

Computer Networks

3. Network layer

Boni García

<http://bonigarcia.github.io/>

boni.garcia@urjc.es

Departamento de Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación
Escuela Técnica Superior de Ingeniería de Telecomunicación
Universidad Rey Juan Carlos

2019/2020

Table of contents

1. Introduction
2. IPv4
3. IPv6
4. Routing in Internet
5. Takeaways

Table of contents

1. Introduction

I. Routers

II. Network service

2. IPv4

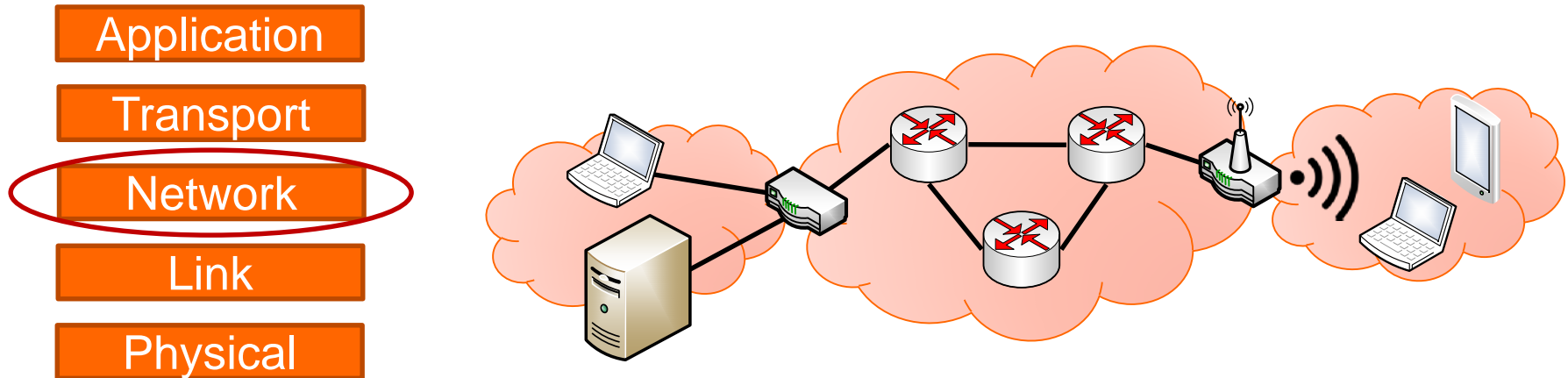
3. IPv6

4. Routing in Internet

5. Takeaways

1. Introduction

- Network layer in the TCP/IP model (i.e. Internet) is responsible for the communication between hosts in the whole **network** (not only in the same physical network)
- The devices which interconnect different physical networks (segments) are called **routers**



TCP/IP model

1. Introduction

- The name of the Protocol Data Unit (PDU) at network layer is called **packet**
- In the TCP/IP model (i.e. Internet) the protocol for the network layer is always **IP** (Internet Protocol)
 - There are two version of IP: IPv4 and IPv6
- The main objective of the network layer (i.e. the service provided for the upper layer, i.e. the transport) is the packet **routing** (i.e. find a path for packets from a source hosts to a destination)
 - Packets are going to traverse a number of intermediate routers
 - Routers use a forwarding table to find the path for packets
 - There are different routing protocols aimed to set and update the forwarding table of routers

1. Introduction - Routers

- Routers are devices which forwards packets in computer networks
- Routers work at network level (3rd layer in the TCP/IP model)
- Routers have 2 or more network interfaces
 - Home routers have two interfaces: one to the LAN (Local Area Network) and the other to the ISP (Internet Service Provider) network. These kind routers are usually called gateways
- Routers implement storage and forwarding features:
 - When a packet arrives at a router, it is stored in the router internal memory (buffer) and its destination address is examined
 - Each router has a forwarding table that assigns the destination addresses (or a part of them) to the outgoing links
 - There are a number of routing protocols that are used to automatically define the forwarding tables

