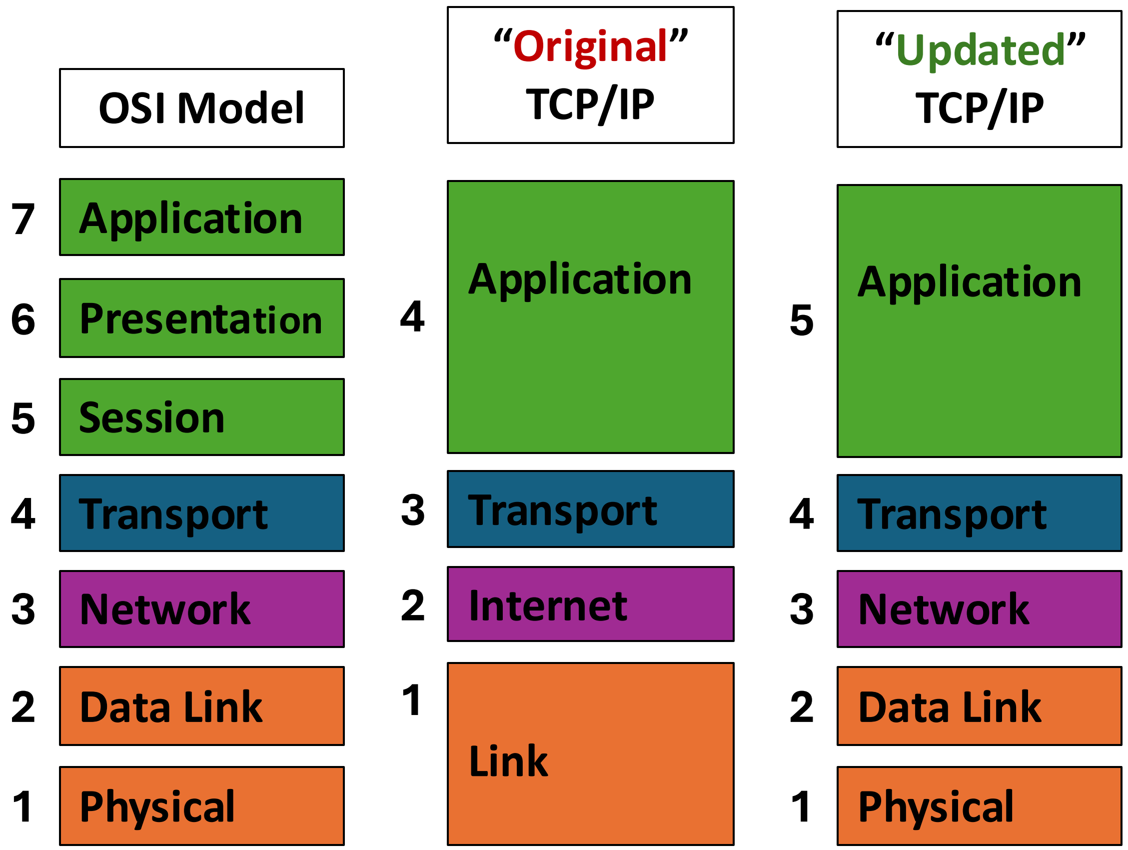


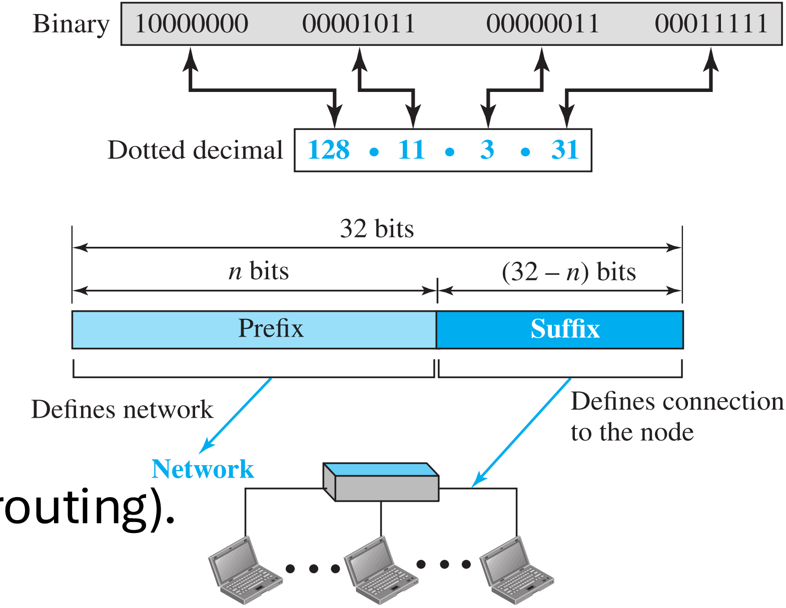
How the Internet Works

c) Layered Organization (5/5)

OSI Model and TCP/IP

- OSI:
  - Theoretical model.
  - Helps understanding, design.
- TCP/IP:
  - Concrete model
  - Aggregates upper layers to Application layer.
  - *Aggregates bottom layers to Link layer.*
  - "Almost" respects the separation of layers.





How the Internet Works

d) The Addresses (1/5)

General Principles

- Identifies a device in a network and allows global communication (routing).
- **Network Address (Prefix)**: identifies the entire network or subnet.
  - All devices in the same subnet share the same network portion (i.e., direct communication).
  - Reduces Routing Table size!
  - The host portion is all 0s in binary.
  - **Analogy**: a **street name** (e.g., "Main Street") in a city.
  - **Example** of a Network Address: **128.11.3.0** (i.e., the first address in the subnet).
- **Host Address (Suffix)**: uniquely identifies a device (host) within the network or subnet (*reminder*: ICANN IANA, RIR, LIR/ISP).
  - **Analogy**: a **house or building number on a street** (e.g., "**123** Main Street").
  - **Example** of a Host Address: **128.11.3.31**
- **Subnet Mask** / CIDR* **Prefix**: defines/distinguishes the network and host parts of an address.
  - **Example** of a Subnet Mask (obsolete): **128.11.3.31** Subnet Mask **255.255.255.0**
  - **Example** of a CIDR Prefix (number of Most Significant Bits): **128.11.3.31/24**

