Corso reti per neo assunti (PNRR)

Netgroup

Introduction

This course is thought as an introduction on networking themes.

- It will focus on the general concepts of networking
  - Main protocols
  - Network Devices
  - Network Services
  - Common configurations
  - Best practices

ISO/OSI Model

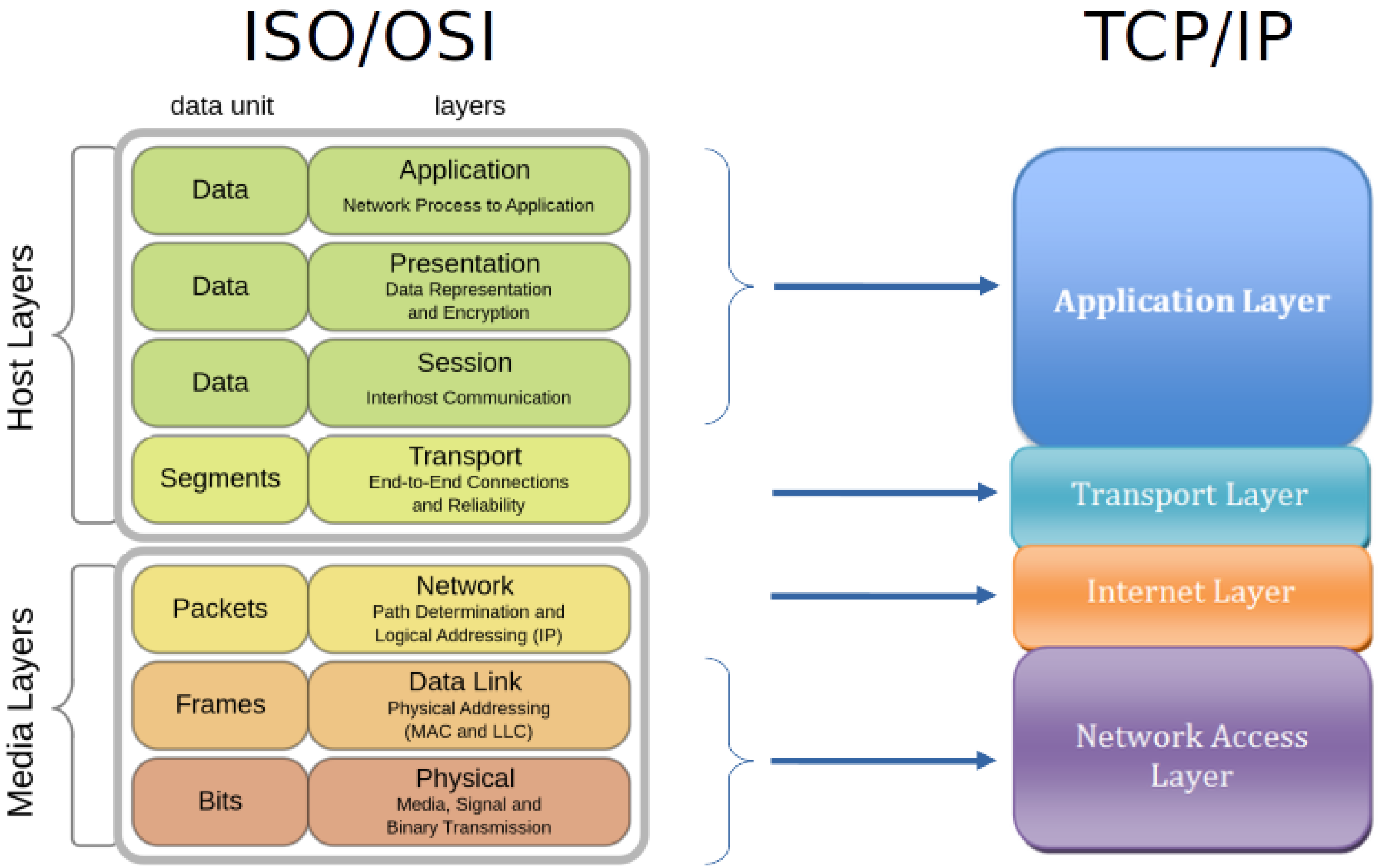
ISO/OSI Model

- The OSI model (an acronym for **Open Systems Interconnection**, also known as the ISO/OSI model) is a standard established in 1984 by the International Organization for Standardization (ISO) as a conceptual framework that **describes how network communication should occur.** It is a layered model that defines a set of protocols and standards for communication between devices on a network.

- The ISO/OSI model consists of **seven layers**, each of which is responsible for a specific aspect of network communication. The layers are as follows.

- The **ISO/OSI model is a conceptual framework**, meaning that it does not specify the exact protocols or technologies that should be used at each layer. Instead, it provides a guide for the development of networking standards and protocols. Different networking technologies, such as Ethernet or TCP/IP, implement the ISO/OSI model in different ways, but the general principles of the model remain the same.

ISO/OSI Model vs TCP/IP Model



Physical layer (1)

- The Physical layer is the lowest layer in the ISO/OSI model and is **responsible for transmitting raw data bits over a physical medium,** such as copper or fiber optic cables, the voltage levels used to transmit data, and the frequency of the signal.
- It provides the **physical and electrical interface** between the network device and the **transmission medium**.
- It is responsible **for encoding and decoding data**, as well as converting digital signals to analog signals and vice versa.
- It is also responsible for maintaining the quality of the signal as it is transmitted over the network, by **amplifying or repeating** the signal as needed.

Data link layer (2)

- The Data Link layer is the second layer in the ISO/OSI model and **provides a reliable, error-free transfer of data between adjacent network devices over a physical medium, such as a cable.**
- The Data Link layer provides the following services.
  - **Framing:** The Data Link layer frames the data into packets for transmission over the physical medium. It adds a header and trailer to the data to create a frame and includes information such as the source and destination MAC addresses, frame type, and error checking information.
  - **Flow control:** The Data Link layer controls the flow of data between network devices to prevent data loss or congestion. It uses techniques such as windowing or buffering to ensure that data is transmitted at an appropriate rate.
  - **Error detection and correction:** The Data Link layer detects and corrects errors that may occur during transmission. It uses techniques such as cyclic redundancy checking (CRC) to detect errors, and retransmission of lost packets to correct errors.
  - **Access control:** The Data Link layer controls access to the physical medium, preventing multiple devices from transmitting at the same time and causing collisions. It uses techniques such as Carrier Sense Multiple Access with Collision Detection (CSMA/CD) or Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to coordinate access to the medium.
  - **Addressing:** The Data Link layer uses MAC addresses to identify devices on the network. MAC addresses are unique identifiers assigned to each network interface and are used by the Data Link layer to transmit data between devices.
- **The Data Link layer and Physical layer of the ISO/OSI model are roughly equivalent to the Network Access layer of the TCP/IP model.**

Network layer (3)

- The Network layer is the third layer in the ISO/OSI model **and provides logical addressing and routing of data between different networks.** It is responsible for delivering data between end devices, regardless of their physical location in the network.
- The Network layer provides the following services.
  - **Logical addressing:** Network layer uses logical addressing to identify devices on the network. IP addresses are commonly used for logical addressing in the Internet Protocol (IP) suite. Logical addressing provides a hierarchical structure that allows efficient routing of data across large networks.
  - **Routing:** Network layer is responsible for selecting the best path using static or dynamic routing protocols for data to travel between devices on different networks. Typical Routing protocols are: RIP (Routing Information Protocol), OSPF (Open Shortest Path First) and BGP (Border Gateway Protocol).
  - **Fragmentation and reassembly:** Network layer can fragment (and finally reassemble) large packets into smaller ones for transmission over networks with smaller Maximum Transmission Units (**MTUs**).
  - **Quality of Service (QoS):** Network layer can provide QoS by prioritizing certain types of traffic, for example, real-time traffic such as voice or video.
  - **Error detection and handling:** Network layer can detect and handle errors that may occur during transmission. It uses protocols such as Internet Control Message Protocol (ICMP) to detect errors and notify the sender or receiver of them.
- **The ISO/OSI Network layer is roughly equivalent to the Internet layer in the TCP/IP model.**

Transport layer (4)

- The Transport layer is the fourth layer in the ISO/OSI model and is responsible for **providing reliable end-to-end communication between applications running on different devices.** The Transport layer provides the following services.
  - **Connection-oriented communication**: the Transport layer provides a connection-oriented communication mode, in which a connection is established between the sender and receiver before data transmission can begin.
  - **Connectionless communication**: the Transport layer also provides a connectionless communication mode, in which data is transmitted without a connection being established between the sender and receiver. This mode is useful for applications that require low overhead and do not require reliability guarantees.
  - **Flow control:** the Transport layer controls the flow of data between devices to prevent data loss or congestion. It uses techniques such as *windowing* or buffering to ensure that data is transmitted at an appropriate rate.
  - **Error recovery**: the Transport layer provides error recovery mechanisms to ensure that data is transmitted accurately and completely. It uses techniques such as **acknowledgement**, **retransmission, and error correction** to recover from errors.
  - **Multiplexing and demultiplexing:** the Transport layer can support multiple applications running on a single device, and can multiplex or demultiplex data from different applications to ensure that the data is delivered to the correct application.
- The TCP/IP Transport layer is roughly equivalent to the ISO/OSI Transport layer.
- **The Transport layer is implemented in two main protocols: the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). TCP provides connection-oriented communication and reliable delivery of data, while UDP provides connectionless communication and is useful for applications that require low overhead and do not require reliability guarantees.**

Session (5) and Presentation(6) layer

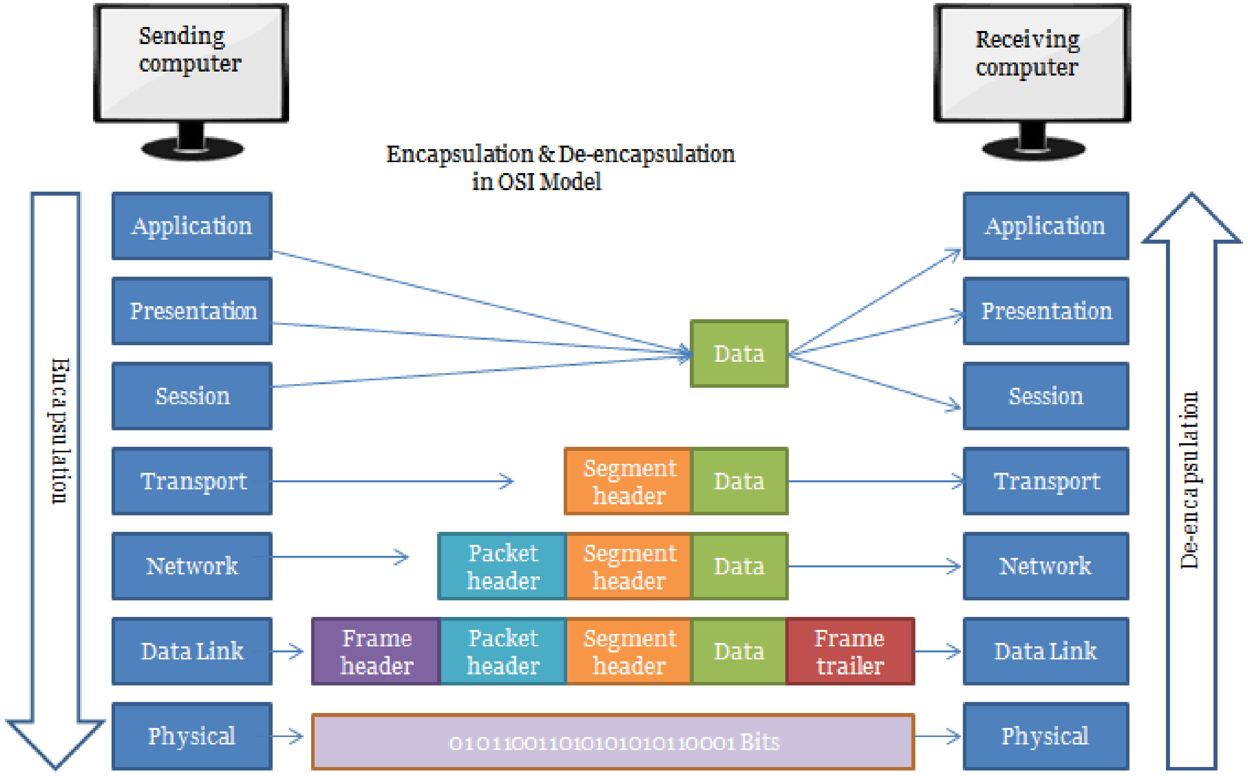
- **The Session layer** provides a mechanism for applications to communicate and exchange data in a structured and organized way and provides the following services: ***Session establishment, Session management and Session termination.***
- The **Presentation layer** is the sixth layer in the ISO/OSI model, and it is responsible for managing the way data is presented to the Application layer and performs the following functions: ***Translation, Encryption, compression and formatting.***
- **The ISO/OSI Session and Presentation layers are included in the TCP/IP Application layer**

Application layer (7)

- The **Application layer** is the seventh and topmost layer in the ISO/OSI model, and it is responsible for **providing a user interface to access network services**. The Application layer interacts directly with the end user or application and provides high-level services such as file transfer, email, remote login, and web browsing.
- It also provides protocols for applications to exchange data with each other, such as the Simple Mail Transfer Protocol (SMTP) for email, or the Hypertext Transfer Protocol (HTTP) for web browsing.

Encapsulation

- Packet encapsulation is the process of adding headers and trailers to a packet as it travels through the layers of the network protocol stack. Each layer in the protocol stack adds its own header and trailer to the data, forming a new encapsulated packet that is sent to the next layer in the stack.
- When the packet is received by the receiving device, the headers and trailers are removed in the reverse order. The receiving device uses the information in the headers to route the packet to its intended destination, and the data payload is passed up from physical to application layer.
- With this workflow, different network devices communicate each other using its own level headers and trailers



ETHERNET

Ethernet History

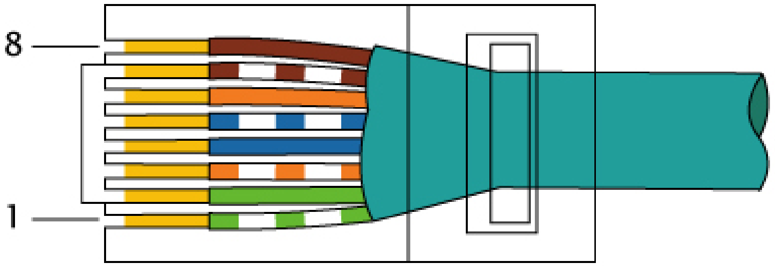
- Ethernet starts as a collaboration between DEC/Intel/Xerox and was standardized in 1978 and few years later the IEEE 802.3 standard was published.
- The protocol allow a data communication on a shared medium and constitutes an implementation of the CSMA/CD protocol (Data Link Layer).
- First version of Ethernet protocol specified a speed of 10/100 Mbps, followed by 1/10/25/40/50/100/400 Gbps

Ethernet and ISO/OSI model

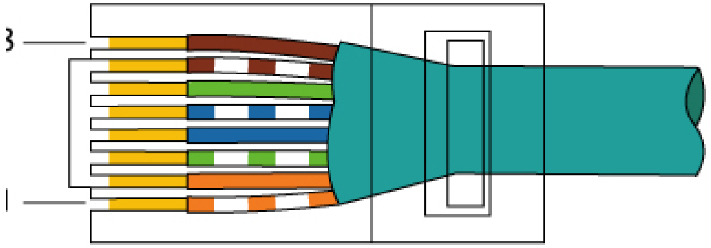
- IEEE 802.3 defines **both the physical layer and the datalink layer** (**LLC** and **MAC**)
- The naming convention of the physical layer specify the **transmission speed**, the **frequency range** of the band (base, broad, pass) and **the transmission medium** (T = twisted pair, -T1 = single-pair twisted pair, S = 850 nm short wavelength on fiber, etc.).
  - Ex.: 10GbaseT = 10 Gbps in baseband on UTP copper cable
- The datalink layer has two sublayers:
  - **MAC (Media Access Control)**: different for each type of physical layer. Determines the access modes to the medium
  - **LLC (Logical Link Control)**: provides a common interface to the higher layers

Physical layer example: BaseT

- Wiring that uses copper pairs
- Each station is connected via a UTP cable (cat. 5e or higher for 1Gbs) to a multi-port device (HUB or Switch)
- Use of RJ45 connectors
- Data transmission for 10/100Mbps on two pairs (pin 1-2 and 3-6) and for higher speed all pairs are used.