1. Introduction - Network service

- There are different kinds of **services** offered by protocols:
 1. Depending on the connection type:
 - Connection-oriented: Before sending data, an initial handshaking between entities is required to setup a connection between them
 - Connectionless: There is no any handshaking between entities before sending data
 2. Depending on how the packets are routed:
 - Datagram mode: Target address is sent in each PDU and routing is independent (there could be different paths for the same target)
 - Virtual circuit: At the beginning a fixed path called virtual circuit is established
 3. Depending on the reliability:
 - Reliable: The delivery and order of sent PDUs is warranted. To that aim, lost PDUs needs to be retransmitted
 - Not reliable (*best effort*): The delivery and order of sent PDUs is not warranted

1. Introduction - Network service

- All combinations of these features (connection-oriented vs connectionless, datagram mode vs virtual-circuit, reliable vs non reliable) are possible, although in practice there are some likely combinations
- At network level, the usual network service is:
 - Connectionless, datagram mode, non reliable: For example **IP**
 - Connection-oriented, virtual-circuit, reliable: For example X.25

Table of contents

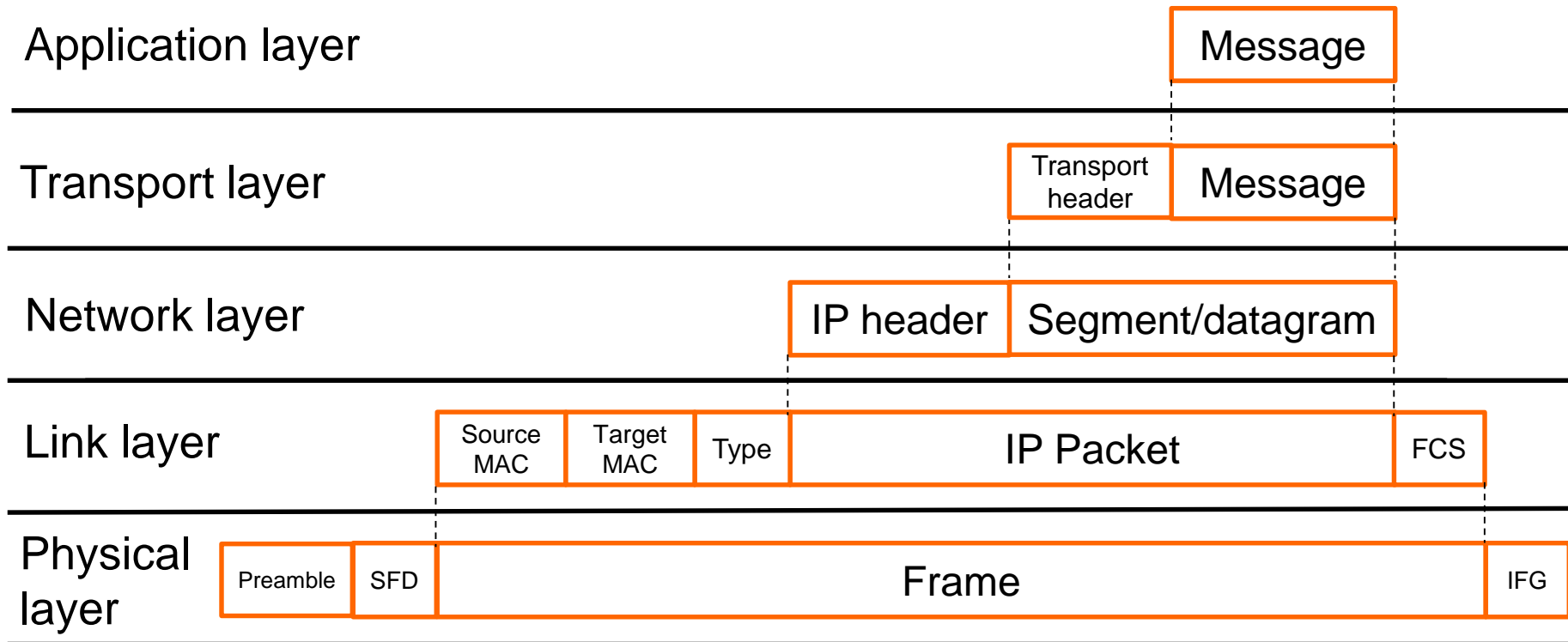
1. Introduction
2. IPv4
 - I. Introduction
 - II. Packet format
 - III. Addresses
 - a. Classful
 - b. Special addresses
 - c. Subnetting
 - d. Classless
 - IV. NAT
 - V. Routing table
 - VI. ARP
 - VII. DHCP
 - VIII. IP configuration in hosts
 - IX. ICMP
3. IPv6
4. Routing in Internet
5. Takeaways