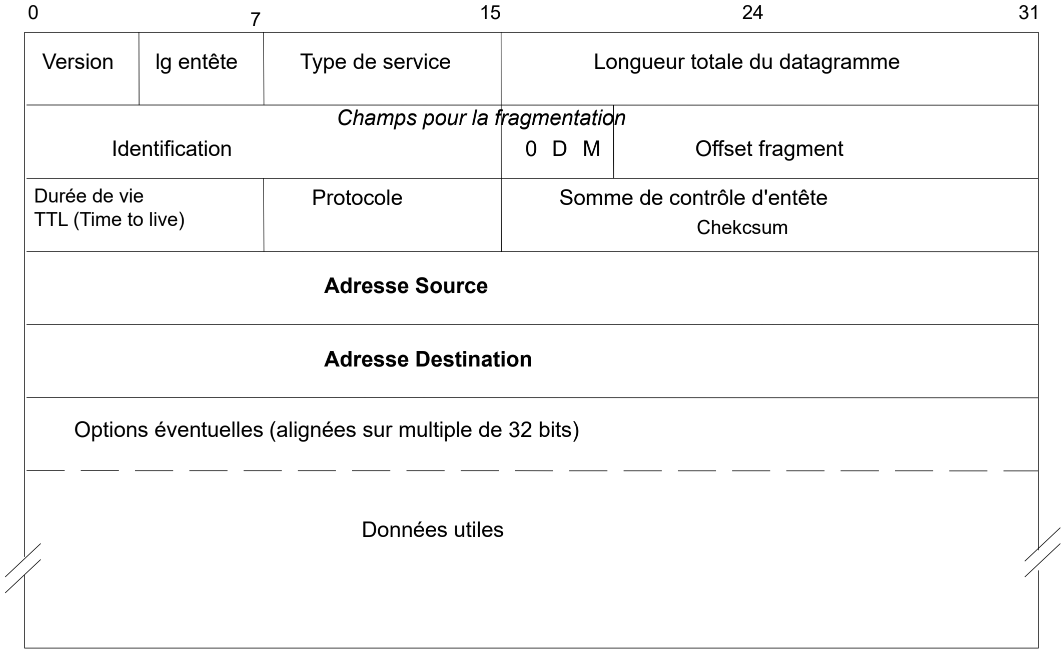
*Classless Inter-Domain Routing (CIDR)

How the Internet Works

d) The Addresses (2/5)

IPv4 (since 1983)

- Size: 32 bits
  - Dotted decimal notation.
  - e.g., **128.11.3.31**, a typical **Unicast** IPv4 address.
- **Classes** with Subnet Masks (legacy)
  - Class A /8 – Class B /16 – Class C /24
- **Multicast**, to a group number
  - Class D – 224.0.0.0 to 239.255.255.255
- **Broadcast**, to the entire local network
  - 255.255.255.255, or suffix to 255
  - eg.128.11.3.255
- **Configuration:**
  - by hand (e.g., randomly, hash) ...
  - dynamically from a **DHCP server**.



Class	Starting IP Range	Default Subnet Mask	CIDR Notation	Number of Hosts
A	1.0.0.0 – 126.0.0.0	255.0.0.0	/8	16,777,214, why?
B	128.0.0.0 – 191.255.0.0	255.255.0.0	/16	65,534, why?
C	192.0.0.0 – 223.255.255.0	255.255.255.0	/24	254, why?
D (multicast)	224.0.0.0 – 239.255.255.255	N/A	N/A	N/A
E (experimental)	240.0.0.0 – 255.255.255.255	N/A	N/A	N/A

- **Class A**: Used for **large networks** (e.g., **multinational corporations**).
  - First octet (8 bits) represents the network, remaining 24 bits for hosts.
- **Class B**: Used for **medium-sized networks** (e.g., **universities, ISPs**).
  - First two octets (16 bits) represent the network, remaining 16 bits for hosts.
- **Class C**: Used for **small networks** (e.g., **small businesses**).
  - First three octets (24 bits) represent the network, last 8 bits for hosts.
- **Class D**: Reserved for multicast applications (not for regular networking).
  - Does not have a subnet mask.
- **Class E**: Reserved for experimental and future use.
  - Not used for public or private networking.
- **Binary Representation**: Each "**1**" in a subnet mask represents the network portion, while each "**0**" represents the host portion.
  - Example for Class C (255.255.255.0): 11111111.11111111.11111111.**00000000**

How the Internet Works

Example: Class C – Number of Hosts

- In a **/24 prefix network** (**Class C**), there are **256 IPv4 addresses** in total.
- The **first** and **last** IPv4 addresses typically serve specific purposes:
  - **First IP (Network Address)**: this is the network identifier and is **not assignable** to a host.
  - **Example:** If the subnet is 192.168.1.0/24, then the **network address** is: **192.168.1.0**

  - **Last IP (Broadcast Address)**: this is used to send packets to all hosts on the subnet and is **not assignable** to a host.
    Example: If the subnet is 192.168.1.0/24, then the **broadcast address** is: **192.168.1.255**

  - **Usable IP Range**:
  - The usable host IPv4 addresses fall between these two: **192.168.1.1 to 192.168.1.254**., **which makes 254 IPv4 addresses in total for the hosts**!

How the Internet Works

Example: Class C – Number of Hosts

How many hosts are in 255.255.252.0 subnet mask?

How the Internet Works

Example: Class C – Number of Hosts

How many hosts are in 255.255.252.0 subnet mask?

- The subnet mask 255.255.252.0 in binary is:
- 11111111.11111111.11111100.00000000
- To determine the number of hosts, we first calculate the number of host bits, which is the number of 0 bits in the subnet mask. In this case, there are 22 1s (network bits) and 10 0s (host bits).
- To calculate the number of hosts, use the formula:
  - $2^{\text{number of host bits}} - 2$
- We subtract 2 to account for the network address and the broadcast address. So:
  - $2^{10} - 2 = 1024 - 2 = 1022$
- Therefore, the number of hosts for the subnet mask 255.255.252.0 is 1022.

How the Internet Works

Example: Class C – Number of Hosts

- The subnet mask 255.255.252.0 in binary is:
- 11111111.11111111.11111100.00000000
- To determine the number of hosts, we first calculate the number of host bits, which is the number of 0 bits in the subnet mask. In this case, there are 22 1s (network bits) and 10 0s (host bits).
- To calculate the number of hosts, use the formula:
  - $2^{\text{number of host bits}} - 2$
- We subtract 2 to account for the network address and the broadcast address. So:
  - 2 10 − 2 = 1024 − 2 = 1022 2 10 −2=1024−2=1022
- Therefore, the number of hosts for the subnet mask 255.255.252.0 is 1022

How the Internet Works

DHCP

DHCP (Dynamic Host Configuration Protocol)

How it Works:

1. **Client Request:** Device sends a DHCP Discover message to find a DHCP server.
2. **Server Offer:** DHCP server responds with a DHCP Offer, including an available IP address.
3. **Client Acceptance:** Device replies with a DHCP Request to accept the offered IP.
4. **Lease Confirmation:** Server sends a DHCP Acknowledgment (ACK), assigning the IP for a set period.