T568A

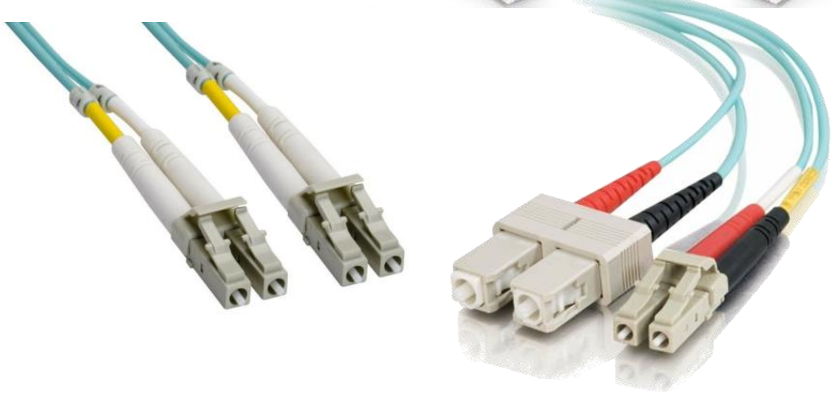


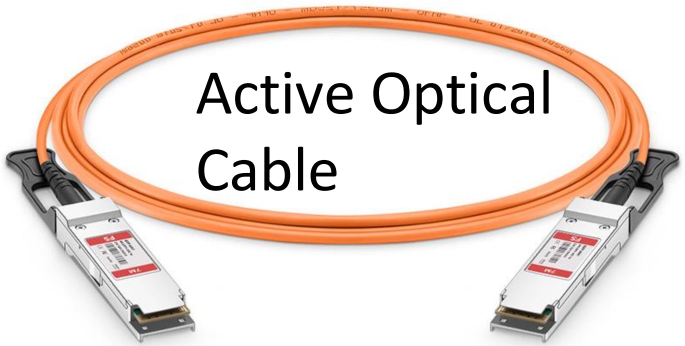
T568B

Some cables

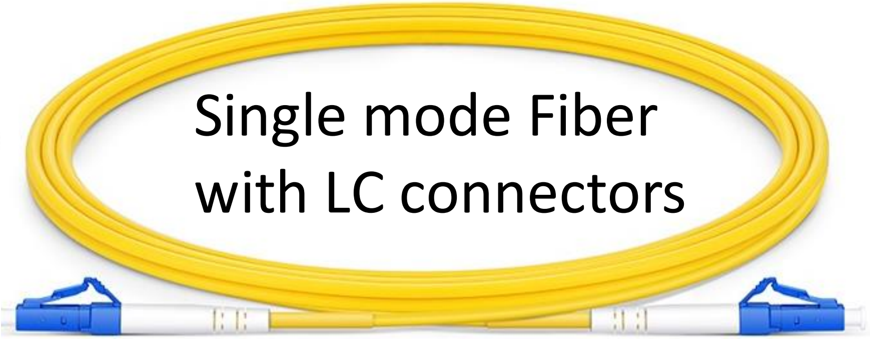
COPPER

Direct
Attached
Copper (DAC)





Active Optical
Cable



Single mode Fiber
with LC connectors

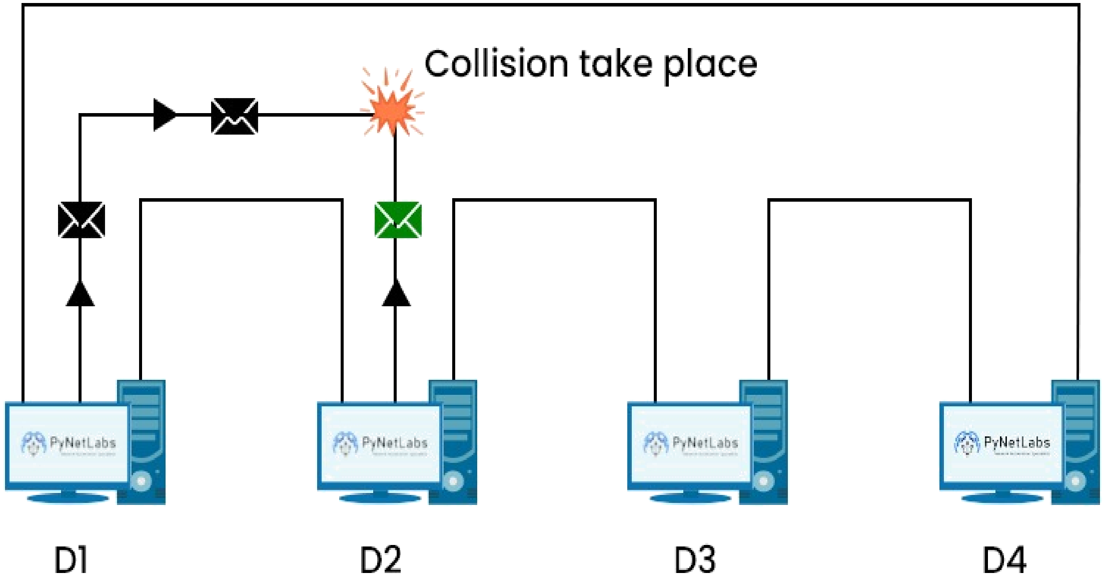
Multimode Fiber with
LC SC connectors

FIBER

Multimode Fiber with
MPO connectors

The Media Access Control

- The MAC manages the shared access to the transmission medium and the possible presence of collisions (simultaneous accesses) using the **CSMA/CD** technique
  - The station that has to transmit, waits and listens on the medium
  - if it finds the medium free, it transmits immediately
  - if it finds the medium busy, it stays listening, and as soon as it frees up it transmits

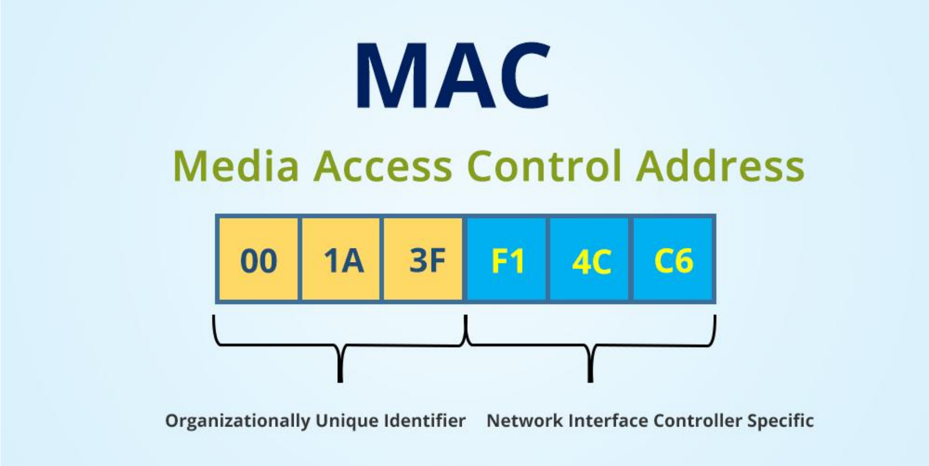


During transmission, the station stays listening, in order to detect a possible collision; in case of collision the station transmits a short invalid sequence, then terminates the transmission of the frame immediately

The contention phase starts, regulated by a mechanism called binary exponential backoff

The MAC address

- Each Ethernet connected devices must be equipped with a network card NIC (Network Interface Card) which is identified by a worldwide unique MAC Address
- The address (**MAC address**) consists of 6 bytes.
  - The first 3 octets are called **OUI** (Organizationally Unique Identifier) and refer to the manufacturer, the others are an identifier of the single card

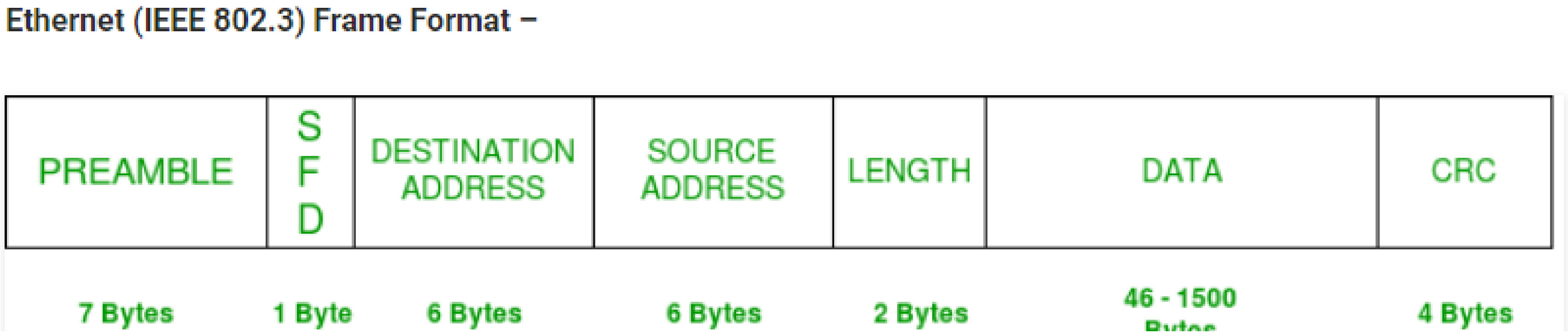

MAC
Media Access Control Address
00
1A
3F
F1
4C
C6
Organizationally Unique Identifier
Network Interface Controller Specific

The transmission

- The Protocol Data Unit (PDU) of the datalink layer is called **frame** and allows to separate the data flow into smaller and more easily manageable units
- Depending on the network topology, a frame sent by a NIC (Network Interface Card) should directly reach another device or be retransmitted by intermediate devices :
  - **HUB**: device with multiple interfaces that acts as a repeater on each network connection
  - **SWITCH**: device that maintains a mapping between destination addresses and ethernet ports
- Normally a NIC accepts only frames sent to its MAC address. It is possible to force it to allow every frame transmitted on the medium, putting it in **promiscuous** mode.

The frame

- The frame begins with a "preamble" of 8 bytes with sequence 10101010
- The last byte of the preamble is a "start of frame" byte with value of 10101011
- Two fields contain to the **destination address** and **source address** of the frame
- The "length" field contains the length of the data field expressed in octets
    - The default size of the data field (MTU) is 1500 bytes. Sometimes applications performs better with larger MTUs, and 9000 bytes is the default for **jumbo frames.**

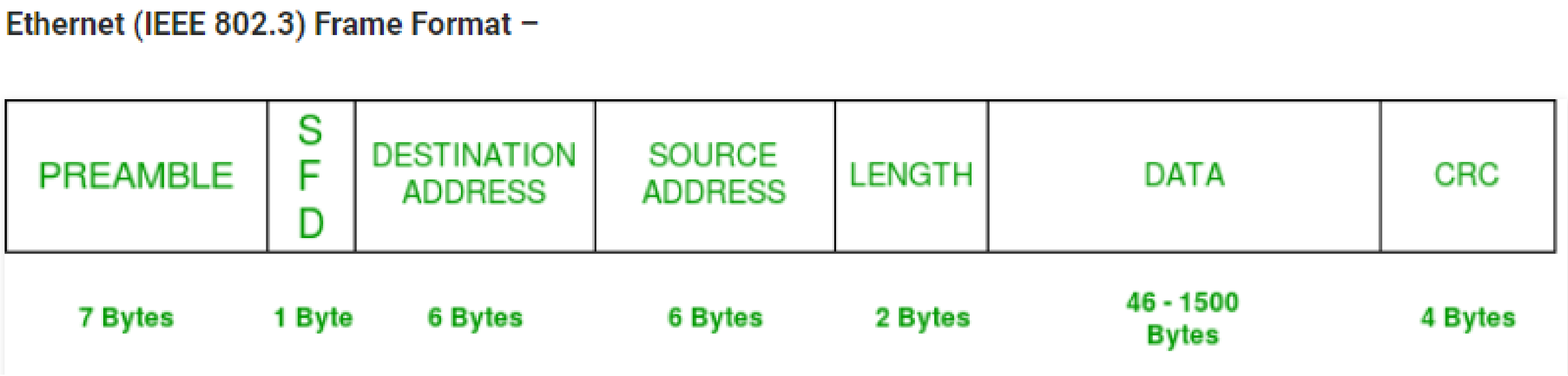


Ethernet (IEEE 802.3) Frame Format –

PREAMBLE	SFD	DESTINATION ADDRESS	SOURCE ADDRESS	LENGTH	DATA	CRC
7 Bytes	1 Byte	6 Bytes	6 Bytes	2 Bytes	46 - 1500 Bytes	4 Bytes

The Cyclic Redundancy Check (CRC)

- The last field is dedicated to the checksum, realized by a 32-bit CRC code
- The field is used only to identify any data corruption and discard bad frames
- **Protocols at higher level are in charge to retransmit any missing packets**



Ethernet (IEEE 802.3) Frame Format –

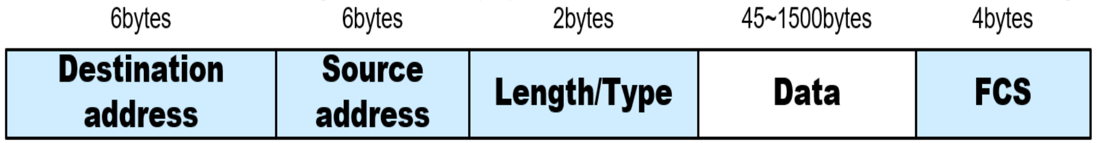
PREAMBLE	SFD	DESTINATION ADDRESS	SOURCE ADDRESS	LENGTH	DATA	CRC
7 Bytes	1 Byte	6 Bytes	6 Bytes	2 Bytes	46 - 1500 Bytes	4 Bytes

VLAN 802.1q

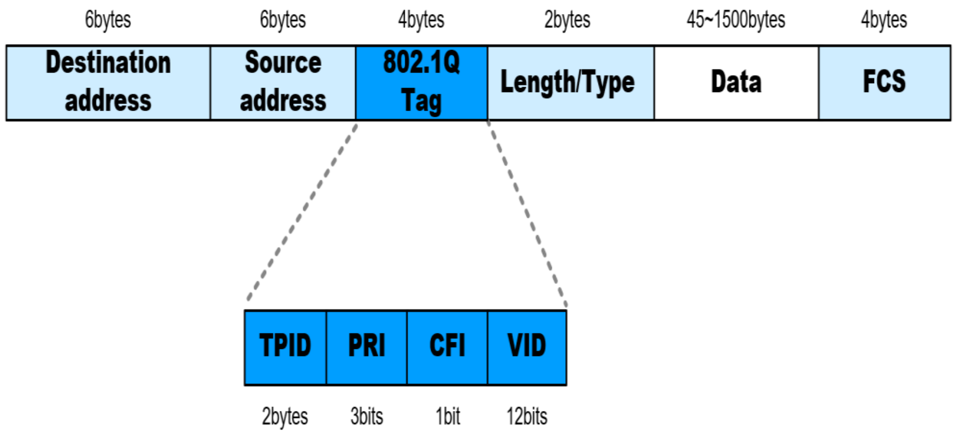
- A VLAN is a logical grouping of devices on a network, usually based on function or department, that allows network administrators to control network traffic and improve network performance and security.
- The 802.1q is a standard protocol that defines how VLAN information is carried over Ethernet networks. It adds a 4-byte VLAN tag to Ethernet frames, which allows switches and other network devices to identify the VLAN to which the frame belongs
- VLANs can improve network performance and security by reducing the amount of broadcast traffic on the network, enabling network administrators to apply security policies to specific VLANs, and allowing different VLANs to have different Quality of Service (QoS) policies to prioritize different types of traffic.
- Switches use the VLAN tag to forward traffic between VLANs. They maintain a VLAN database that maps VLAN IDs to physical switch ports and use this information to route traffic between VLANs based on the VLAN ID in the frame's VLAN tag.