2. IPv4 - Introduction

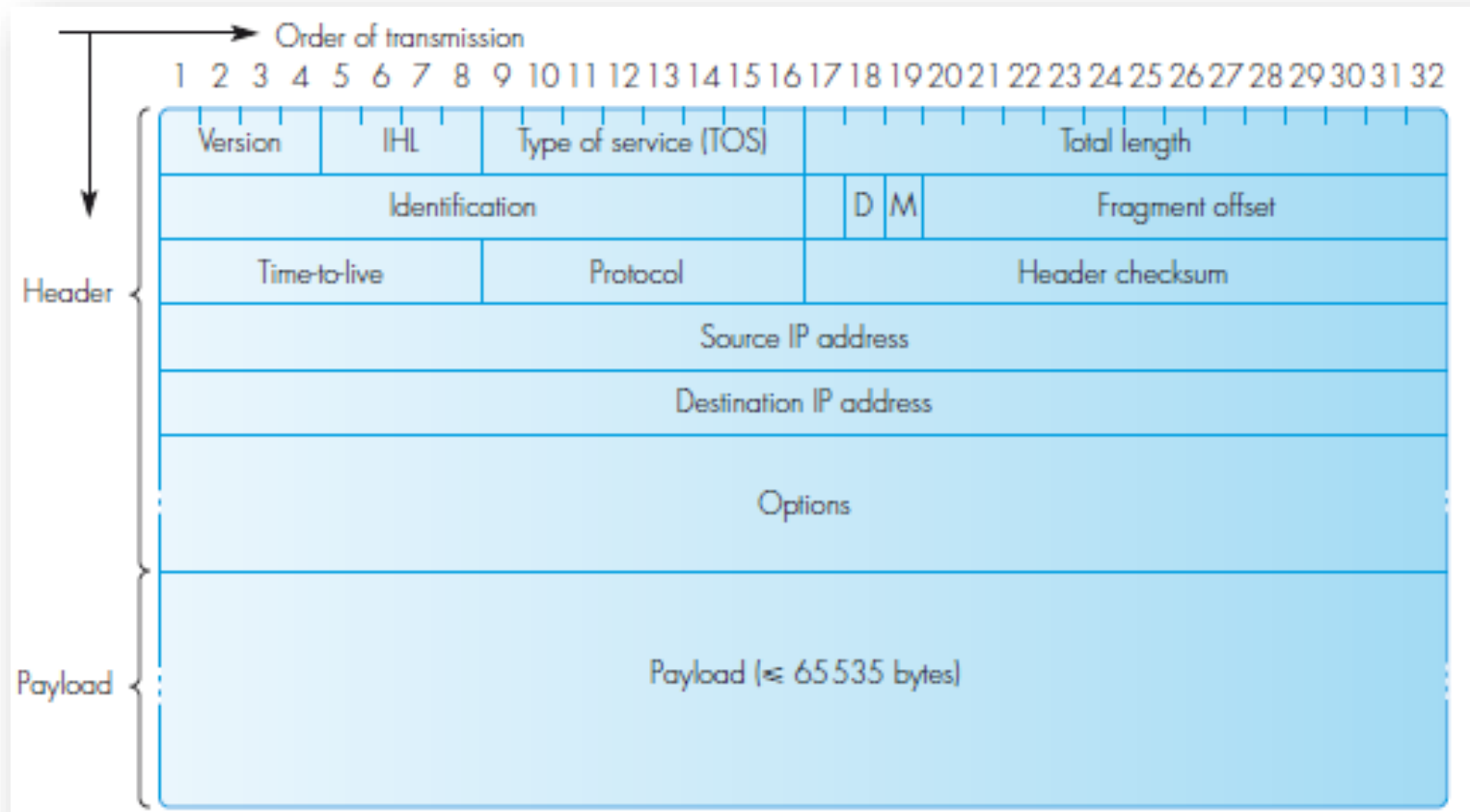
- IP (Internet Protocol) is the network layer protocol of the TCP/IP model (Internet)
- It has been defined in the [RFC 791](#) (Request for Comments) by the IETF (Internet Engineering Task Force)
- The main features of IP are:
 - Connectionless protocol: no initial handshake
 - Datagram mode: each packet is routed independently
 - Non reliable: No Quality of Service (QoS) by default (best effort service)