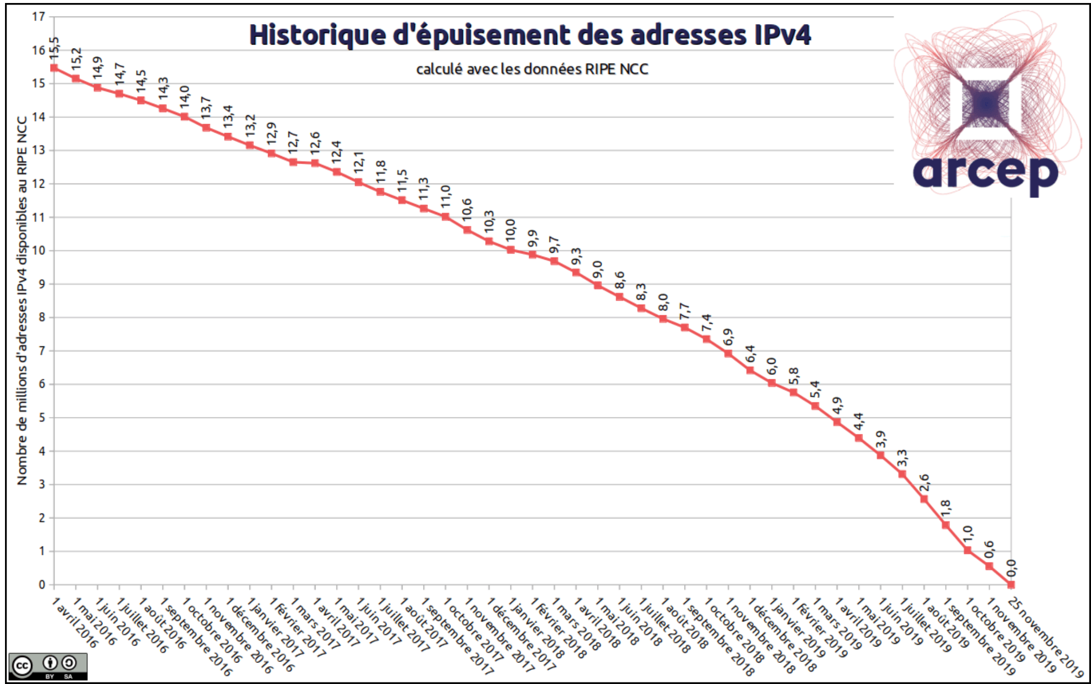
Key Benefits:

- Automatic IP assignment
- Reduces network admin work
- Prevents IP conflicts

d) The Addresses (3/5)

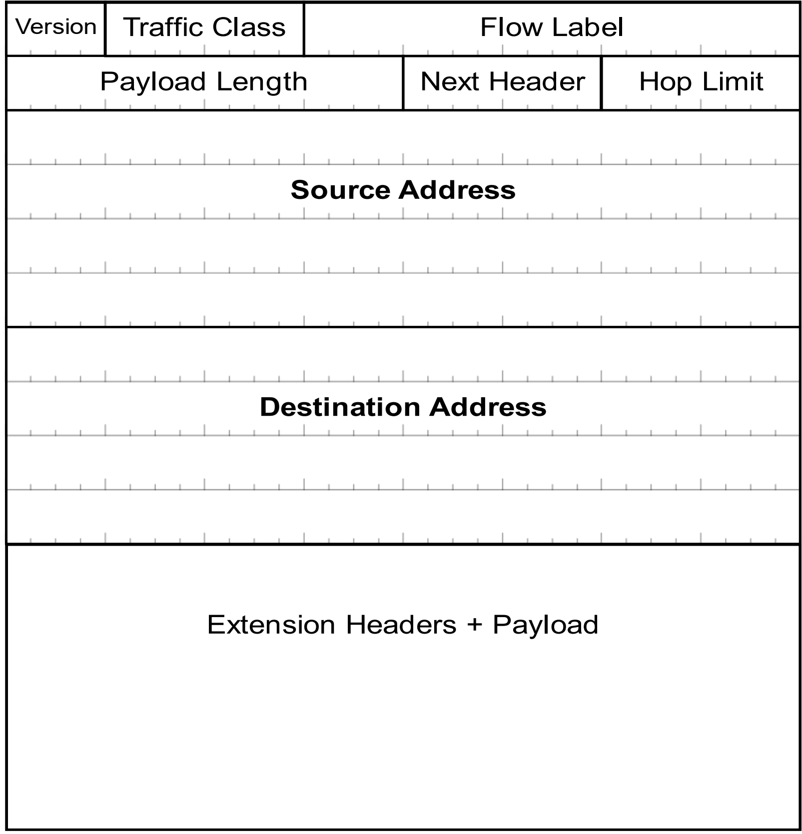
IPv4

- Special address **Loopback**, localhost only scope:
  - 127.0.0.1
  - **Testing Network Stack**: ping the loopback address to check if IPv6 is working on the machine.
  - Inter-Process Communication (**IPC**), allowing programs on the same machine to communicate.
- **Private** IPv4 addresses:
  - Class A: 10.0.0.0 to 10.255.255.255
  - Class B: 172.16.0.0 to 172.16.255.255
  - Class C: 192.168.0.0 to 192.168.255.255
  - Not routed on the Internet.
  - **except** Network Address Translation (NAT).
- Shortage in February 2011 (IANA & RIR):
  - Recovery of allocated but unused blocks, Resale.
  - ... **IPv6** (see next slide)!



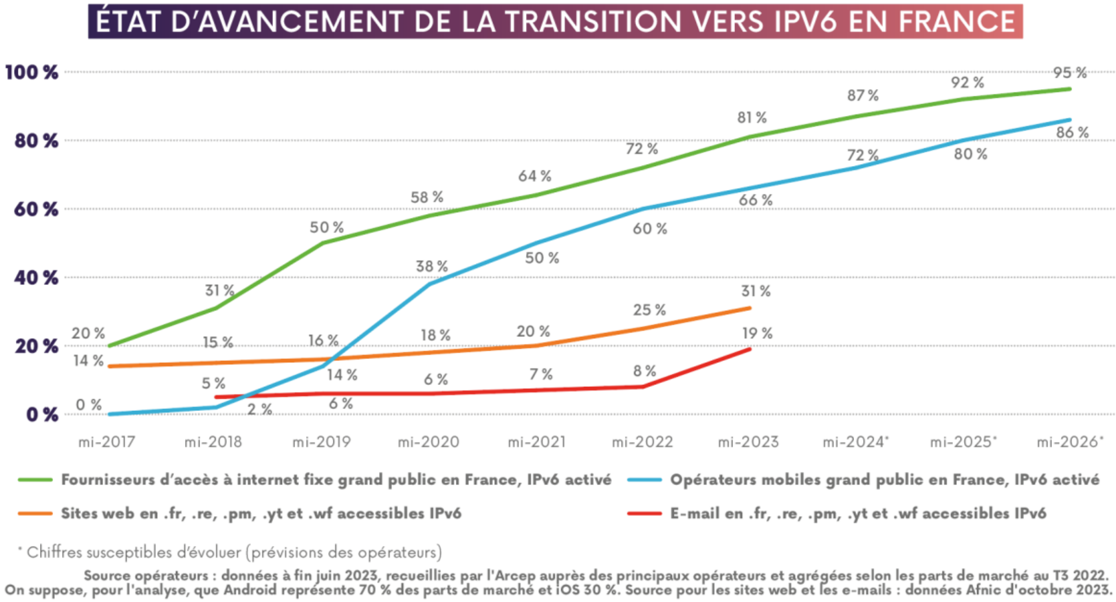
How the Internet Works

d) The Addresses (4/5)



IPv6 (since 1995)

- Every interface has at least 2 addresses
- Size: 128 bits grouped in 8 blocks.
  - Hexadecimal notation.
  - 2001:DB8:0:0:8:800:200C:417A
  - 2001:DB8::8:800:200C:417A
  - FF02::9: calculate the ‘0’ bits replaced by “::”!
- CIDR prefix:
  - 2001:0DB8:0:CD30::/60
- Configuration
  - by hand: uh... no thanks!
  - by a DHCPv6 server.