VLAN tagging

- The VLAN tag consists of a 12-bit priority field, a 3-bit drop eligibility indicator, and a 16-bit VLAN ID field.
- An Ethernet frame is encapsulated with the Length/Type field for an upper-layer protocol following the Destination address and Source address fields.
- IEEE 802.1Q adds a 4-byte field between the Source address and the Length/Type fields of the original frame.



6bytes	6bytes	2bytes	45~1500bytes	4bytes
Destination address	Source address	Length/Type	Data	FCS



6bytes	6bytes	4bytes	2bytes	45~1500bytes	4bytes
Destination address	Source address	802.1Q Tag	Length/Type	Data	FCS

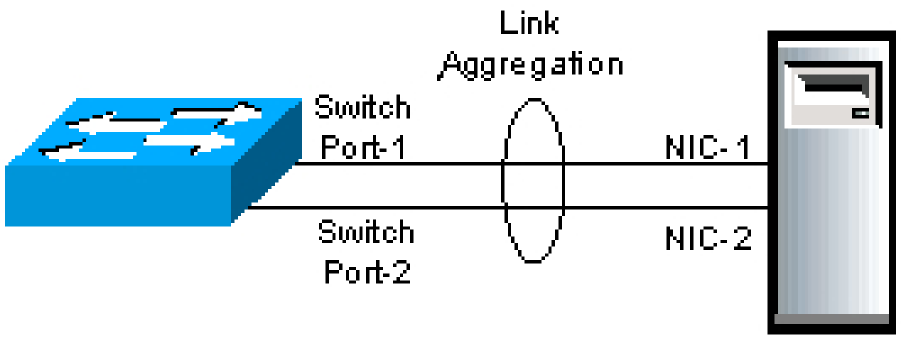
TPID	PRI	CFI	VID
2bytes	3bits	1bit	12bits

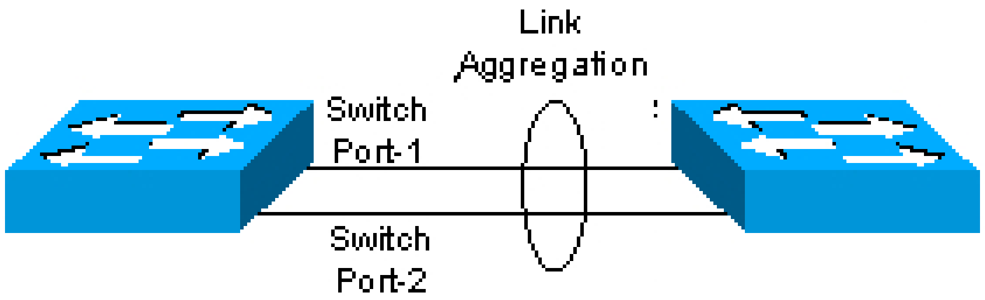
Link Aggregation Group (LAG)

- It is the aggregation of ports on a device. It can improve some aspects of transmission:
- **Increase in bandwidth**: the transfer speed of a LAG can reach the sum of the speeds of the individual ports
- **Fault tolerance**: the disconnection of a single link does not compromise the functionality of the LAG
- **Traffic balancing**: the LAG can be configured to route different packets through different ports of the aggregation, depending on the characteristics of the packet

Link Aggregation - dictionary

- **LA** - Link Aggregation: the combination of multiple physical links that work as if they were one. The standard that regulates LA is IEEE 802.3ad
    - Synonyms: trunking, bundling, bonding, port-channeling, teaming.
- **LAG** - Link Aggregation Group: a group of links that forms an aggregation. Each port of a switch can be part of a single LAG
- **LACP** – Link Aggregation Control Protocol: the protocol that declares the standard for the automatic configuration of a LAG (IEEE 802.3ad-2000)

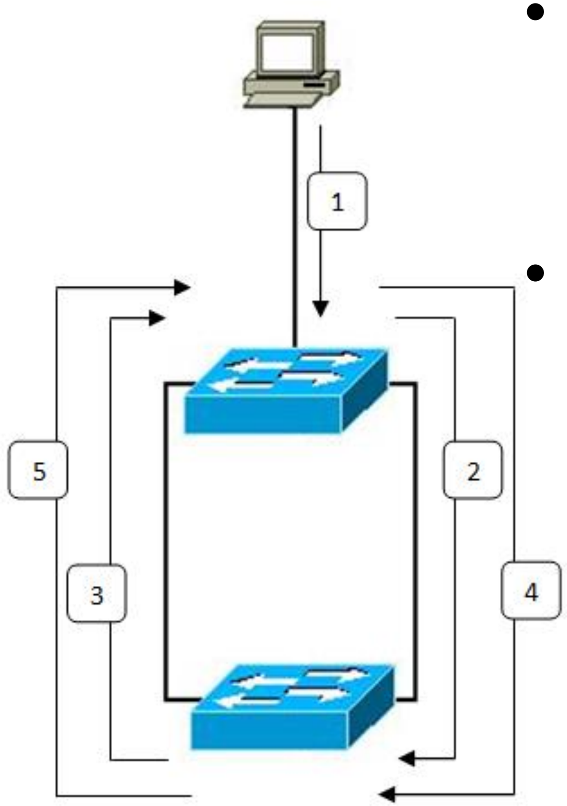




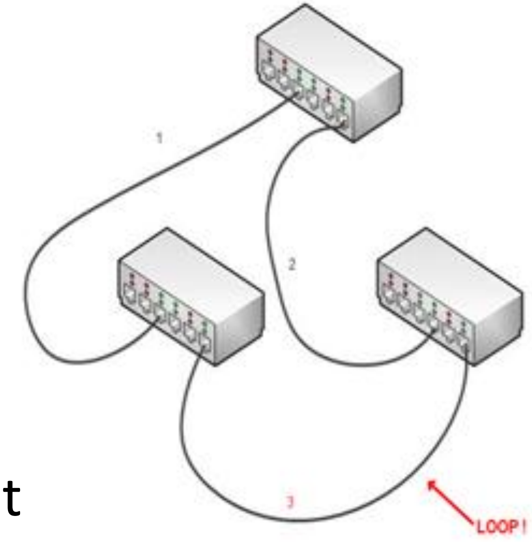
Frame distribution and LACP

- Data transmission through a LAG occurs by distributing the individual frames on the available ports.
- Generally, the distribution algorithms use a hash calculated on combinations of data included in the frame headers, at one or more levels of the stack:
  - L2, e.g.: destination mac address and/or source mac address
  - L3, e.g.: source IP address and/or destination IP address
  - L4, e.g.: source port and/or destination port
- **Static** LAG:
  - Both devices will be manually configured with the same parameters and the same number of ports
- **Dynamic** LAG:
  - Through the activation of the LACP protocol
- **LACP**: the operating principle consists in sending LACP packets to the partner equipment, directly connected and configured to use LACP.
- The LACP mechanism will allow to identify if the equipment in front supports LACP and will group the ports configured similarly (speed, duplex mode, VLAN, trunk vlan, etc.)

Avoiding ethernet switching loop



- A **network loop** occurs when a network has more than one active path that carries information from the same source to the same destination
- It can give rise to a broadcast storm that damages LAN availability:
  1. The host s**ends a broadcast message** on the network
  2. The first switch analyzes the packet received and **forwards** it to the lower part of the network
  3. The second switch **receives the copy** of the packet and operates accordingly as the first switch, forwarding it to the upper part of the network
  4. Since the packet is of broadcast type, the switches always repeat it on all ports excluding the port from which it arrived. The cycle is thus destined to **continue indefinitely**.

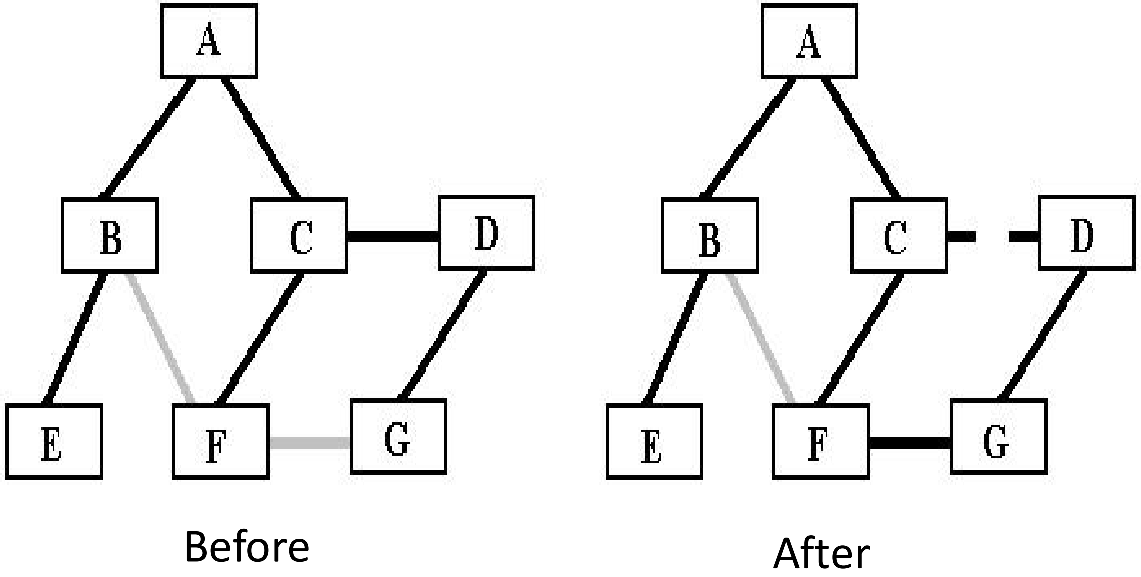


Spanning Tree Protocol (802.1d)

- The **802.1d** standard defines a protocol through which it is possible to realize a redundant topology avoiding ethernet loops.
- The switches communicate each other to elect the root spanning tree node which collects data and **reconstructs the network topology.**
- Based on the topology, one or more links of the LAN are identified and dynamically disabled in order to **remove circular paths.**
- The disabled links can be automatically re-enabled in the event of problems that generate a network partition, in order to regain connectivity
- STP can be implemented at VLAN level

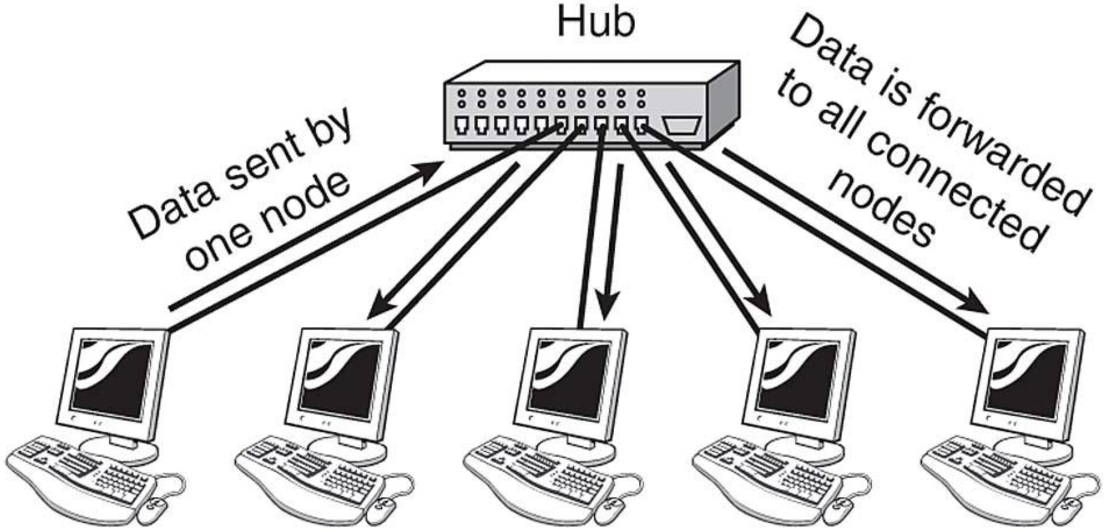
Spanning tree example

- Initially, the spanning tree protocol disables the links between switches B-F and F-G
- The failure of the C-D link makes D and G unreachable
- The activation of the F-G link restores connectivity to both switches without introducing circular paths that would give rise to a **switching loop**



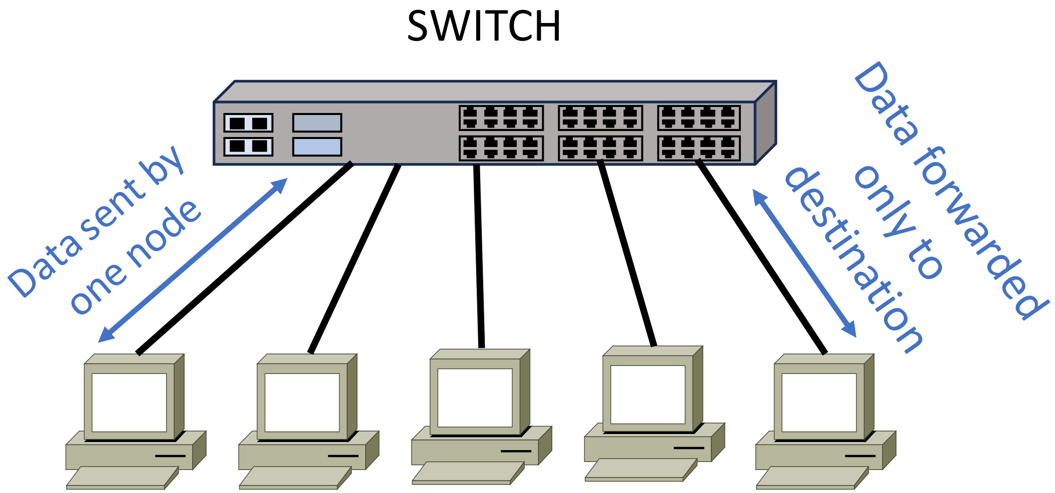
Hub

A hub is a simple network device (physical layer) that connects multiple devices together within a local area network. **It broadcasts all network traffic to all ports** and consequently to all connected devices, which can lead to network congestion and reduced performance. Hubs are generally not suitable for larger networks or networks that require high performance because they **do not provide dedicated bandwidth to each connected device** and can cause unnecessary network traffic.



Switch

A network switch commonly operates at the Data Link layer of the ISO/OSI model, uses MAC address to identify different network devices and provides a way for devices to communicate with each other.



- When a data packet arrives at a switch, the switch reads the destination address in the packet's header and forwards the packet to the **appropriate port** that connects to the destination device.
- When a switch receives a data packet, it examines the source MAC address in the packet's header and stores this address along with the port number that the packet was received on in its MAC address table. The switch then forwards the packet to the appropriate port based on the destination MAC address in the packet's header.
- If the switch receives a packet with a destination MAC address that is not in its MAC address table, it will broadcast the packet to all ports on the switch except the port that received the packet. This process is known as flooding, and it is used to ensure that the packet reaches its intended destination device. When the destination device responds, the switch adds the device's MAC address to its MAC address table so that future packets can be forwarded directly to the device.

Switch

There are several types of switches: 1U stand alone and modular switches. layer 2 and layer 3 switches.

Characterized by the aggregate throughput and by the speed of the interfaces from 1 Gbps to 400 Gbps

Layer3 switches can operate at the Network layer of the ISO/OSI model and provide **advanced routing and filtering capabilities.**

Switches now can support advanced features such as quality of service (QoS), virtual LANs (VLANs

Modern L3 switches can operate a routers including dynamic routing protocols like OSPF and BGP and supports and overlay protocols such as EVPN VXLAN.