2. IPv4 - Introduction

- The encapsulation in TCP/IP is as follows:




2. IPv4 - Packet format



2. IPv4 - Packet format

- Version: 4 (IPv4)
- IHL (Internet Header Length): length of the IP header in units of 4 bytes
 - Default value = 5, i.e. without options
- TOS (Type of Service):
 - Priority (3 bits): 0-7
 - Delay (1 = low delay required)
 - Throughput (1 = high bit rate)
 - Reliability (1 = high reliability)
 - Cost (1 = low cost)



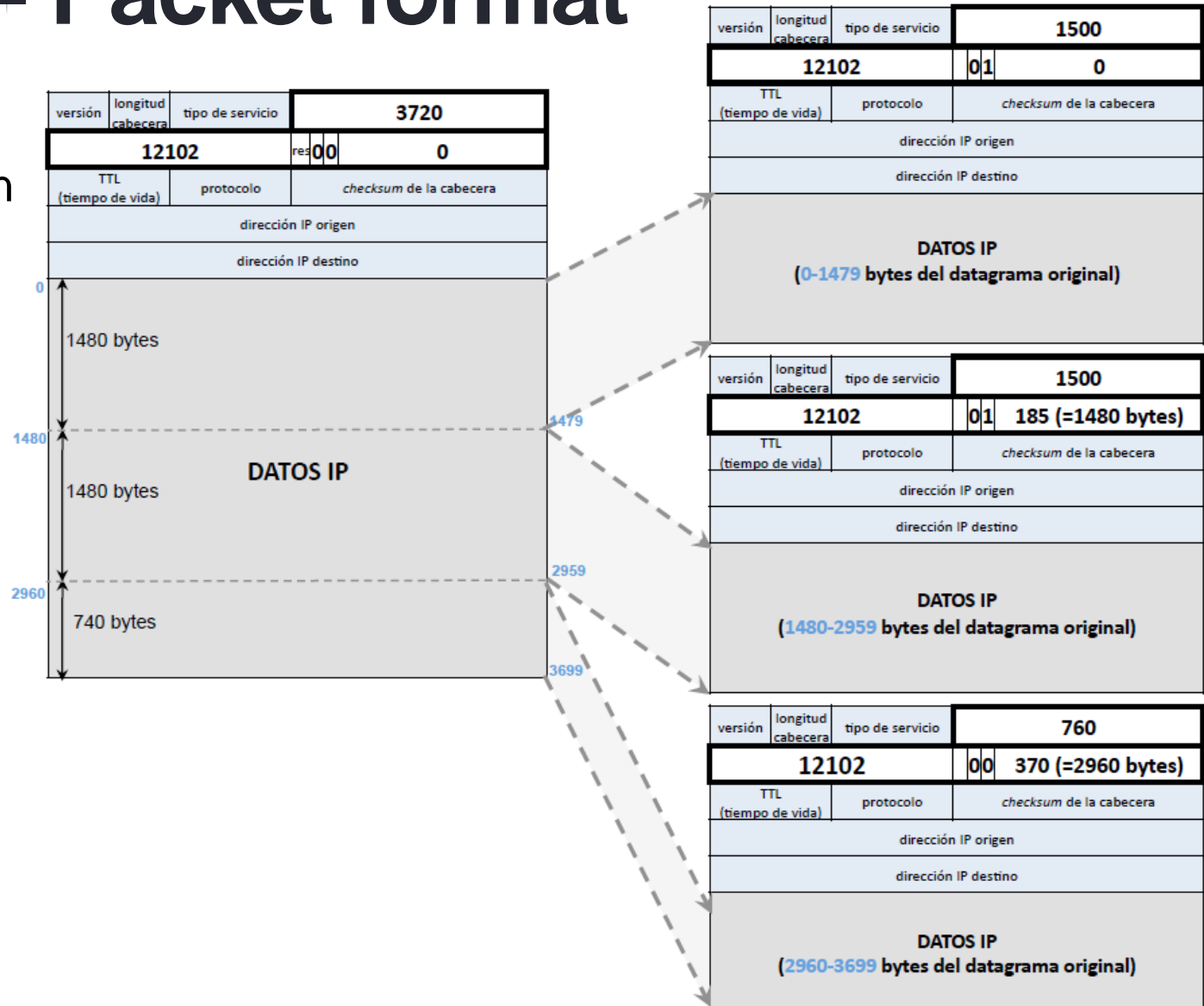
In practice, these flags are not implemented in Internet routers. [RFC 2474](#) redefines these flags in the technique called DiffServ (Differentiated Services), which aims to provide quality of service (QoS) in IPv4 networks