How the Internet Works

d) The Addresses (5/5)

IPv6 Type of Addresses

- **Unicast**
  - A **unicast address identifies a single interface** of a device.
- **Anycast**
  - An **anycast address is assigned to a group of interfaces**, usually belonging to different devices.
  - **A packet** sent to an anycast address **is routed to the "*nearest*" interface having that address**, according to the router's routing table.
- **Multicast** (FF00::/8)
  - A **multicast address is used by multiple devices.**
  - A packet that is transmitted to a multicast address is delivered to all interfaces that have registered the corresponding multicast group.

There are no broadcast addresses in IPv6, the multicast addresses are employed instead.

HOW THE INTERNET WORKS

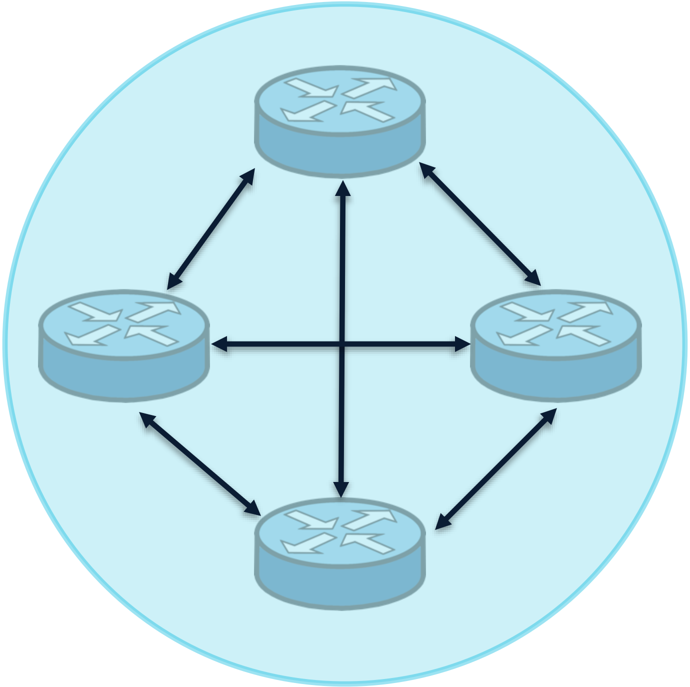
d) The Addresses (5/5)

Scopes (portée) of Unicast Addresses

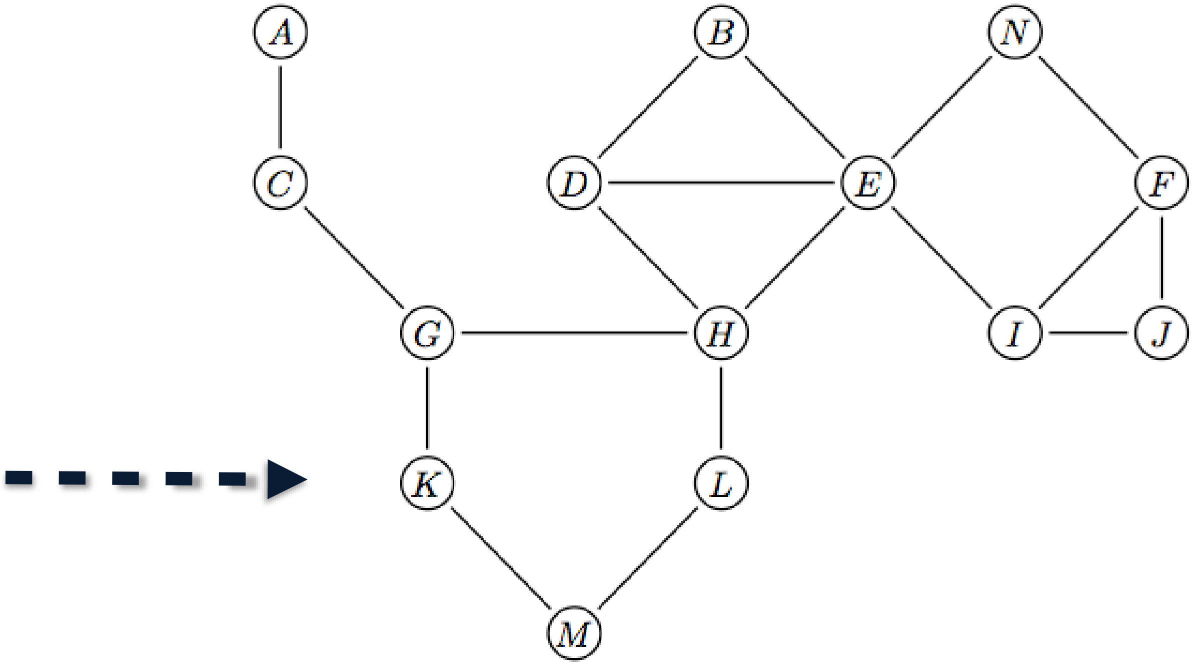
- **A Scope specifies in which part of the network an address is valid.**
- **Link-local Scope**: addresses that are reachable only on the same local link (same network segment or subnet), not forwarded by routers:
  - **Loopback Address**
    - ::1/128
    - **It is the equivalent of IPv4's 127.0.0.1, used for testing and local communication.**
  - **Link-Local Address (LLA)**:
    - **Used for communication between nodes on the same local link** (e.g., devices connected to the same switch).
    - Every IPv6-enabled interface automatically gets a link-local address, i.e., based on fe80::/10 prefix.
    - Used for essential functions like router discovery, neighbour discovery, and automatic address configuration.
- Global scope: addresses that are (or could be) globally routable.
  - **Unique Local Address (ULA)**:
    - **Used for private/internal networks**, similar to IPv4 private addresses.
    - Manually assigned or generated randomly (e.g., using the fd00::/8 prefix with a random 40-bit global ID).
  - **Global Unicast Address (GUA)**:
    - **Used for devices that need to communicate over the public internet**, e.g., 2601:db8:abcd:1234::1.
    - Configured either manually or via DHCPv6 server.