MAC address Table example

```
 vlan   mac address     type    learn      age      ports
------+----------------+--------+-----+----------+--------------
*  126  001e.791d.0400    static  No            -   Router
* 2048  ccc5.e5f8.cd47   dynamic  Yes           5   Po1
*    8  a078.175b.dbec   dynamic  Yes          15   Gi3/37
*    6  0050.b625.e946   dynamic  Yes           0   Gi8/40
* 1052  dcf4.012c.3a4d   dynamic  Yes         120   Po1
*    4  88ae.dd59.ced4   dynamic  Yes           0   Po20
*    6  001a.4aa8.3107   dynamic  Yes         190   Po1
*    6  482a.e373.0c06   dynamic  Yes           0   Gi3/13
* 2048  ccc5.e5f8.d3b6   dynamic  Yes           0   Po1
*    1  44d9.e7fc.b1ae   dynamic  Yes           0   Gi3/12
```

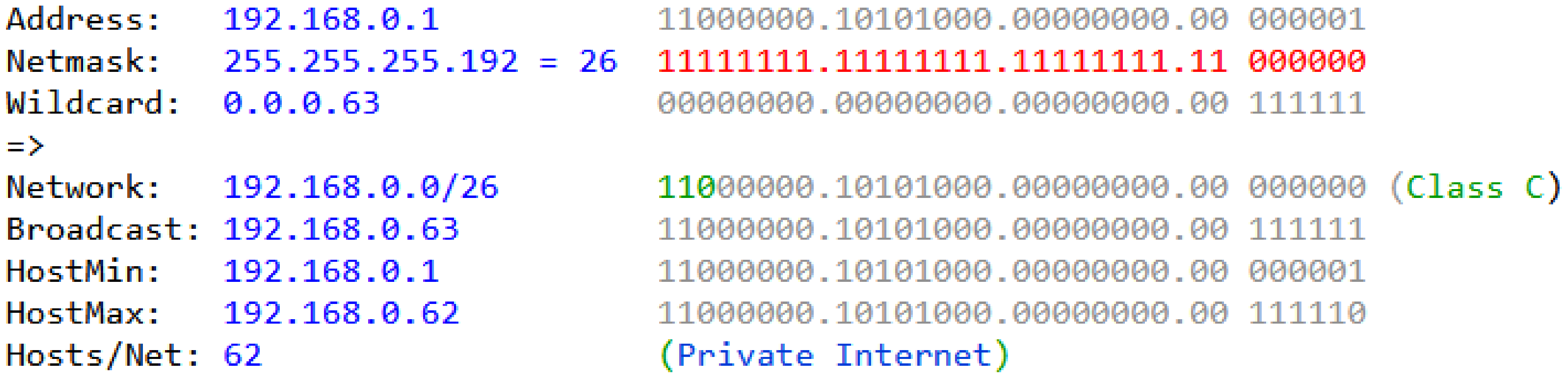
Internet Protocol

Internet Protocol addressing: why ?

- The IP project had one main goal: to allow hosts interconnected by any Local Area Network protocol to talk to each other.
- This required *at least* a **global** scheme for identifying/addressing each host.
- We'll be covering the two versions of the IP protocols in use (v4 and v6) in parallel, as the semantics is *the same*.

Internet Protocol IPv4 addressing

- The IP address is divided into a network part (fixed) and a host part (variable), which determines the group of hosts belonging to the same network and the number of possible hosts in that network.
- The netmask identifies which part of the IP belongs to network or host

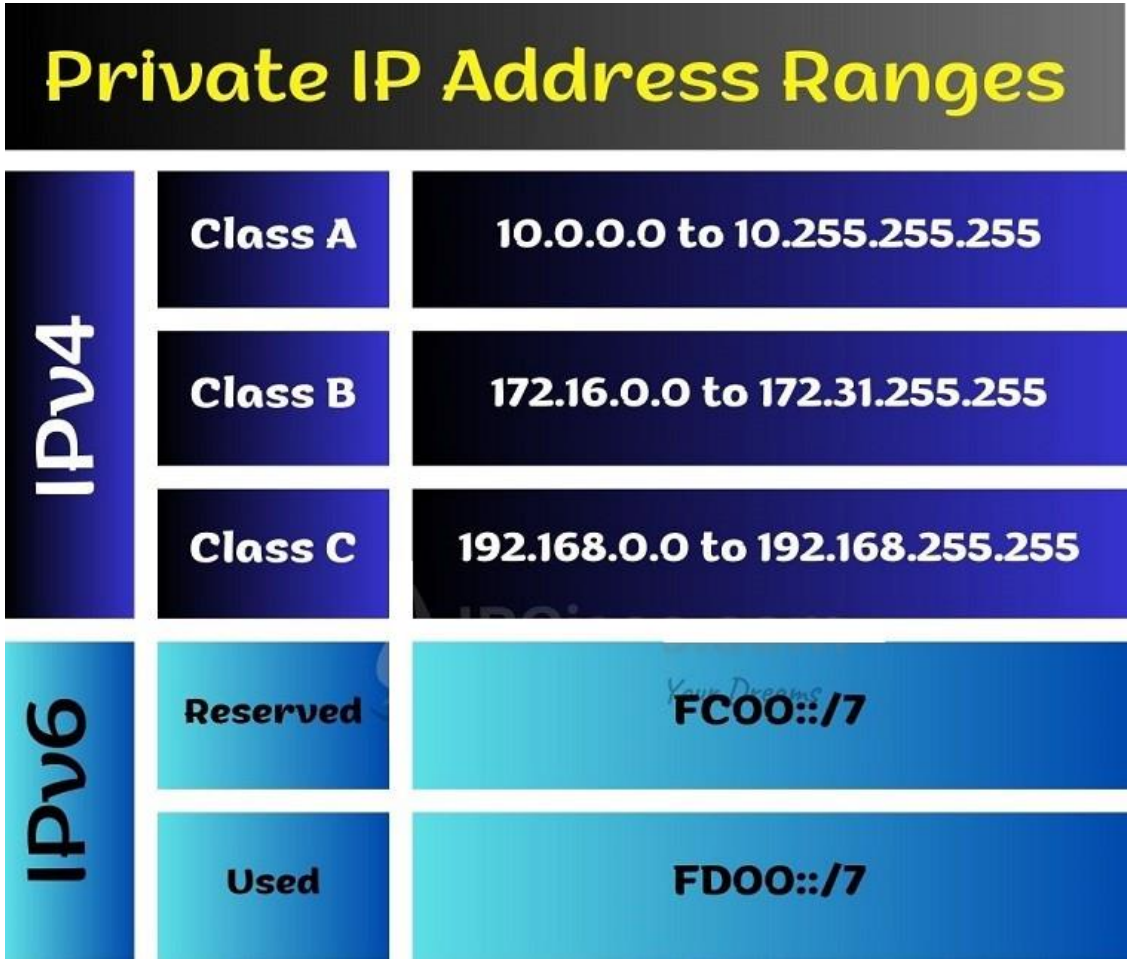


```
Address:   192.168.0.1           11000000.10101000.00000000.00 000001
Netmask:   255.255.255.192 = 26  11111111.11111111.11111111.11 000000
Wildcard:  0.0.0.63              00000000.00000000.00000000.00 111111
=>
Network:   192.168.0.0/26        11000000.10101000.00000000.00 000000 (Class C)
Broadcast: 192.168.0.63          11000000.10101000.00000000.00 111111
HostMin:   192.168.0.1           11000000.10101000.00000000.00 000001
HostMax:   192.168.0.62          11000000.10101000.00000000.00 111110
Hosts/Net: 62                    (Private Internet)
```

- The same for IPv6 but 128-bit long

Private IP Address ranges

- Private IP networks are local networks that use reserved IP addresses that are not routable on the Internet
- Private IP networks have the advantage of reducing the requests for public IP addresses



Private IP Address Ranges

IPv4	Class A	10.0.0.0 to 10.255.255.255
	Class B	172.16.0.0 to 172.31.255.255
	Class C	192.168.0.0 to 192.168.255.255
IPv6	Reserved	FC00::/7
	Used	FD00::/7

Address notation

IPv4	IPv6
A 32-bit number is used to identify a host globally. Notation: *dotted quad* (4 bytes in **decimal** format, separated by dots): 192.167.140.123 - 127.0.0.1	A 128-bit number is used to identify a host globally Notation: 8 16-bit words in **hexadecimal** format**,** separated by colons. A double colon represents *one* contiguous range of zero bits: 2001:0760:4206:baba:cafe:f00d:0000:0001 == 2001:760:4206:baba:cafe:f00d::1 - ::1
Hosts that communicate to each other directly (no routing needed) are said to belong to the same subnet and share a leading address *prefix.* Bit count format: 192.167.140.123/24 *Or* Subnet mask: 255.255.255.0	The same goes for IPv6, but the subnet mask format would be too clumsy and was dropped: 2001:0760:4206:baba::/64 /64 subnets are the most frequently used for IPv6 – almost a default.
IP addresses that circulate globally must be unique => their assignment has to be managed. A host can be configured with any number (and any 6/4 combination) of addresses assigned to it. A global Domain Name resolution Service exists, that can translate both styles of addressed to and from a human-readable names. More on this later.	

Address assignment

IPv4	IPv6
Manual address assignment is always an option!	
'Zero configuration' protocols can automatically assign a *link-local* address (in the 169.254.0.0/16 subnet).	A link-local address in the fe80::/10 subnet is ***always*** automatically assigned to *any* IP-enabled interface.
DHCPv4 (Dynamic Host Configuration Protocol): a request is broadcast on the local network – where the requesting host is usually identified by its Ethernet address. A server then replies with address assignment and other config info.	DHCPv6: almost the same, except that the host and interface issuing the request are identified by unique DUID and IAID numbers. The Ethernet address of the interface *may* be used to generate the DUID, but this can also be randomized for privacy.
	Stateless autoconfiguration (SLAAC): the non-subnet (least significant) part of the *public* address is assigned automatically by the host using a unique ID known to the host. Usually the DUID (see above).

ARP and Neighbor Discovery

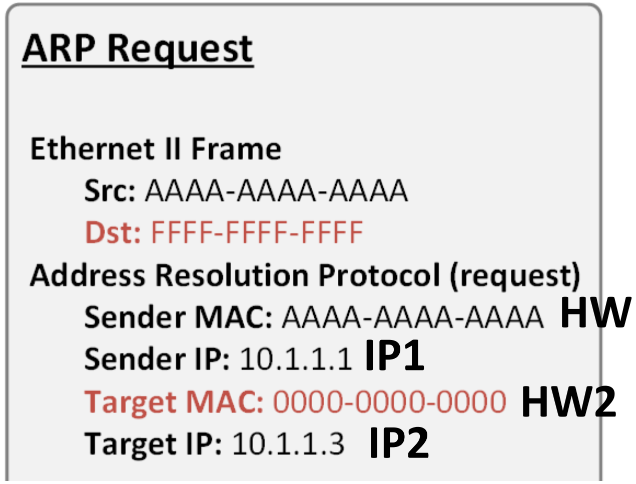
Hosts are *only* able to communicate via their local LAN protocol (e.g., Ethernet). **Some mechanism to translate destination IP addresses to the corresponding LAN (Ethernet) address is needed.**

IPv4	IPv6
The Address Resolution Protocol (ARP), that transfers messages using directly the LAN (e.g. Ethernet) protocol was developed. An ARP request is broadcast on a network segment, triggering a reply identifying the LAN (MAC/Ethernet) address of the destination node.	Every IPv6-enabled interface is able to communicate via IPv6 out of the box, via link-local addresses. The resolution of IP addresses to LAN address is done via ICMPv6 'Neighbour disccovery' messages.

```
$ ip neigh show nud reachable
192.135.8.190 dev br0 lladdr 14:9d:99:83:4b:6f REACHABLE
192.84.138.254 dev br0 lladdr f8:c1:16:68:f7:00 REACHABLE
2001:760:4210:1::21:180 dev br0 lladdr 00:26:18:fd:a1:9e REACHABLE
```

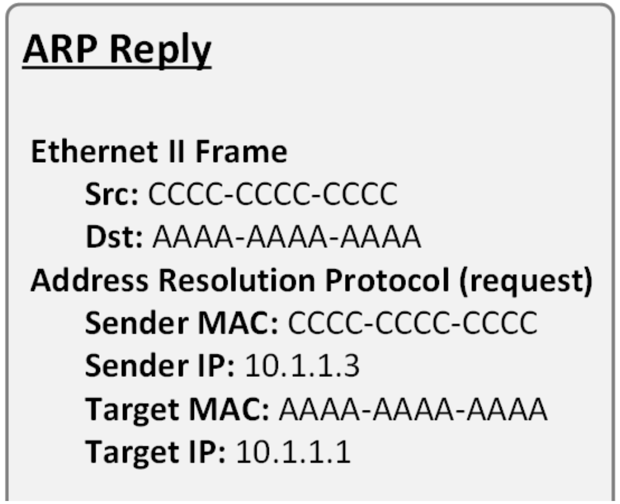
ARP mechanism

- When a host with IP address IP1 and hardware address HW1 has to send an IP packet to a host with IP address IP2 on the same network, ARP obtains the necessary information in this way:
  - a data link packet (ARP request) containing **IP1, HW1,** and **IP2** is built, with a field dedicated to **HW2 filled with all 0s**
  - this packet is **broadcast** on the local network
  - **everyone receives the ARP packet**, but only the host that has the IP2 address processes it (the others discard it)
  - the destination host builds a data link packet (ARP response) containing the missing information, and sends it directly to HW1 (not broadcast)
  - ARP on the first host then acquires the information of the Ethernet address of the remote host, and communicates it to IP, which can thus encapsulate its own IP packets in frames of the data link protocol addressed to the correct destination



ARP Request

Ethernet II Frame
    Src: AAAA-AAAA-AAAA
    Dst: FFFF-FFFF-FFFF
Address Resolution Protocol (request)
    Sender MAC: AAAA-AAAA-AAAA **HW1**
    Sender IP: 10.1.1.1 **IP1**
    Target MAC: 0000-0000-0000 **HW2**
    Target IP: 10.1.1.3 **IP2**



ARP Reply

Ethernet II Frame
    Src: CCCC-CCCC-CCCC
    Dst: AAAA-AAAA-AAAA
Address Resolution Protocol (request)
    Sender MAC: CCCC-CCCC-CCCC
    Sender IP: 10.1.1.3
    Target MAC: AAAA-AAAA-AAAA
    Target IP: 10.1.1.1

ARP Table example

```
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.150.253        62   c89c.1d67.f9ff  ARPA   Vlan1
Internet  192.168.150.250         0   5254.007f.e625  ARPA   Vlan1
Internet  192.168.150.249       225   5254.005b.6e23  ARPA   Vlan1
Internet  192.168.10.74           8   544b.8ce3.8dcf  ARPA   Vlan126
Internet  192.168.10.73           -   001e.791d.0400  ARPA   Vlan126
Internet  192.168.150.158        31   c89c.1def.567f  ARPA   Vlan1
Internet  192.168.150.156       211   6c8b.d380.bb1f  ARPA   Vlan1
Internet  192.168.150.157         -   001e.791d.0400  ARPA   Vlan1
Internet  192.168.150.140        95   24e9.b327.3bbf  ARPA   Vlan1
Internet  192.168.150.139        94   24e9.b327.37bf  ARPA   Vlan1
Internet  192.168.150.89          0   0004.9636.a0e0  ARPA   Vlan1
```

ARP Cache

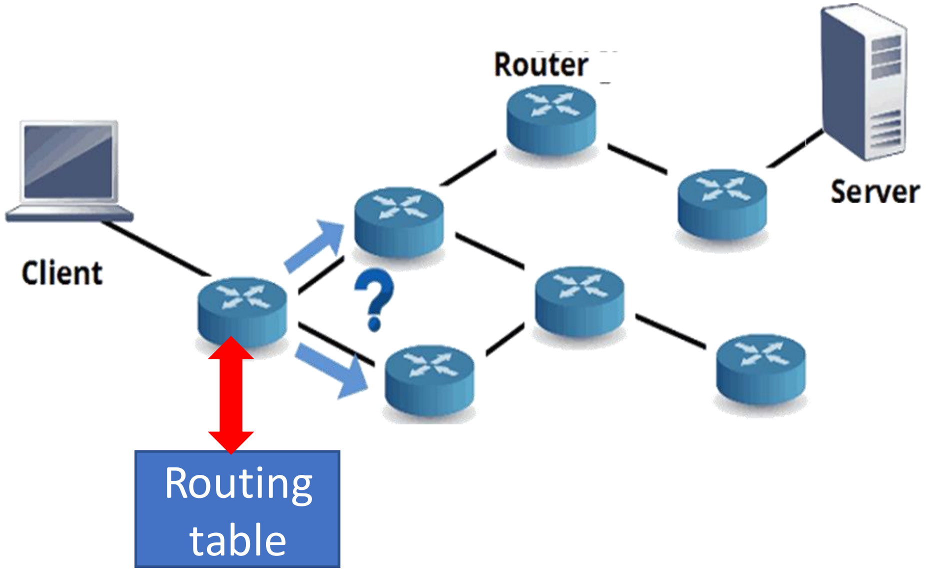
- To improve performance, ARP can manage a cache in memory on the local host
- Every time a new IPaddress - HWaddress association is learned, it is stored in the cache
- When ARP has to locate an HW address, it first checks in the cache: if the information is present it is used without sending packets on the network
- The entries in the ARP cache have an expiration time, to avoid that events such as replacement of network cards or redirection of hosts can make communication impossible
  - At the expiration of the validity time the entry is removed from the cache, and a subsequent request for that address will cause a new emission of ARP request on the LAN

IP Routing

“In a packet switching system, routing refers to the process of choosing a path over which to send packets, and router refers to any computer making such a choice” - Douglas E. Comer ‘Internetworking with TCP/IP’.