2. IPv4 - Packet format

- Total length of the packet (header + data) in bytes
 - Theoretical maximum length of an packet = $2^{16} - 1 = 65,535$ bytes
 - Real maximum length of an IP packet = link layer **MTU** (1500 bytes in Ethernet)
 - If IP packet length > MTU → packet is fragmented
- Identification: unique value for each of the fragments of the same message
- Flags:
 - DF = Do not fragment (1 = data cannot be fragmented)
 - MF = More fragment (1 = intermediate fragment, there is more, 0 = last fragment)
- Offset: Pointer with respect to origin for fragments in 8 bytes words. Its value is 0 when there are no fragments

2. IPv4 - Packet format

- Example of fragmentation in IPv4:



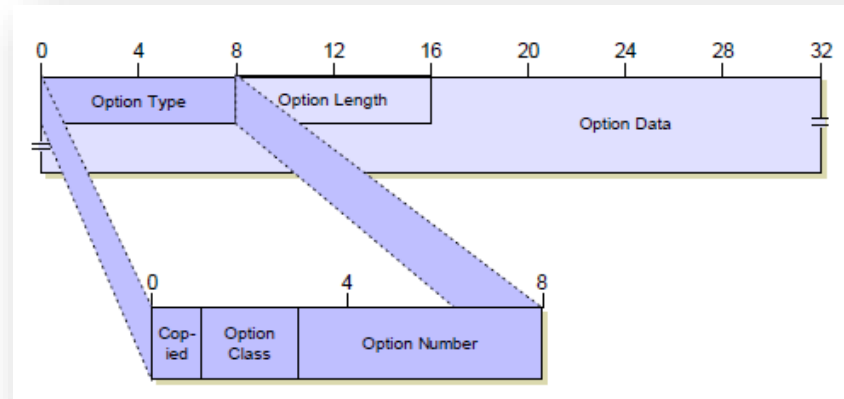
2. IPv4 - Packet format

- TTL (Time To Live). It is the packet life time.
 - Source host gives an initial value to the TTL of each packet.
 - Each router decreases it when packet traverse it.
 - If TTL reaches 0, the package is discarded
- Transport protocol. [RFC 1700](#) defines the possible values:

| Value (Hexadecimal) | Value (Decimal) | Protocol |
|------------------------|--------------------|--|
| 00 | 0 | Reserved |
| 01 | 1 | ICMP |
| 02 | 2 | IGMP |
| 03 | 3 | GGP |
| 04 | 4 | IP-in-IP Encapsulation |
| 06 | 6 | TCP |
| 08 | 8 | EGP |
| 11 | 17 | UDP |
| 32 | 50 | Encapsulating Security Payload (ESP) Extension Header |
| 33 | 51 | Authentication Header (AH) Extension Header |

2. IPv4 - Packet format

- Header checksum: Sum (in 1's complement) of **header only** in words of 16 bits
 - Calculated by source host
 - When a packet arrives at a router, the router calculates the checksum of the header and compares it to the checksum field
 - If the values do not match, the router (or destination) discards the packet
- Options: extra features of IPv4:



- Padding: "1's" if options are not multiple of 32 bits

2. IPv4 - Addresses

- IPv4 addresses have a length of 32 bits
- This figure gives a total of 2^{32} available IP addresses, i.e. almost 4.3 billion (10^9)
- Each IP address must be unique within the same network
- The decimal notation with points is used to write IPv4 addresses, e.g.:

172 . 16 . 254 . 1

Dot-decimal notation: divide the 32 bits of an IPv4 address into 4 parts and represent each part (8 bits) in decimal separated with dots

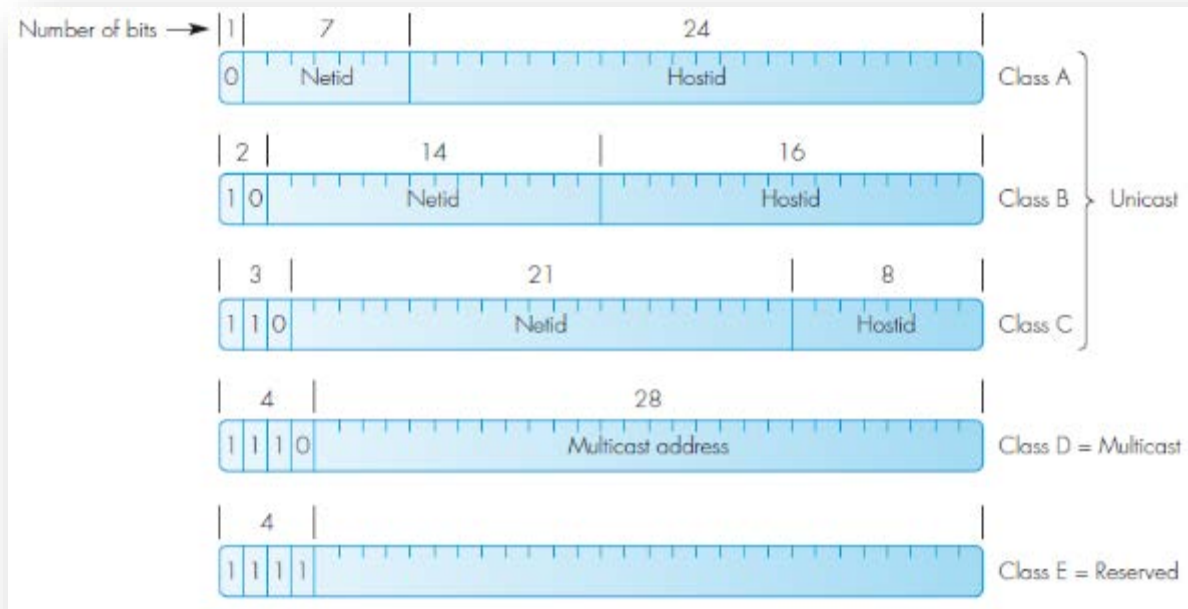
- IANA (Internet Assigned Numbers Authority) is the organization in charge of managing and assigning IP addresses
- IPv4 addresses were officially finished on February 3, 2011