How the Internet Works

e) Routing



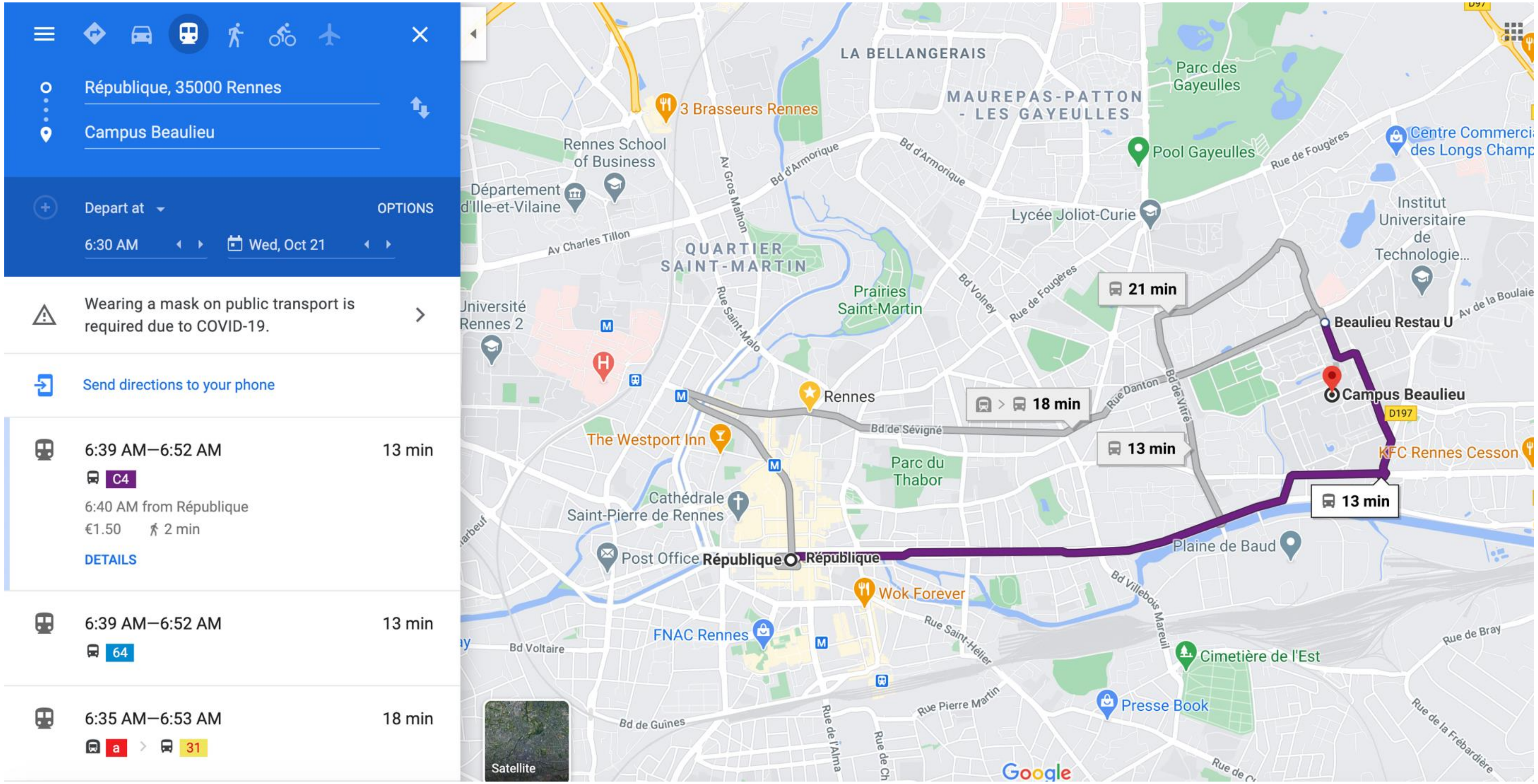
All routers are direct communication



Extended network area

How the Internet Works

e) Routing

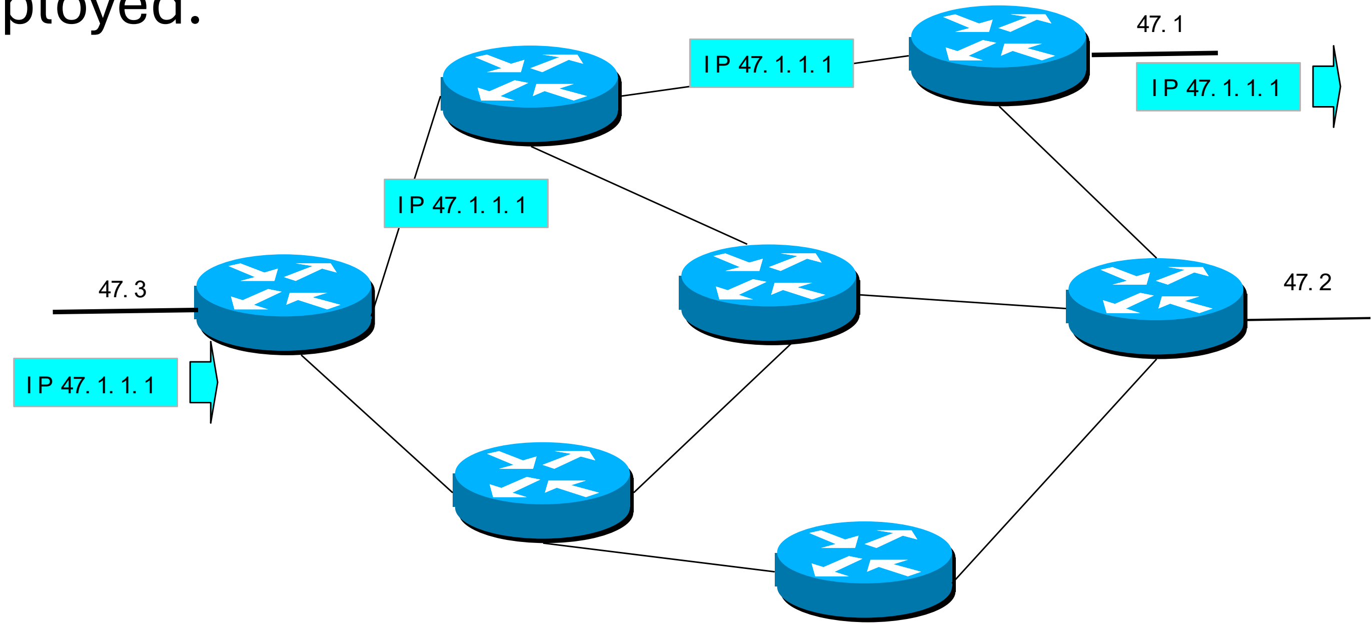


How the Internet Works

e) Routing

IP Routing – Layer 3. Network

- Switching from one router to another.
- Hop-by-hop routing principle
- Layer 3 based machines (routers, hosts) may route the packets.
- Public/Global IP addresses are employed.



How the Internet Works

e) Routing Protocol: the Two Principles (1/2)

1. Routing

Create/calculate and maintain the routes:

- Configuration by administrator.
- Goals:
  - No loops, fast convergence, energy consumption, Minimum signalling, etc …
- Routers must have either a **local** or **global** view of the network topology (e.g., link quality, neighbours).
- Build **Routing Tables**
  - Destination - gateway - weight
- Between ASes: i.e., BGP protocol.
- Inside an AS: e.g., OSPF, IS-IS protocols.