- The router is a device with at least two interfaces that connects two or more different IP networks.

- Every router has a so-called routing table which contains the list of all reachable destinations.



Routing types

The path selection can be based on static or dynamic routes:

- Static: predefined routes, manually configured by the network manager.
- Dynamic: routes generated by an algorithm specific to the routing protocol.

Routing protocols

Basically, there are two kind of dynamic protocols:

- **Distance Vector:** in these protocols, each router **does not possess information about the full network topology**. It advertises its distance value (DV) calculated to other routers

- **Link State:** in link-state routing protocols, **each router possesses information about the complete network topology.** Each router then independently calculates the best next hop from it for every possible destination in the network using local information of the topology. The collection of best-next-hops forms the routing table.

Distance vector protocols

- **RIP**: Routing Information Protocol (RIP) is a dynamic routing protocol that uses hop count as a routing metric to find the best path between the source and the destination network. Hop count is the number of routers occurring in between the source and destination network. **The path with the lowest hop count is considered as the best route to reach a network and therefore placed in the routing table.** RIP prevents routing loops by limiting the number of hops allowed in a path from source and destination. **The maximum hop count allowed for RIP is 15.**
- **IGRP**: It supports **multiple metrics for each node** which includes delay, load, and bandwidth, to compare the 2 routes which are combined into single metrics. By default, every 90 seconds it updates the routing information.
- **EIGRP**: It’s an **evolution of IGRP**, in a stable configuration only few protocol packets need to be transmitted and when network topology changes occur only the route changes are transmitted. Furthermore, convergency time is significantly lower than IGRP, something can be almost instantaneous.

Link state protocols

OSPF: Based on **Dijkstra's** algorithm, it calculates the shortest path to all destinations. Upon initialization or due to any change in routing information, a router generates a link-state advertisement. All these information are collected by every router and used to compute the shortest path tree, destinations, associated cost, and next hop to reach those destinations form the IP routing table.

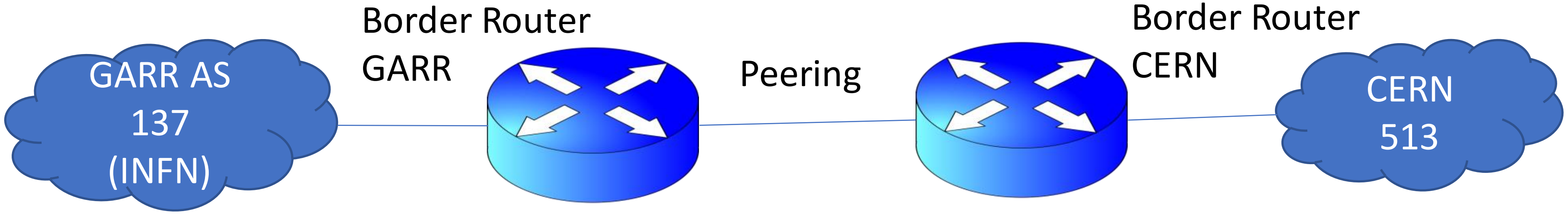
IS-IS: As OSPF, it uses the same algorithm but has two major strengths. The first is its **scalability**. It's much easier to build large networks with IS-IS than it is with OSPF. This makes it a common choice with service providers for their infrastructure. The second strength is its **agnostic** approach to the data it carries. IS-IS carries a payload of reachability data, but for the most part it doesn't care what's in the payload.

Border Gateway Protocol

An **Autonomous System (AS)** is a set of a networks that are all managed, controlled and supervised by a single entity or organization

BGP is the protocol that enables different AS to exchange routing information. This connection is called peering.

Routers that runs BGP are usually called border gateway.



Routing Table example

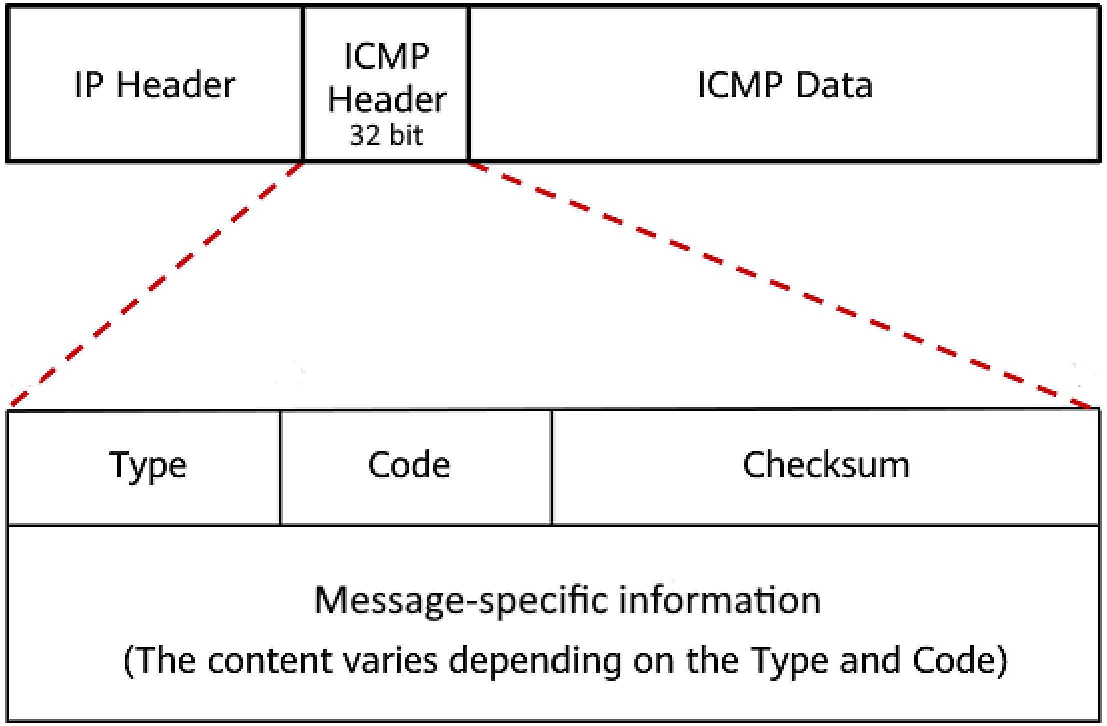
```
#show ip route
Gateway of last resort is 193.206.128.17 to network 0.0.0.0
B*    0.0.0.0/0 [20/0] via 193.206.128.17, 7w0d
      10.0.0.0/8 is variably subnetted, 28 subnets, 5 masks
C        10.0.0.0/22 is directly connected, Vlan260
L        10.0.1.254/32 is directly connected, Vlan260
C        10.10.16.0/24 is directly connected, Vlan1116
L        10.10.16.1/32 is directly connected, Vlan1116
S        10.10.23.0/24 [1/0] via 192.168.251.2
S        10.10.25.0/24 [1/0] via 192.168.150.99

B – BGP C – Connected L – Local S- Static
```

Internet Control Message Protocol

- ICMP (Internet Control Message Protocol) is an error-reporting protocol that network devices such as routers use to generate error messages to the source IP address when network problems prevent delivery of IP packets. **It does not provide suggestions about the action to take in response to error reports.**

- Unlike the Internet Protocol, ICMP is not associated with a transport layer protocol such as TCP or UDP. This makes ICMP a connectionless protocol: one device does not need to open a connection with another device before sending an ICMP message.

ICMP message format



IP Header
ICMP Header 32 bit
ICMP Data
Type
Code
Checksum
Message-specific information
(The content varies depending on the Type and Code)

Type	ICMP Message Type
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect (change a route)
8	Echo Request
30	Traceroute

Code	Destination Unreachable Code (some examples)
1	Host is unreachable
4	Fragmentation is needed and Don't Fragment was set
6	Destination network is unknown

Your network companion: “The ping”

One of the mostly used application of the ICMP protocol is **ping**, its main scope is to check host reachability and measure the round trip time. RTT, is the amount of time it takes for a signal to be sent plus the amount of time it takes for acknowledgement of that signal having been received.

Ping, without any doubt, represents one of the most used tools to carry out troubleshooting operations.

```
ping www.infn.it
Pinging wwwinfn.lnf.infn.it [193.206.84.44] with 32 bytes of data:
Reply from 193.206.84.44: bytes=32 time=10ms TTL=54
Reply from 193.206.84.44: bytes=32 time=10ms TTL=54
Reply from 193.206.84.44: bytes=32 time=10ms TTL=54
Reply from 193.206.84.44: bytes=32 time=10ms TTL=54

Ping statistics for 193.206.84.44:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 10ms, Maximum = 10ms, Average = 10ms
```

Your network companion: “traceroute”

```
$ traceroute www.infn.it

traceroute to www.infn.it (193.206.84.44),
 1   switch-lan-cnaf-3.infn.it (131.154.3.57)
 2   ru-cnaf-rx1-bo1.bo1.garr.net (193.206.128.17)
 3   rl1-bo01-rs1-bo01.bo01.garr.net (185.191.180.53)
 4   rs1-bo01-rs1-rm02.rm02.garr.net (185.191.181.76)
 5   rx2-rm2-rx1-fra.fra.garr.net (90.147.81.162)
 6   193.206.136.62 (193.206.136.62)
 7   lnfgw-lnf.lnf.infn.it (193.205.228.122)
 8   swcorebb900.lnf.infn.it (193.205.228.118)
 9   wwwinfn.lnf.infn.it (193.206.84.44)
```

traceroute (or **tracert**) is a command that helps you find out the path taken by a packet from source to destination reporting the number of hops

```
$ traceroute6 www.mi.infn.it

traceroute to www.mi.infn.it (2001:760:4210:1::13), 30 hops max, 80 byte packets
 1   gw2-128.cr.cnaf.infn.it (2001:760:4205:128::52)  0.793 ms  0.856 ms  0.904 ms
 2   fd00:0:0:150::156 (fd00:0:0:150::156)  0.353 ms  0.402 ms  0.540 ms
 3   2001:760:ffff:110::1c (2001:760:ffff:110::1c)  0.425 ms  0.405 ms  0.407 ms
 4   2001:760:ffff:ffaa::180:51 (2001:760:ffff:ffaa::180:51)  3.300 ms  3.493 ms  3.674 ms
 5   2001:760:ffff:ffbb::180:172 (2001:760:ffff:ffbb::180:172)  3.131 ms  3.190 ms  3.114 ms
 6   rx1-mi3-ru-infnmi.mi3.garr.net (2001:760:ffff:128::45)  4.495 ms  4.263 ms  4.235 ms
 7   joomla-mi.mi.infn.it (2001:760:4210:1::13)  3.203 ms !X  3.225 ms !X  3.235 ms !X
```

ICMP in IPv6?

Internet Control Message Protocol version 6 (ICMPv6) is the implementation of the Internet Control Message Protocol for Internet Protocol version 6. ICMPv6 is an integral part of IPv6 and performs error reporting and diagnostic functions. The new protocol extensions make ICMPv6 mandatory for IPv6 to function properly.

Of course, there is a version 6 of ping:

```
ping6 -c4  storm-cms.cr.cnaf.infn.it.
PING storm-cms.cr.cnaf.infn.it.(storm-cms.cr.cnaf.infn.it  (2001:760:4205:128::128:27)) 56 data bytes
64 bytes from storm-cms.cr.cnaf.infn.it  (2001:760:4205:128::128:27): icmp_seq=1  ttl=64 time=0.239 ms
64 bytes from storm-cms.cr.cnaf.infn.it  (2001:760:4205:128::128:27): icmp_seq=2  ttl=64 time=0.205 ms
64 bytes from storm-cms.cr.cnaf.infn.it  (2001:760:4205:128::128:27): icmp_seq=3  ttl=64 time=0.276 ms
64 bytes from storm-cms.cr.cnaf.infn.it  (2001:760:4205:128::128:27): icmp_seq=4  ttl=64 time=0.268 ms
--- storm-cms.cr.cnaf.infn.it.  ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev  = 0.205/0.247/0.276/0.027 ms
```

TCP/UDP (Transport Layer)

Transport Layer (Layer 4)

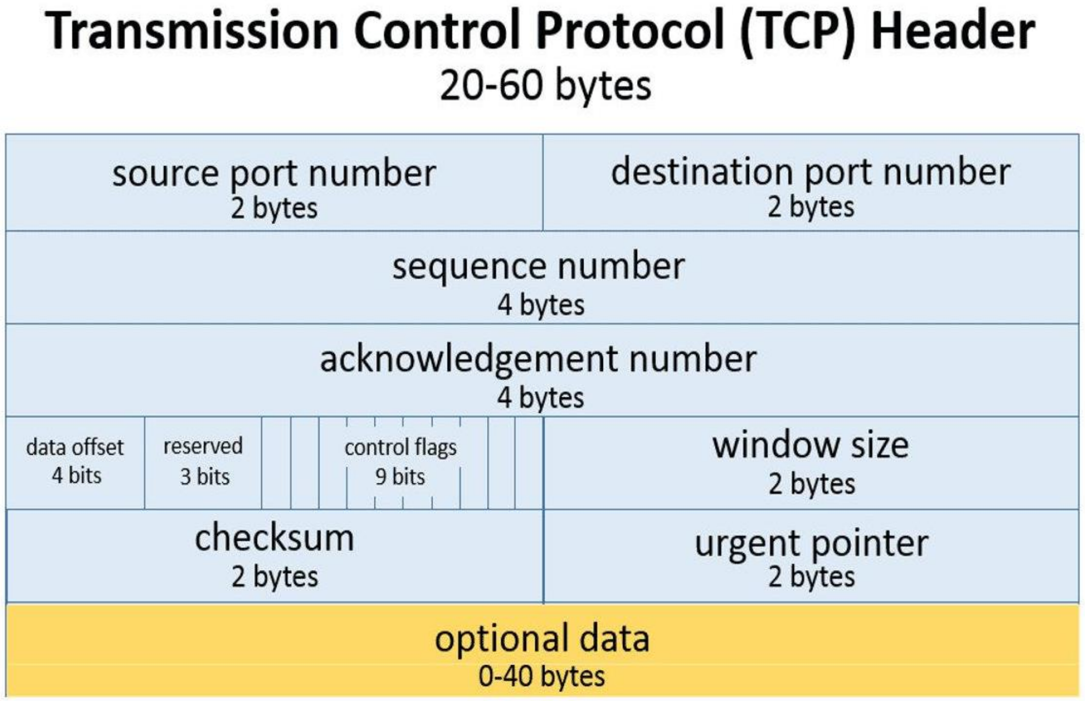
- The transport layer main goal is to provide end-to-end communication services for applications.

- The transport layer's tasks include error correction as well as segmenting and de-segmenting data before and after it's transported across the network. This layer is also responsible for flow control and making sure that segmented data is delivered over the network in the correct sequence.

- The main actors in this layer are the Transmission Control Protocol (TCP) and User Data Protocol (UDP) to carry out its tasks.

Transmission Control Protocol

- **Connection oriented**:
  - Establishing a TCP (3way handshake) connection permits data to be transferred from both sides.
  - Reliable (retransmission)
  - Error detection (checksum)
  - Slow start
  - Congestion management
  - Slower than UDP but reliable, is used by application that needs a guaranteed delivery of the data like HTTP/S, FTP, SMTP, IMAP, etc...