2. IPv4 - Addresses

- There are 5 types of IP addressing schemes that have been used in the life of the Internet:
 1. Classful addresses: First schema of IPv4 addresses
 2. Subnets: Division of host identifier in subnet + host
 3. Classless addresses: Removal of the network boundary of the classes
 4. NAT (Network Address Translation)
 5. IPv6: Next version of IP using longer addresses (128 bits)

2. IPv4 - Addresses - Classful

- IPv4 addresses always have two parts: network identifier (*netid*) and host identifier (*hostid*)
- In the classful schema, each class determines the network size



2. IPv4 - Addresses - Classful

- Class A:
 - The first bit has the value at '0' and the other 7 bits are used to identify the network (*netld*)
 - The other 24 bits (3 bytes) are used to identify the machine (*hostld*)
 - **Large networks**
- Class B:
 - The first two bits have the value '10' and together with the next 14 bits, they are used to identify the network (*netld*) [16 bits of *netld*]
 - The other 16 bits (2 bytes) are used to identify the machine (*hostld*)
 - **Medium-sized networks**
- Class C:
 - The first 3 bits of the address have the value '110', which together with other 21 bits, identify the network (*netld*) [24 bits of *netld*]
 - The other 8 bits are used to identify the host (*hostld*)
 - **Small networks**

Note: Nowadays the use of A, B, C classes is deprecated

2. IPv4 - Addresses - Classful

- Class D:
 - The first 4 bits of the address have the value '1110'
 - These addresses are called **multicast** (i.e. an IP address which represent a group of hosts in the same network)
 - Not confuse with broadcast (i.e. all hosts in the network)
- Class E:
 - The first 4 bits of the address have the value '1111'
 - These are addresses reserved for **experimental use** (268 million IP, which in practice are not used)

2. IPv4 - Addresses - Special addresses


- Special IPv4 addresses:
 - **Loopback** address: Class addresses to 127.x.x.x are reserved for the internal loopback interface. It is usually used **127.0.0.1** (*localhost*) to test communications between processes on the same machine, where it identifies the own host address
 - **0.0.0.0** address: it means “any IPv4 address at all”
 - As host address, if a server listens to 0.0.0.0, requests are heard on all network interfaces
 - In the context of routing, it is usually used as the default route in routing tables

2. IPv4 - Addresses - Special addresses

- Special IPv4 addresses:
 - **Broadcast** address. All bits of the host identifier (*hostId*) to 1. It means “all hosts in this network”. It is used to send an IP packet to all hosts in that network (for UDP traffic usually). If the network identifier (*netId*) is not known, the broadcast address 255.255.255.255 is used, which makes sense for the local network
 - **Netmask**. Indicates that part of the IP address has all the hosts of that subnet in common. The network bits (*netId*) are at '1' and the host bits (*hostId*) are at 0. It is used to find out the size of the network and the specific broadcast address
 - **Network prefix**: All bits to 0 in the host identifier (*hostId*). Indicates the direction of the network itself

network prefix = IP address & netmask

Boolean
operator AND



2. IPv4 - Addresses - Special addresses

- In Unix-like systems (e.g. Linux, Mac OS) we can check the network configuration using the command `ifconfig` (equivalent to `ipconfig` in Windows systems)

```
bgarcia@a-a1105-pc01:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 212.128.253.65 netmask 255.255.255.0 broadcast 212.128.253.255
    ether e4:b9:7a:f8:0d:93 txqueuelen 1000 (Ethernet)
    RX packets 158798 bytes 11973787 (11.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 29983 bytes 3860526 (3.8 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 20 memory 0x91000000-91020000
```

Network configuration
(for a given NIC,
Ethernet in this
example)