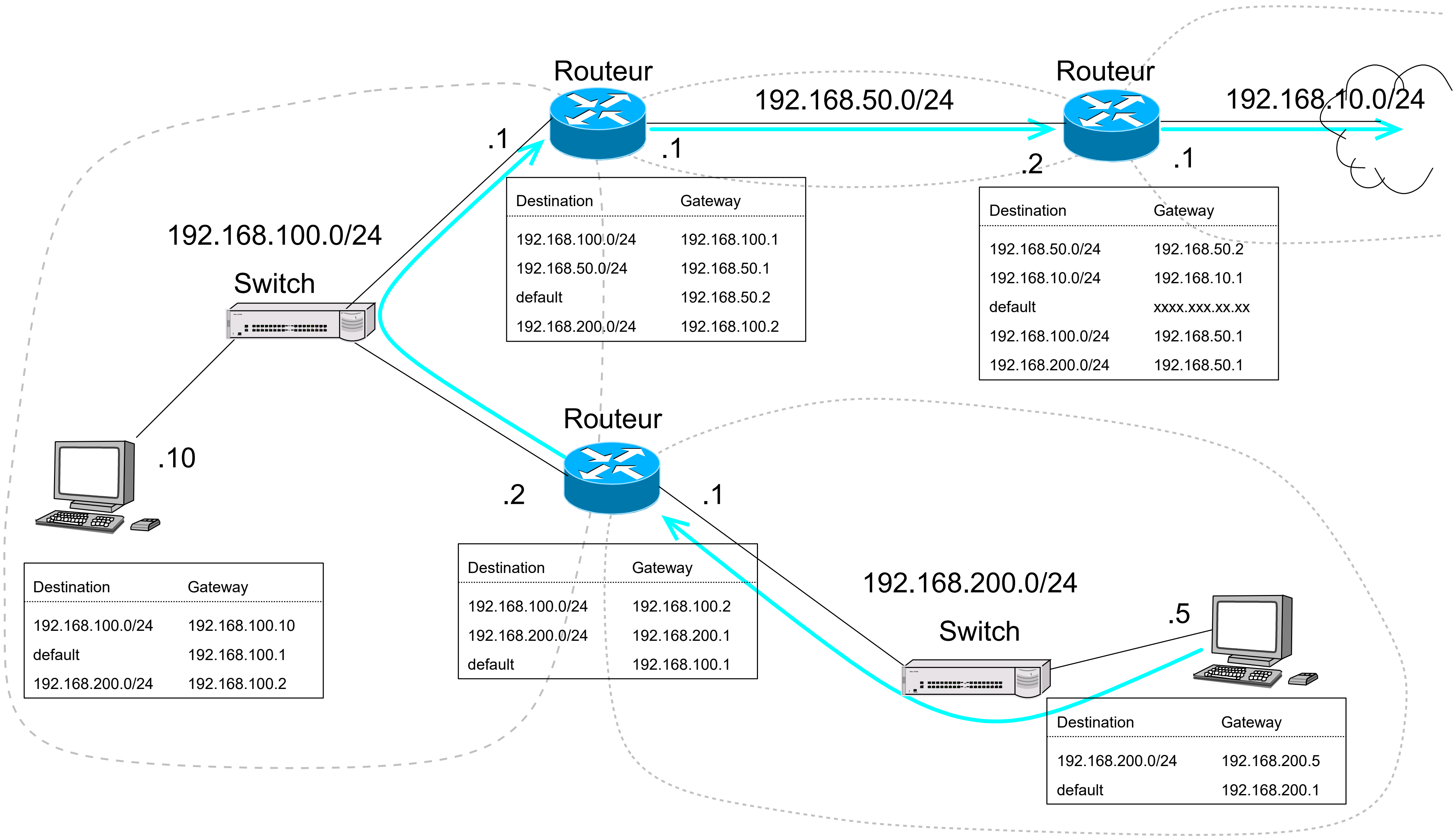
How the Internet Works

e) Routing Protocol: the Two Principles (2/2)

2. Forwarding

Forward the received packets:

- to the correct next router.
- based on the packet's IPv4/6 destination address.
- by applying routing tables:
  - if destination matches, forwards to the specified gateway
  - chooses the longest (precise) prefix



Destination	Gateway
192.168.100.0/24	192.168.100.1
192.168.50.0/24	192.168.50.1
default	192.168.50.2
192.168.200.0/24	192.168.100.2

Destination	Gateway
192.168.50.0/24	192.168.50.2
192.168.10.0/24	192.168.10.1
default	xxxx.xxx.xx.xx
192.168.100.0/24	192.168.50.1
192.168.200.0/24	192.168.50.1

Destination	Gateway
192.168.100.0/24	192.168.100.10
default	192.168.100.1
192.168.200.0/24	192.168.100.2

Destination	Gateway
192.168.100.0/24	192.168.100.2
192.168.200.0/24	192.168.200.1
default	192.168.100.1

Destination	Gateway
192.168.200.0/24	192.168.200.5
default	192.168.200.1

How the Internet Works

f) Transport of Information (1/3)

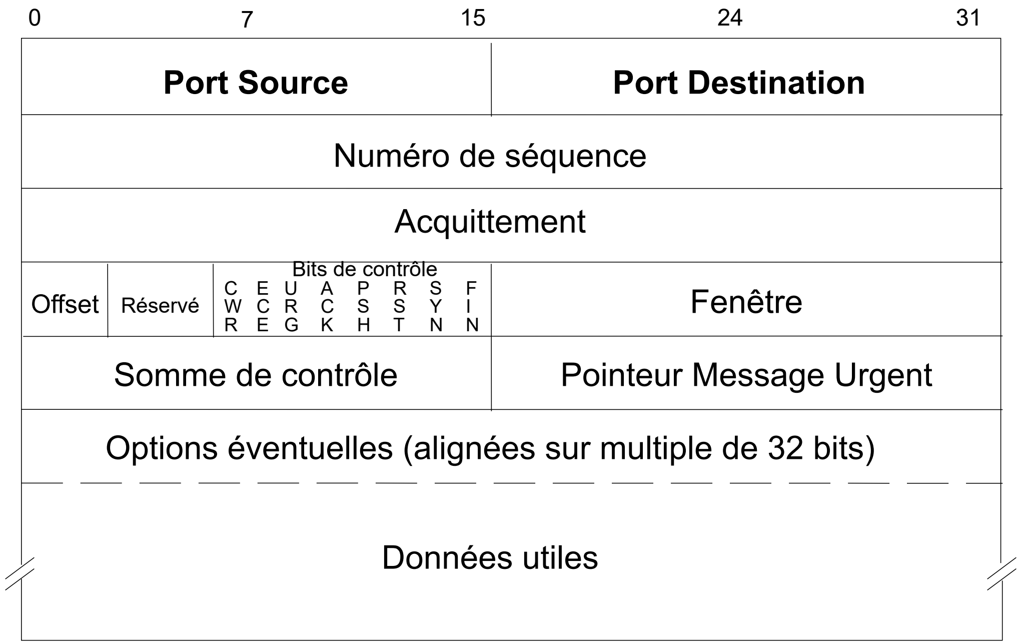
Layer 4. Transport

- **Objective:**
  - Ensures **reliable or fast** data delivery between devices.
- Uses **Port numbers** to identify specific applications/services on a device.
  - **Examples**: HTTP → Port 80, Email (SMTP) → Port 25, DNS (Domain Name System) → Port 53.
  - **Analogies**: 1. apartment numbers in a building, 2. a specific department inside the company.
- **Main Protocols:**
  - **TCP (Transmission Control Protocol)**: Reliable, connection-oriented, e.g., Web (HTTP), Email (SMTP).
  - **UDP (User Datagram Protocol)**: Fast, connectionless (e.g., VoIP calls, Video Streaming).
- **Analogy:**
  - **TCP = 1. Certified Mail** (*avec accusé de réception*) 📩 (Ensures delivery, tracks each step, requests re-delivery if lost), **2. Phone Call** ☎️ (You establish a connection, talk in order, and confirm messages).
  - **UDP = Regular Postcard** 📪 (Sent quickly but no guarantee of arrival).