Transmission Control Protocol (TCP) Header

20-60 bytes

source port number 2 bytes | destination port number 2 bytes

sequence number 4 bytes

acknowledgement number 4 bytes

data offset 4 bits | reserved 3 bits | control flags 9 bits | window size 2 bytes

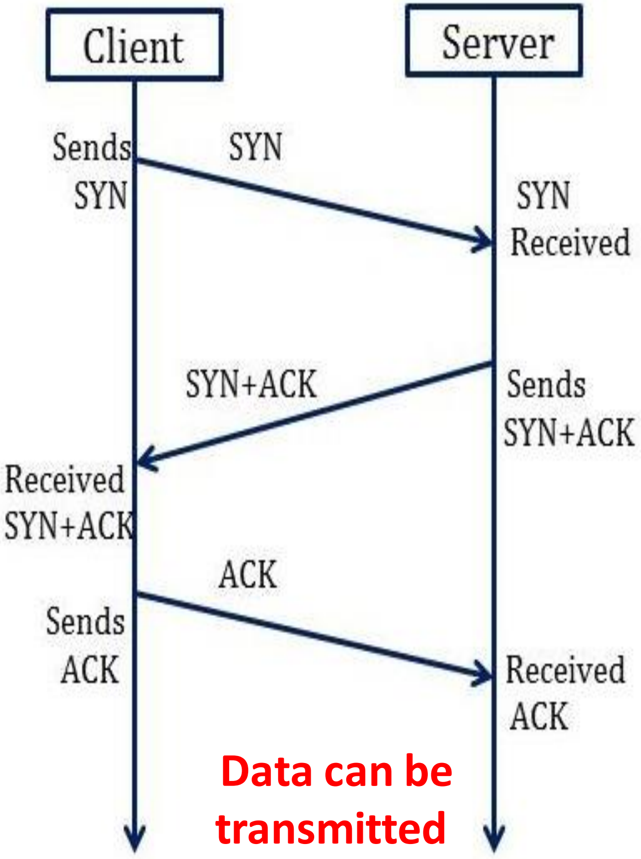
checksum 2 bytes | urgent pointer 2 bytes

optional data 0-40 bytes

TCP connection (3-way handshake)

1. **Synchronization Sequence Number (SYN) The client sends the SYN to the server**
   When the client wants to connect to the server sends the message to the server by setting the SYN flag as 1.
   The ACK is set to 0.
2. **Synchronization and Acknowledgement (SYN-ACK) to the client**
   The server acknowledges the client request by setting the ACK flag to 1.
   The ACK indicates the response of the segment it received, and SYN indicates with what sequence number it will start the segments.
   The server will set the SYN flag to '1' and send it to the client if the server also wants to establish the connection.
3. **Acknowledgment (ACK) to the server**
   The client sends the acknowledgment (ACK) to the server after receiving the synchronization (SYN) from the server.
   After getting the (ACK) from the client, the connection is **ESTABLISHED** between the client and the server.

The connection will eventually end with an RST (reset or tear down the connection) or FIN (gracefully end the connection).



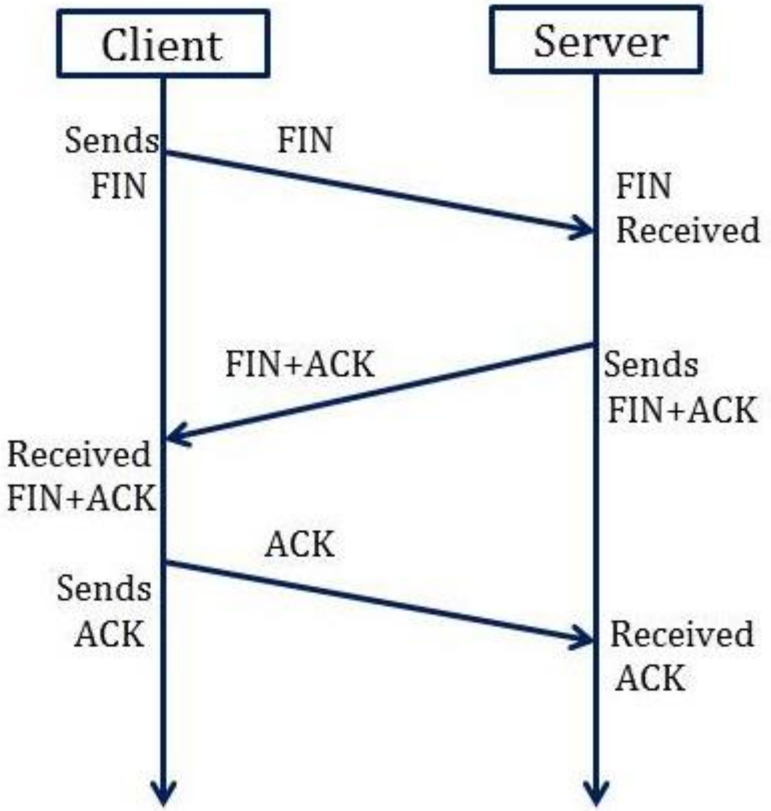
Data can be transmitted between the client and server

Closing a TCP connection

3 -Way Handshake Closing Connection Process

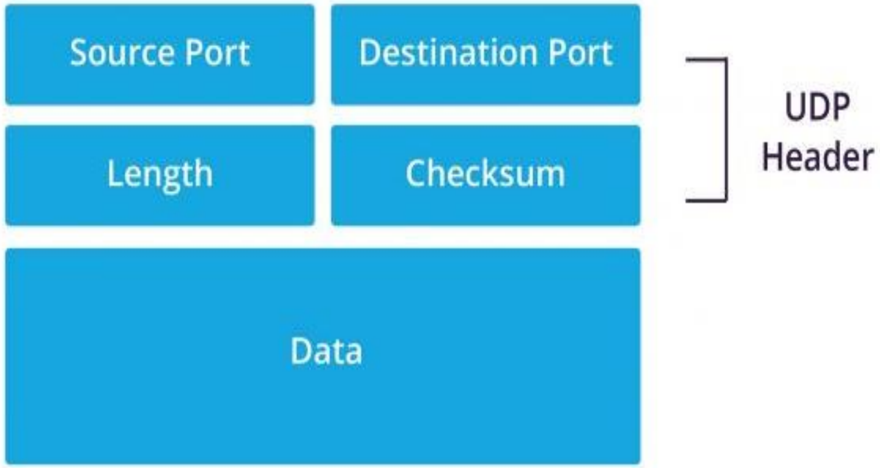
To close a 3-way handshake connection,

- First, the client requests the server to terminate the established connection by sending FIN.
- After receiving the client request, the server sends back the FIN and ACK request to the client.
- After receiving the FIN + ACK from the server, the client confirms by sending an ACK to the server.



User Datagram Protocol (UDP)

- **Connectionless**:
  - User Datagram Protocol (UDP) is a communications protocol that is primarily used to establish low-latency and loss-tolerating connections between applications on the internet.



- Mostly used for process-to-process communication
- No overhead due to a connection creation
- Not reliable
- Faster than TCP is used for real time application like streaming or video calls

TCP and UDP ports

- A single IP address permits the establishment of a single connection between two hosts, since we want to make possible the communication among multiple hosts it’s mandatory to introduce the concept of ‘port’. The port number is a 16-bit number (0-65535).This range has been separated by the IANA (Internet Assigned Numbers Authority) into several different segments:
  - Port 0 is not used for internet/network traffic, but it’s sometimes utilized in communications going down between different programs on identical computers.
  - Ports 1-1023 are alluded to as system ports (a.k.a. **well-known** ports).
  - Ports 1024-49151 are known as registered ports.
  - Port 49152-65535 are known as ephemeral ports.

Examples

TCP 80 HTTP
TCP 443 HTTPS
TCP 22 SSH

UDP 161 SNMP
UDP 53 DNS

TCP/UDP typical flows

TCP (HTTP)

c/s:open connection (host, port)

- c:http get (url)
- s:http response (status,headers,body)
- c:http post (form)
- s:http response (status, headers, body)

c/s :close connection

UDP (DNS)

c:dns query (domain name,type,class)

s:dns response (domain name,type,class,ttl,data)

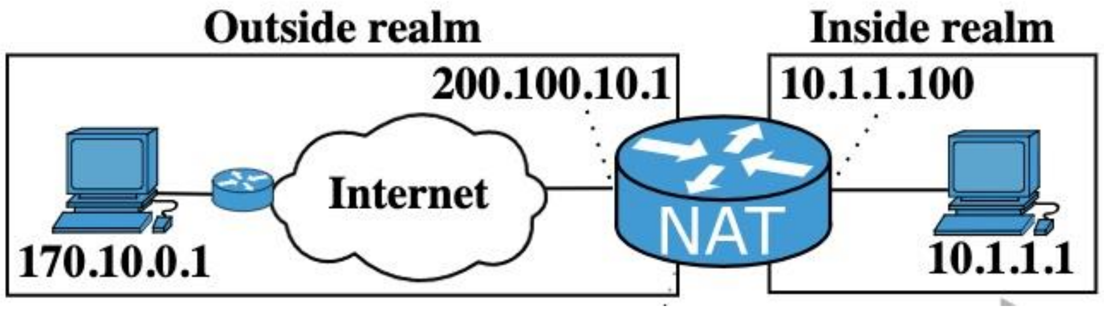
c:dns query (domain name,type,class)

s:dns response (domain name,type,class,ttl,data)

NAT: IP Network Address Translation

Network Address Translator [RFC 2663] is a method by which IP addresses are mapped from one realm to another, in an attempt to provide transparent routing to hosts.

Address translation allows for example hosts in a private network to transparently communicate with destinations on an external network.



NAT: How it works

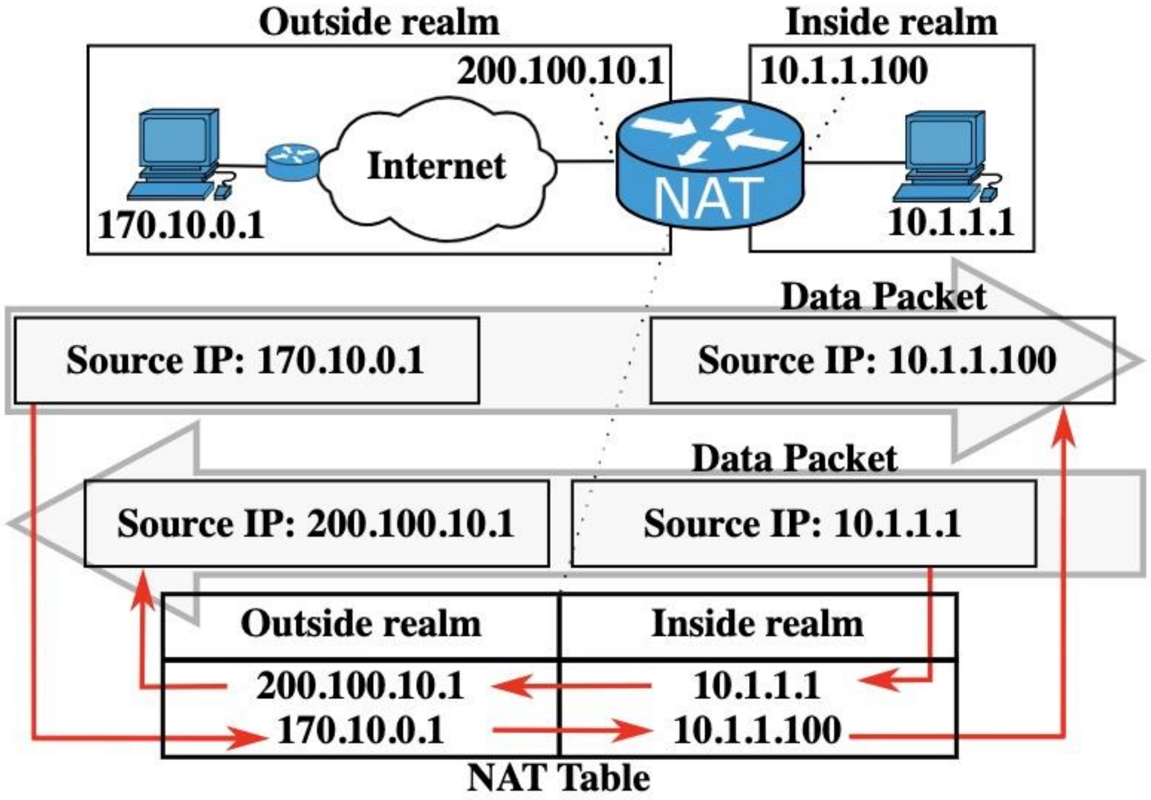
NAT modifies the addresses in the IP header of a packet to be valid in the addresses realm into which the datagram is routed.

The translations are session based.

E.g., TCP sessions are uniquely identified by the tuple of

(source IP address, source TCP port, target IP address , target TCP port)

The NAT service maintains a ***translation table*** for every sessions.



Outside realm	Inside realm
200.100.10.1	10.1.1.1
170.10.0.1	10.1.1.100

NAT: Main usage @INFN

NAT in INFN computing infrastructures is mainly used to:

- provide outbound connectivity for computing farm nodes with IP addresses in private range [rfc 1918].
- provide WiFi connectivity for non local users (e.g. eduroam)

Any other circumstance where insulation and connectivity are needed at the same time.

NAT: implementation

NAT can be done in several different ways. Here are some examples:

- enabling NAT on a network device (switch/router);
- iptables on a linux host;
- pf on a FreeBSD host;

- any other circumstance where insulation and connectivity are needed at the same time.

NAT: pf on FreeBSD

The **pf** firewall is a BSD licensed stateful packet filter.

It should be enabled in **/etc/rc.conf**:

```
pf_enable="YES”
pf_rules="/etc/pf.rules"
```

Here is a simple configuration **(/etc/pf.rules**):

```
ext_if = "bce0"               # External network interface for IPv4
ext_addr = "193.205.66.144"   # External IPv4 address (i.e., global)
int_if = "ix1"                # Internal network interface for IPv4
int_addr = "192.168.128.254"  # Internal IPv4 addr (i.e., gateway for priv net)

nat log (to pflog1) on $ext_if from 192.168.128.0/24 to any -> 193.205.66.144
```

NAT translation table example

```
#show ip nat translations

Pro     Inside global         Inside local          Outside local         Outside global
tcp     131.154.4.3:4623      172.16.10.40:33938    52.108.196.24:443     52.108.196.24:443
tcp     131.154.4.3:4119      172.16.10.40:37550    142.251.209.14:443    142.251.209.14:443
tcp     131.154.4.3:4919      172.16.10.40:38656    52.113.194.132:443    52.113.194.132:443
udp     131.154.4.3:1408      172.16.1.10:47659     217.61.62.224:123     217.61.62.224:123
udp     131.154.4.3:1467      172.16.1.14:123       162.159.200.1:123     162.159.200.1:123
```

NAT Public IP

Hosts Private IP

Public IP (Destinations)

Application layer

DHCP

Why do we need dynamic IP assignment?

- To communicate via TCP/IP a host must
  - know its own IP address
  - know its own network (netmask)
  - know the address of a router on its own network
- In addition to this, the complete functioning of the TCP/IP protocols require other information
  - the address of one or more DNS servers
  - its own domain name
  - the address of one or more NTP servers
  - the address of one or more WINS servers
- This information can be statically configured on the host, **however, it’s convenient to dynamically configure it during the network configuration process.**

DHCP

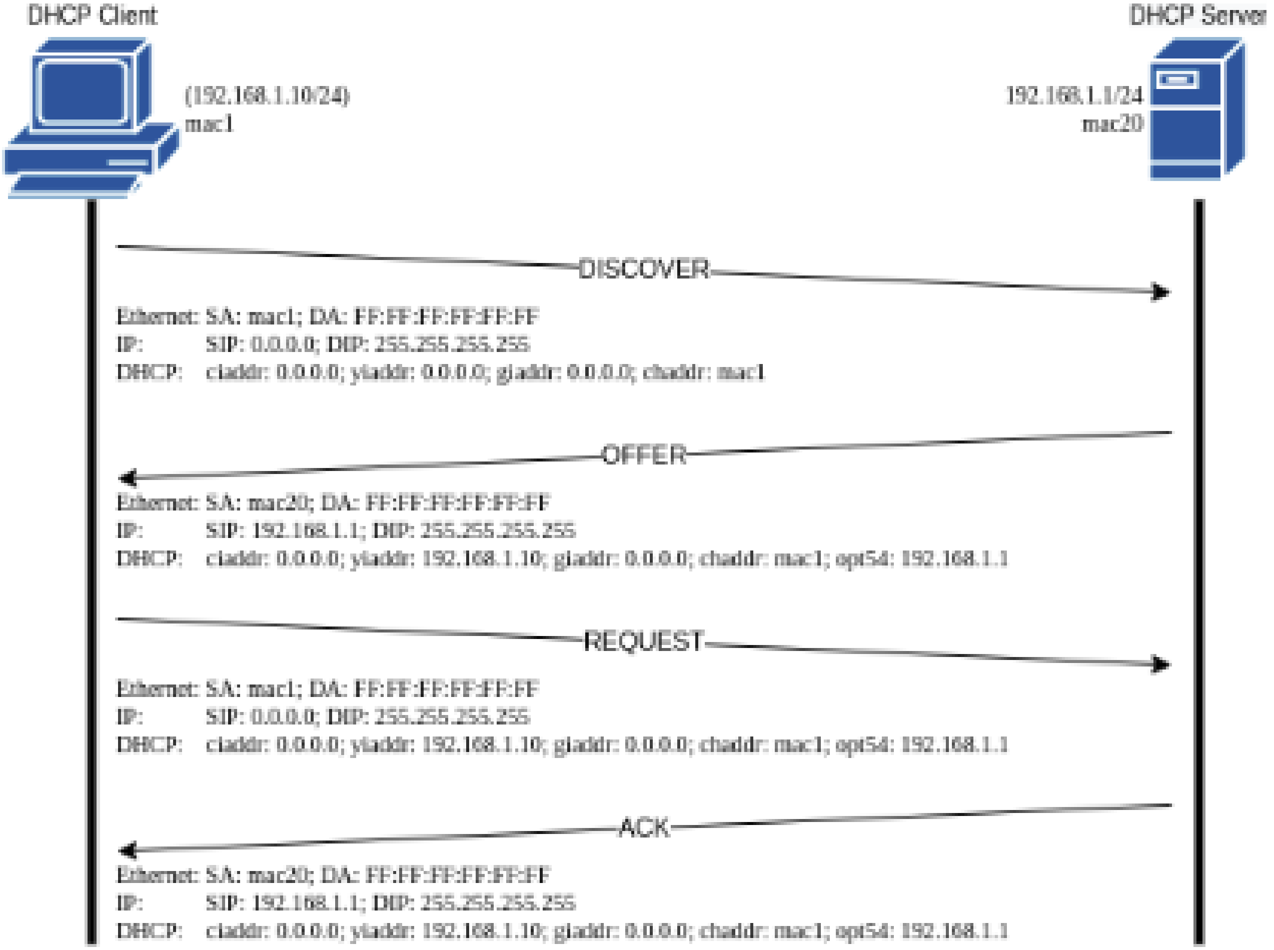
- The Dynamic Host Configuration Protocol was developed to enable dynamic configuration of hosts and is essentially an application of the TCP/IP suite that uses UDP (and therefore IP at the network level) to transmit information
- The protocol works through a client-server interaction
  - the client is the node that performs the network initialization, and broadcasts a query on the network from port 68 (UDP) using local broadcast address (255.255.255.255), which is realized by transmitting a level 2 broadcast frame
  - the DHCP server must be properly configured to respond to requests, providing the necessary parameters (server listens on port 67)

DHCP Lease

- DHCP manages address assignments as a lease
  - When a client requests an address, and this is granted, the assigned address is considered by the server to be no longer usable for other clients for a certain period of time (configurable lease time)
  - At the end of the lease period, the client must renew the request or stop using the assigned address
  - if it renews the request, it is granted to use the same address for a new lease time period
  - When the lease time expires and the client does not renew the request, the server can use the same address for another client
- **it is not guaranteed (unless specified in configuration) that the same client receives the same address in two different and non-consecutive periods**
- the server has a default value and a maximum limit

DHCP messages (DORA process)

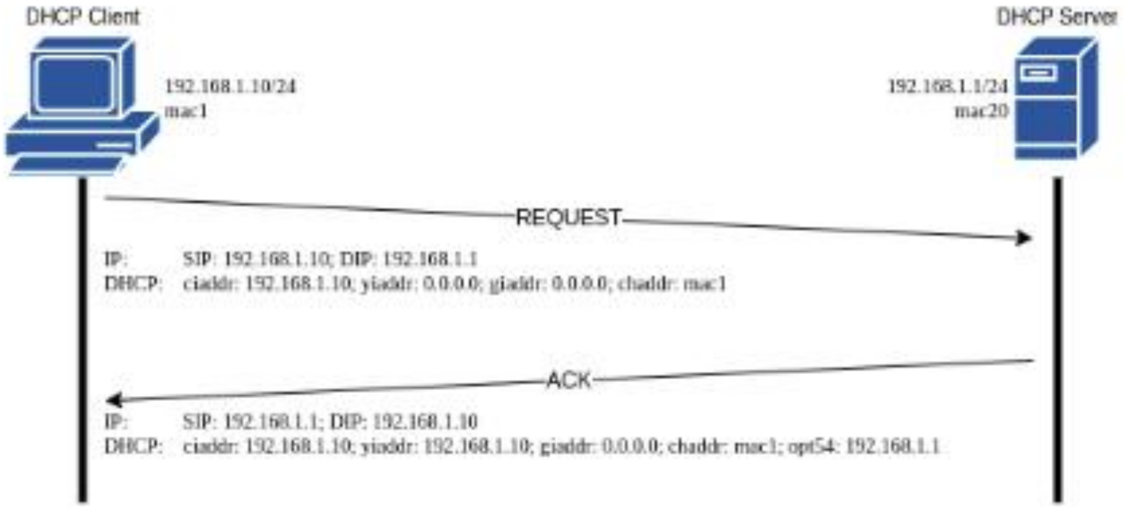
- At initialization the client sends a broadcast message of **DHCPDISCOVER** (configuration request)
- The server (or servers) respond to the request with a **DHCPOFFER** message (offer), containing the configuration information
- The client chooses the first response that arrives, and configures itself in the defined way, but must negotiate the lease parameters: it then sends a **DHCPREQUEST** message to the server from which it accepted the offer
  - this serves to notify the server that offered the address that the client has accepted its offer
  - the servers that do not receive the DHCPREQUEST message consider the offer rejected, and keep the offered address among those available
- The server that receives the DHCPREQUEST message sends a confirmation with a **DHCPACK** message
- from this moment the address is assigned, for the agreed lease time

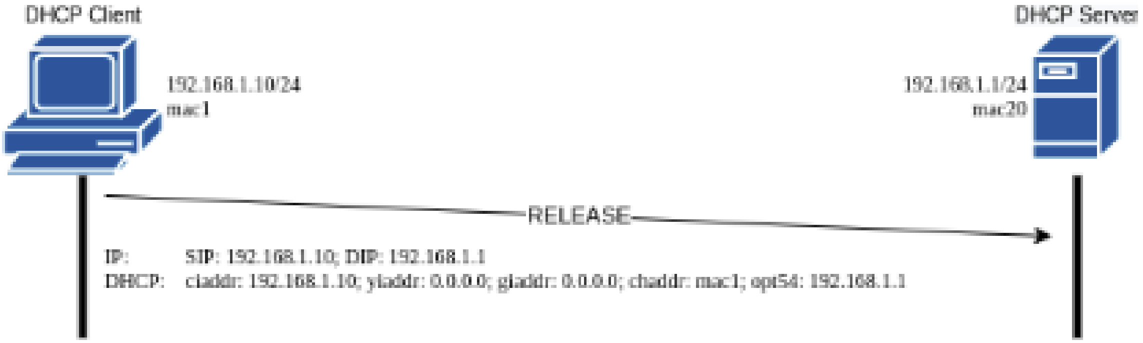


```
Ethernet: SA: mac1; DA: FF:FF:FF:FF:FF:FF
IP:       SIP: 0.0.0.0; DIP: 255.255.255.255
DHCP:     ciaddr: 0.0.0.0; yiaddr: 0.0.0.0; giaddr: 0.0.0.0; chaddr: mac1

Ethernet: SA: mac20; DA: FF:FF:FF:FF:FF:FF
IP:       SIP: 192.168.1.1; DIP: 255.255.255.255
DHCP:     ciaddr: 0.0.0.0; yiaddr: 192.168.1.10; giaddr: 0.0.0.0; chaddr: mac1; opt54: 192.168.1.1

Ethernet: SA: mac1; DA: FF:FF:FF:FF:FF:FF
IP:       SIP: 0.0.0.0; DIP: 255.255.255.255
DHCP:     ciaddr: 0.0.0.0; yiaddr: 192.168.1.10; giaddr: 0.0.0.0; chaddr: mac1; opt54: 192.168.1.1

Ethernet: SA: mac20; DA: FF:FF:FF:FF:FF:FF
IP:       SIP: 192.168.1.1; DIP: 255.255.255.255
DHCP:     ciaddr: 0.0.0.0; yiaddr: 192.168.1.10; giaddr: 0.0.0.0; chaddr: mac1; opt54: 192.168.1.1
```

DHCP Messages (Renew)

- At the expiration of a timer of half the lease time, the client must renew the lease: it sends a new **DHCPREQUEST** message to the server
- The server can:
  - accept the renewal, with a **DHCPACK** message
  - in this case the client resets the timers an continues to use the new proposed configuration (it can have different parameters)
  - refuse the renewal with a **DHCPNACK** message
- The client can:
  - Ask for a renewal, with a **DHCPREQUEST** message
  - Release an ip address, with a **DHCPRELEASE** message


DHCP Client
DHCP Server
192.168.1.10/24
mac1
192.168.1.1/24
mac20
REQUEST
IP: SIP: 192.168.1.10; DIP: 192.168.1.1
DHCP: ciaddr: 192.168.1.10; yiaddr: 0.0.0.0; giaddr: 0.0.0.0; chaddr: mac1
ACK
IP: SIP: 192.168.1.1; DIP: 192.168.1.10
DHCP: ciaddr: 192.168.1.10; yiaddr: 192.168.1.10; giaddr: 0.0.0.0; chaddr: mac1; opt54: 192.168.1.1

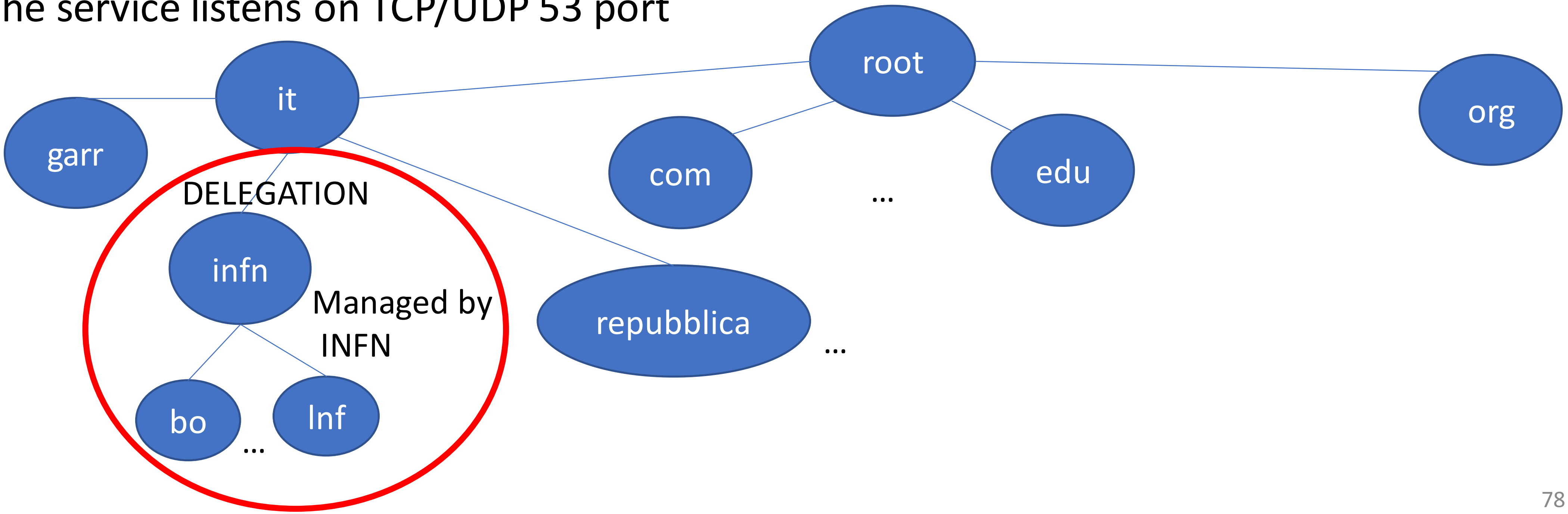

DHCP Client
DHCP Server
192.168.1.10/24
mac1
192.168.1.1/24
mac20
RELEASE
IP: SIP: 192.168.1.10; DIP: 192.168.1.1
DHCP: ciaddr: 192.168.1.10; yiaddr: 0.0.0.0; giaddr: 0.0.0.0; chaddr: mac1; opt54: 192.168.1.1

The "Options" field

- In the DHCP packet there is an options field in which the server can provide additional information.
- In the **OPTIONS** field can be inserted:
  - NTP time server name
  - host name and domain name
  - print server
  - DNS server and search list
  - WINS server
  - TFTP server name for PXE boot
- not all are standard, and not all will be used by the client

Domain Name System

- The **Domain Name System** (**DNS**) is charge of translating IP addresses into human readable hostnames.
- It’s a hierarchical and distributed database naming system for computers, services, and other resources on the Internet.
- Each domain name is essentially just a path in a large inverted tree called domain namespace.
- The service listens on TCP/UDP 53 port



Hierarchy and INFN convention

- A domain name is formed by a sequence of strings dotted separated

www.infn.it

.it is the Top-Level Domain (first level)

.infn is the second level

Eventually more levels 3,4,etc (up to 127 levels)

www is the server's hostname

Every level is child of his previous level

infn.it is usually used for national services otherwise a third level domain is added and refers to an INFN Sites or Laboratory

ftp.mi.infn.it www.lnf.infn.it dsx001.cr.cnaf.infn.it

Domain Name System

- Requesting an IP address from a domain name is called “**DNS resolution**”
  - www.infn.it 193.206.84.44
  - mastercr.cnaf.infn.it → 2001:760:4205:128::128:2

```
$ [nslookup|host|dig] www.infn.it
```

- Requesting a domain name from an IP address is called “**Reverse resolution**”
  - 193.206.84.44 → www.infn.it
  - 2001:760:4205:128::128:2 mastercr.cnaf.infn.it

```
$ [nslookup|host|dig] 193.206.84.44
```

DNS records type

Type	
A AAAA	Name -> IP(v4) IPV6
PTR	IP (IPV6) -> NAME
CNAME	Common Name -> Name aka “alias”
MX	Mail eXchange to define servers responsible to menage mails for a domain
NS	Name Server to define authoritative DNS servers for a domain
SOA	Start Of Autority Specifies *authoritative* information about a DNS zone, including the primary name server, the email of the domain administrator, the domain serial number, and several timers relating to refreshing the zone.
TXT	Used to define arbitrary text for human or machine to implement some specific service
FULL LIST https://en.wikipedia.org/wiki/List_of_DNS_record_types	
RR EXAMPLE: www.infn.it (NAME) 60 (TTL) IN(CLASS) A(TYPE) 193.206.84.44 (RDATA)	

DNS zone examples

```
@       86400   IN      SOA     ns1.dr.infn.it. sysop.dr.infn.it. (
                                2023062701 ; Serial
                                3600       ; Refresh ogni 24 ore
                                600        ; Retry ogni 2 ore
                                604800     ; Expire 30 giorni
                                3600 ; Minimum ttl 4 giorni
                                )
                IN      NS      ns1.dr.infn.it.
                IN      NS      server2.infn.it.
;
$ORIGIN dr.infn.it.
$TTL 86400
;
gw-31                   IN      A       192.135.31.1
ns1                     IN      A       192.135.31.2
```

```
@       84600           IN SOA  ns1.dr.infn.it. sysop.dr.infn.it. (
                                2023062701 ; serial
                                3600       ; refresh (1 day)
                                600        ; retry (2 hours)
                                604800     ; expire (4 weeks 2 days)
                                3600       ; minimum (4 days)
                                )
                IN      NS      ns1.dr.infn.it.
                IN      NS      server2.infn.it.
;
$ORIGIN 31.135.192.in-addr.arpa.
$TTL 86400
;
1       IN      PTR     gw-31.dr.infn.it.
2       IN      PTR     ns1.dr.infn.it.
```

Resolution

DNS servers can act as recursive, not recursive or caching only server

- Iteration (not recursive)

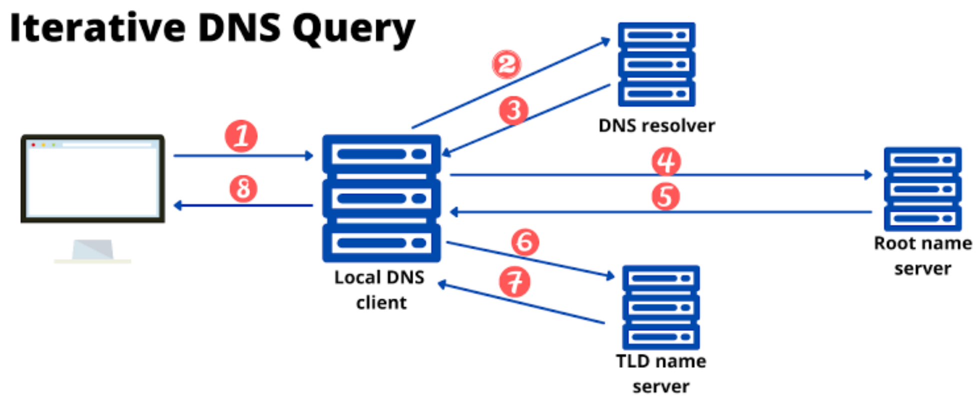
  The DNS server allows the querier to resolve only names that the server directly knows.