How the Internet Works

f) Transport of Information (2/3)

Transmission Control Protocol (TCP)



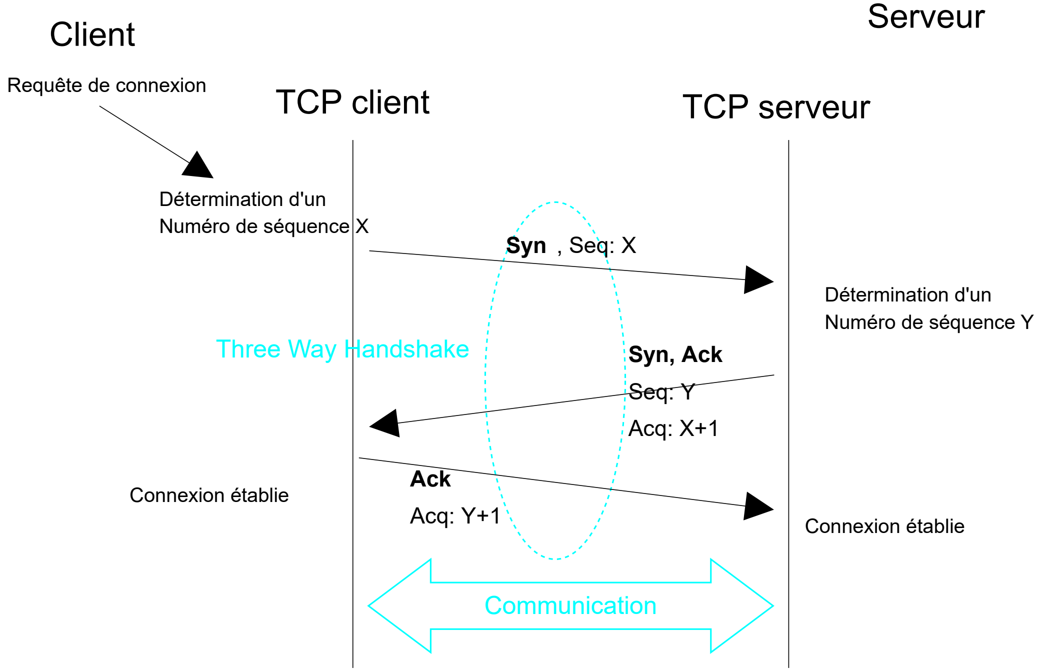
- A **connection-oriented** transport protocol that ensures reliable data delivery.
- **Establishes a virtual connection** between sender and receiver before sending data.
- **Key Functions:**
  - **Segmentation & Reassembly**: Breaks large messages into smaller packets and reassembles them.
  - **Flow Control** 🚦 : Prevents a fast sender from **saturate the (slow) receiver** by adjusting data transmission speed.
  - **Congestion Control** 🚧 : Prevents too much traffic **in the network**, which can cause delays & packet loss.
  - **Error Detection & Recovery**: Detects errors and retransmits lost packets (how?).

How the Internet Works

f) Transport of Information (2/3)

Transmission Control Protocol (TCP)

- **3-Way Handshake** (Connection Setup):
  - SYN (Synchronize): The **client sends** a SYN packet to the server to **request a connection**.
  - SYN-ACK (Synchronize-Acknowledge): The **server responds** with a SYN-ACK **to acknowledge** the **request**.
  - ACK (Acknowledge) – The **client sends** an ACK packet **to confirm the connection**, and communication begins.



f) Transport of Information (3/3)

User Datagram Protocol (UDP)

- A **connectionless** transport protocol that sends data without establishing a connection.
- Offers **faster data transmission** compared to TCP but does not guarantee reliability.
- Best **suited for real-time applications** (suitable for **multicast** traffic) where **speed is more important than error-free delivery**.
- **UDP comes with no control of any kind, so all the features of TCP (see below) are not valid:**
  - **Segmentation & Reassembly**: Breaks large messages into smaller packets and reassembles them.
  - **Flow Control** 🚦 : Prevents a fast sender from **saturate the (slow) receiver** by adjusting data transmission speed.
  - **Congestion Control** 🚧 : Prevents too much traffic **in the network**, which can cause delays & packet loss.
  - **Error Detection & Recovery**: Detects errors and retransmits lost packets.

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

0	Source Port	Destination Port
4	Length	Checksum

OUTLINE

- What is the Internet?
- Internet Architecture
- How the Internet Works
- Making Communicating Applications
    a) Some Examples
    b) The DNS Service

HOW THE INTERNET WORKS

a) Some Examples (1/2)

Web – HTTP 1.x

- Transport: **TCP**, port **80** (HTTPS 443).
- Presentation: text Client-server.
- Mode: **Request-Response**.

```
GET /ueinfo-fise1a/ HTTP/1.1
Host: hub.imt-atlantique.fr
User-Agent: Mozilla/5.0 Gecko/20100101 Firefox/128.0
Accept: text/html,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate, br, zstd
DNT: 1
Connection: keep-alive
Cookie: _wayfIMT_saml_sp=S0lULUVUMDE%3D
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Priority: u=0, i
```

```
HTTP/1.1 200 OK
Date: Fri, 01 Nov 2024 16:36:44 GMT
Server: Apache/2.4.61 (Debian)
Last-Modified: Fri, 25 Oct 2024 11:43:03 GMT
ETag: "b1ab-6254b9f245fc0-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 6342
Content-Type: text/html
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    ../..
    </head>
  <body lang="en" dir="ltr" itemscope itemtype="http://schema.org/Article">

  ../..
  </body>
</html>
```

a) Some Examples (2/2)

Sending Email – SMTP

- Transport: **TCP**, port **25** (SMTPS 587).
- Presentation: text.
- Client-server mode, dialog.