- Recursion

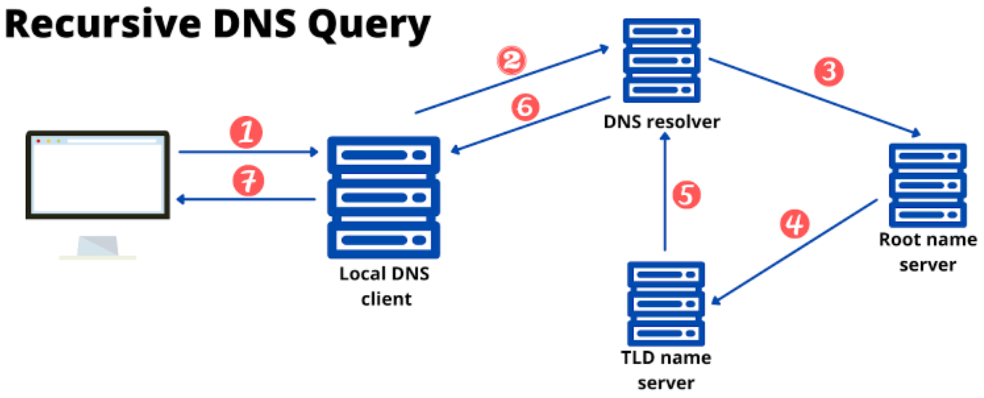
  The DNS server is in charge of resolving names that the server doesn’t directly knows and returns it to the querier.

- Caching nameservers

  DNS that only caches the queries for a specific purpose



Iterative DNS Query



Recursive DNS Query

Views

Sometimes is useful to implement different behaviors depending on the querier.

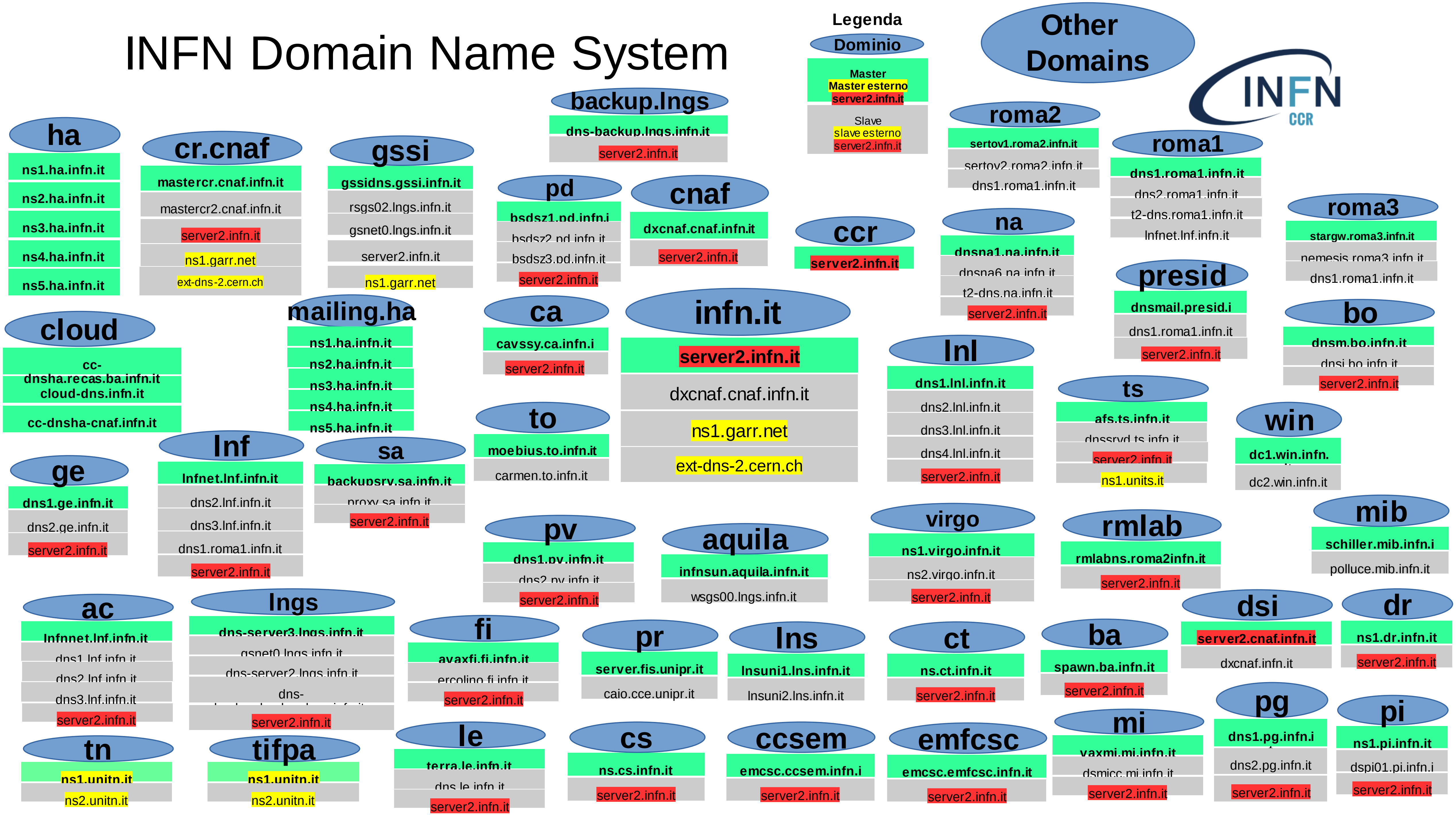
Example:

Query from local networks can be used to resolve everything with recursion and maybe resolve some internal resources accessible only from the internal network (eg printers or private network)

People from internet can only do iterative requests but not resolve some internal resources because of security. (no printer or private network is resolved)

This can be implemented on a single server with different views filtering source/destination with Access Control Lists (ACLs) and pointing to different SOA files.

INFN Domain Name System



Legenda
Dominio
Master
Master esterno
server2.infn.it
Slave
slave esterno
server2.infn.it

Other Domains

ha
ns1.ha.infn.it
ns2.ha.infn.it
ns3.ha.infn.it
ns4.ha.infn.it
ns5.ha.infn.it

cr.cnaf
mastercr.cnaf.infn.it
mastercr2.cnaf.infn.it
server2.infn.it
ns1.garr.net
ext-dns-2.cern.ch

gssi
gssidns.gssi.infn.it
rsgs02.lngs.infn.it
gsnet0.lngs.infn.it
server2.infn.it
ns1.garr.net

backup.lngs
dns-backup.lngs.infn.it
server2.infn.it

pd
bsdsz1.pd.infn.i
bsdsz2.pd.infn.it
bsdsz3.pd.infn.it
server2.infn.it

cnaf
dxcnaf.cnaf.infn.it
server2.infn.it

ccr
server2.infn.it

roma2
sertov1.roma2.infn.it
sertov2.roma2.infn.it
dns1.roma1.infn.it

roma1
dns1.roma1.infn.it
dns2.roma1.infn.it
t2-dns.roma1.infn.it
lnfnet.lnf.infn.it

na
dnsna1.na.infn.it
dnsna6.na.infn.it
t2-dns.na.infn.it
server2.infn.it

roma3
stargw.roma3.infn.it
nemesis.roma3.infn.it
dns1.roma1.infn.it

presid
dnsmail.presid.i
dns1.roma1.infn.it
server2.infn.it

cloud
cc-dnsha.recas.ba.infn.it
cloud-dns.infn.it
cc-dnsha-cnaf.infn.it

mailing.ha
ns1.ha.infn.it
ns2.ha.infn.it
ns3.ha.infn.it
ns4.ha.infn.it
ns5.ha.infn.it

ca
cavssy.ca.infn.i
server2.infn.it

infn.it
server2.infn.it
dxcnaf.cnaf.infn.it
ns1.garr.net
ext-dns-2.cern.ch

lnl
dns1.lnl.infn.it
dns2.lnl.infn.it
dns3.lnl.infn.it
dns4.lnl.infn.it
server2.infn.it

bo
dnsm.bo.infn.it
dnsi.bo.infn.it
server2.infn.it

ts
afs.ts.infn.it
dnssrvd.ts.infn.it
server2.infn.it
ns1.units.it

win
dc1.win.infn.
dc2.win.infn.it

to
moebius.to.infn.it
carmen.to.infn.it

lnf
lnfnet.lnf.infn.it
dns2.lnf.infn.it
dns3.lnf.infn.it
dns1.roma1.infn.it
server2.infn.it

sa
backupsrv.sa.infn.it
proxy.sa.infn.it
server2.infn.it

ge
dns1.ge.infn.it
dns2.ge.infn.it
server2.infn.it

pv
dns1.pv.infn.it
dns2.pv.infn.it
server2.infn.it

aquila
infnsun.aquila.infn.it
wsgs00.lngs.infn.it

virgo
ns1.virgo.infn.it
ns2.virgo.infn.it
server2.infn.it

rmlab
rmlabns.roma2infn.it
server2.infn.it

mib
schiller.mib.infn.i
polluce.mib.infn.it

ac
lnfnet.lnf.infn.it
dns1.lnf.infn.it
dns2.lnf.infn.it
dns3.lnf.infn.it
server2.infn.it

lngs
dns-server3.lngs.infn.it
gsnet0.lngs.infn.it
dns-server2.lngs.infn.it
dns-
server2.infn.it

fi
avaxfi.fi.infn.it
ercolino.fi.infn.it
server2.infn.it

pr
server.fis.unipr.it
caio.cce.unipr.it

lns
lnsuni1.lns.infn.it
lnsuni2.lns.infn.it

ct
ns.ct.infn.it
server2.infn.it

ba
spawn.ba.infn.it
server2.infn.it

dsi
server2.cnaf.infn.it
dxcnaf.infn.it

dr
ns1.dr.infn.it
server2.infn.it

pg
dns1.pg.infn.i
dns2.pg.infn.it
server2.infn.it

pi
ns1.pi.infn.it
dspi01.pi.infn.it
server2.infn.it

mi
vaxmi.mi.infn.it
dsmicc.mi.infn.it
server2.infn.it

tn
ns1.unitn.it
ns2.unitn.it

tifpa
ns1.unitn.it
ns2.unitn.it

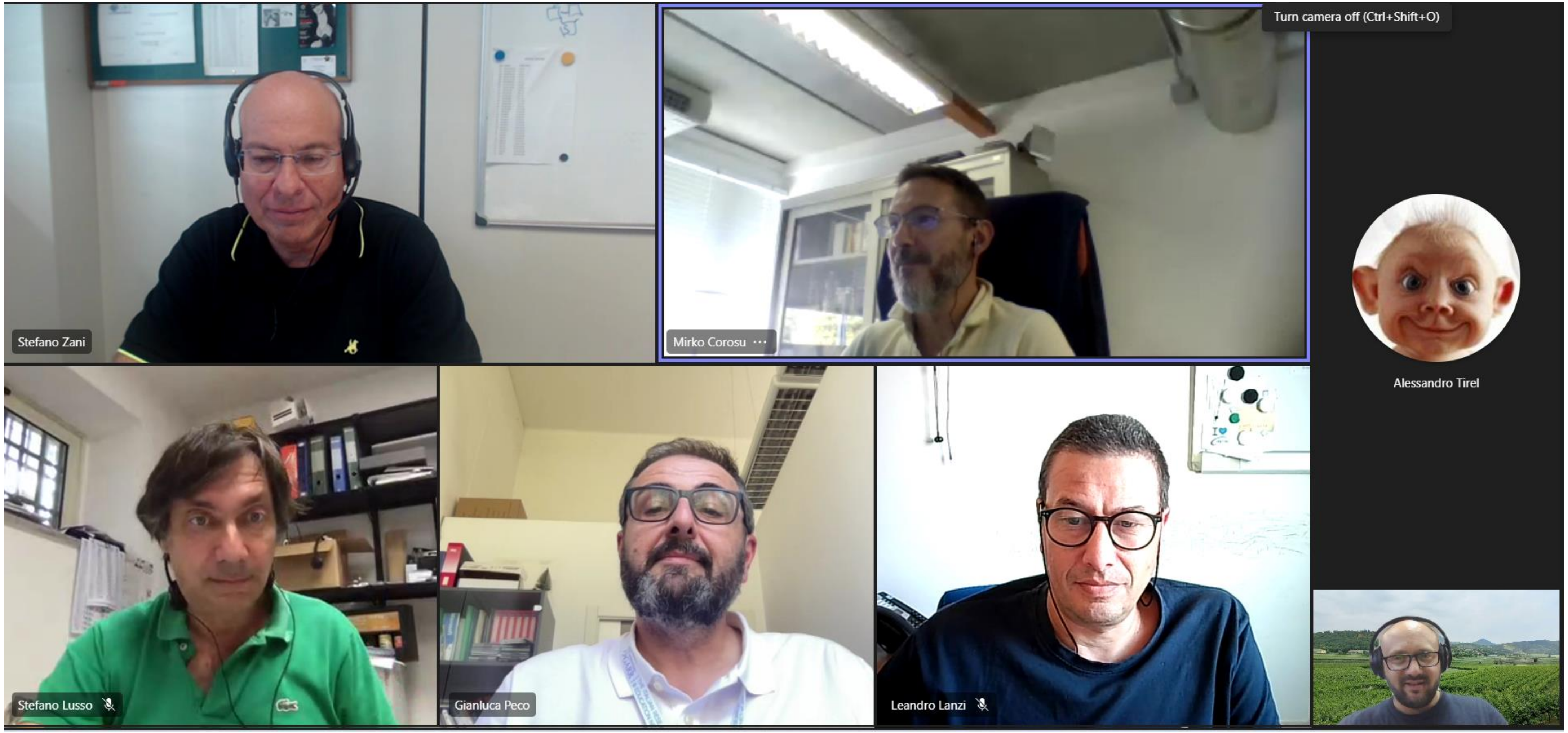
le
terra.le.infn.it
dns.le.infn.it
server2.infn.it

cs
ns.cs.infn.it
server2.infn.it

ccsem
emcsc.ccsem.infn.i
server2.infn.it

emfcsc
emcsc.emfcsc.infn.it
server2.infn.it

NETGROUP


Turn camera off (Ctrl+Shift+O)
Stefano Zani
Mirko Corosu
Alessandro Tirel
Stefano Lusso
Gianluca Peco
Leandro Lanzi