← 220 serveur.domain.tld ESMTP Sendmail
→ EHLO client.domain.tld
← 250-serveur.domain.tld Hello client.domain.tld, pleased to meet you
← 250-ENHANCEDSTATUSCODES
← 250-PIPELINING
← 250-EXPN
← 250-VERB
← 250-8BITMIME
← 250-SIZE
← 250-DSN
← 250-ETRN
← 250-AUTH DIGEST-MD5 CRAM-MD5
← 250-DELIVERBY
← 250 HELP
→ MAIL FROM: user1@domain.tld
← 250 2.1.0 user1@domain.tld... Sender ok
→ RCPT TO: user2@domain.tld
← 250 2.1.5 user2@domain.tld... Recipient ok
→ DATA
← 354 Enter mail, end with "." on a line by itself.
→ From: user1@un-endroit.fr
→ To: user2@ailleur.eu
→ Subject: Sujet du message
→
→ Corps du message...
→ .
← 250 2.0.0 k1QGh9xB020668 Message accepted for delivery
→ QUIT

How the Internet Works

b) The DNS Service

Domain Name System (DNS)

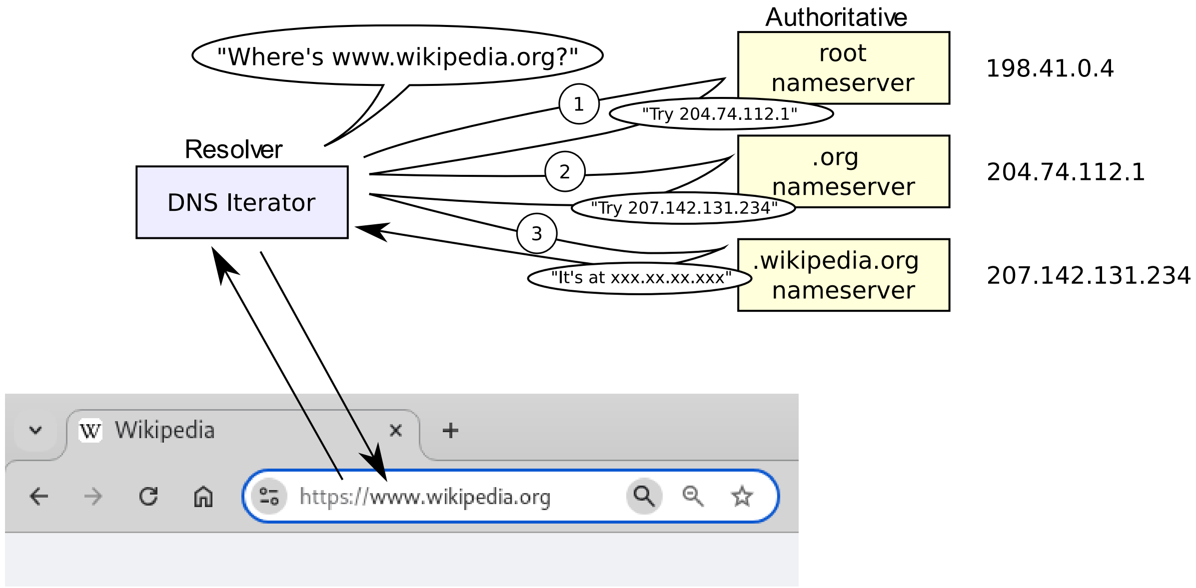
- **What is DNS?**
  - DNS is the system that **translates human-friendly domain names** (e.g., www.example.com) **into IP addresses** (e.g., 192.0.2.1) that computers use to identify each other on the network.
  - Indispensable fundamental network config: IP Address + Gateway (Router) + DNS server.
  - **Analogy**: an **address book**, helping you find locations by name, or a **phonebook/contacts** (Apple app).

HOW THE INTERNET WORKS

b) The DNS Service

Domain Name System (DNS)

- **How does it work?**
  - **User Request**: When a URL is typed, the device asks a DNS server to resolve the domain name.
  - **DNS Lookup**: The server checks its records **or** queries other DNS servers to find the corresponding IP address.
  - **Response**: The IP address is returned, and the browser connects to the website.



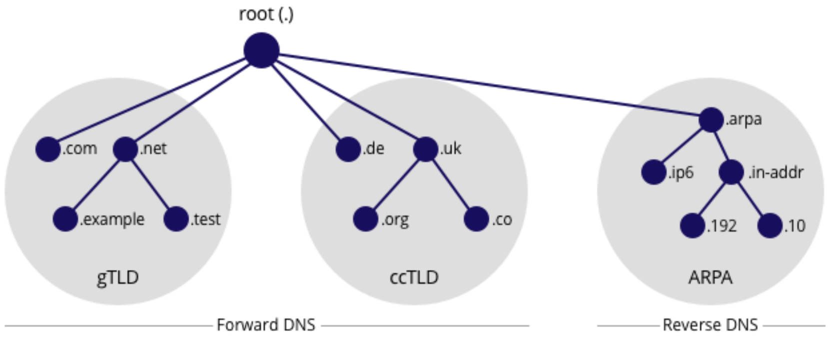
How the Internet Works

b) The DNS Service

DNS Structure & Hierarchy Content

The hierarchical structure of DNS consists of multiple levels:

1. **Root Level** (.): The highest level in the hierarchy (**ICANN / IANA**). There are **13 root servers** globally that store records for Top-Level Domains (TLDs).
2. **Top-Level Domains** (**TLDs**):
   1. Generic TLDs (gTLDs): .com, .org, .net, .edu, etc.
   2. Country Code TLDs (ccTLDs): .uk, .de, **.fr** (**Afnic**), .jp, etc.
3. **Second-Level Domains** (**SLDs**): The main domain name, e.g., example in example.com.
4. **Subdomains**: Additional prefixes before the main domain, e.g., blog.example.com.



How the Internet Works

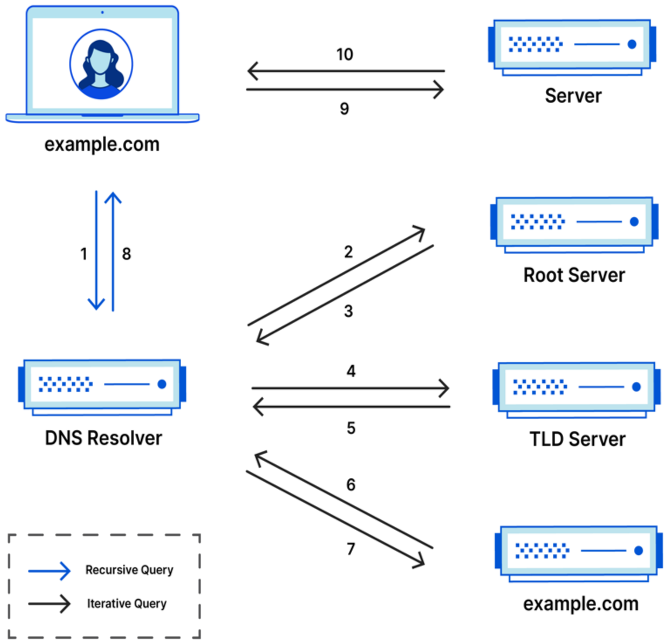
b) The DNS Service

Example: Resolving www.example.com

When a **client** enters www.example.com, the browser initiates a DNS lookup. The query passes through multiple DNS servers to get the correct IP:

1. **Resolver** (ISP DNS or public resolver like Google 8.8.8.8) contacts Root DNS.
2. **Root DNS** directs the resolver to the .com gTLD DNS server.
3. **gTLD DNS Server** refers the query to the Authoritative DNS for example.com.
4. **Authoritative Name Server** provides the IP address (93.184.216.34).
5. The browser establishes a connection to that IP, loading the website.

Complete DNS Lookup and Webpage Query



*FYI: **browsers and operating systems use DNS caching to speed up future lookups.***

Principles of
Networks and the Internet

Bloc Réseaux – Session 1