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Bucle local)
    RX packets 3513 bytes 545882 (545.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3513 bytes 545882 (545.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Loopback device
(virtual NIC), used in
Unix-like to connect to
itself (usually for
testing purposes)

2. IPv4 - Addresses - Subnetting

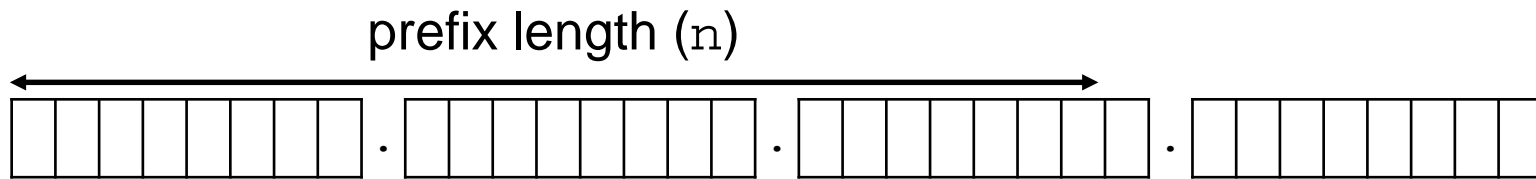
- The classful addressing has a clear limitation: the IPv4 addresses are managed inefficiently
- To overcome this problem, the **subnets** were created
- In subnet addressing, the address assignment is optimized in order to divide a range of addresses A, B, or C into different subnets. To do this, the *hostId* is divided into two parts:
 - A part that identifies the subnet: *subnetId*
 - Another part that identifies the host address: *hostId*



- In this scheme, a **subnet mask** is the special address in which *netId+subnetId* have all their bits to 1 and all bits of *hostId* are 0
- The first subnets schema had an important limitation: the size of the subnet mask must be multiple of 8 bits (8, 16, 24 bits)

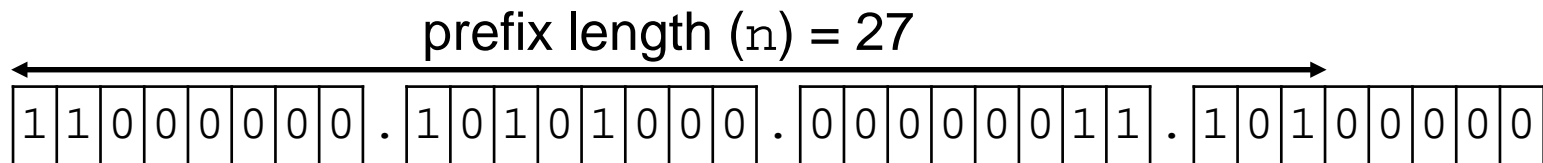
2. IPv4 - Addresses - Classless

- For a more efficient management of the IP addresses (better use of the ranges for the subnets), CIDR (Classless Inter-Domain Routing) was proposed in [RFC 1519](#)
- CIDR use variable length network masks (VLSM, Variable-Length Subnet Masking) and not only multiple of 8 bits
- This schema makes the addresses based on classes A, B, C obsolete
- The CIDR notation to identify a network is as follows: **w.x.y.z/n**, where:
 - w.x.y.z = network prefix
 - n = prefix length (number of bits for the network mask)

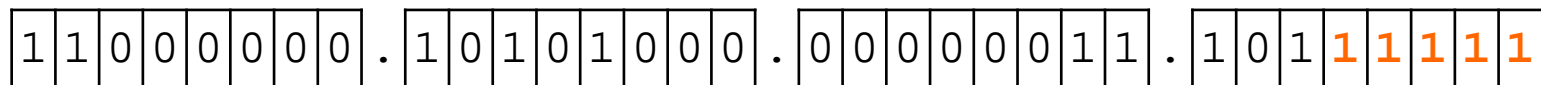


2. IPv4 - Addresses - Classless

- Example 1: What is the **broadcast address** of the network 192.168.3.160/27?



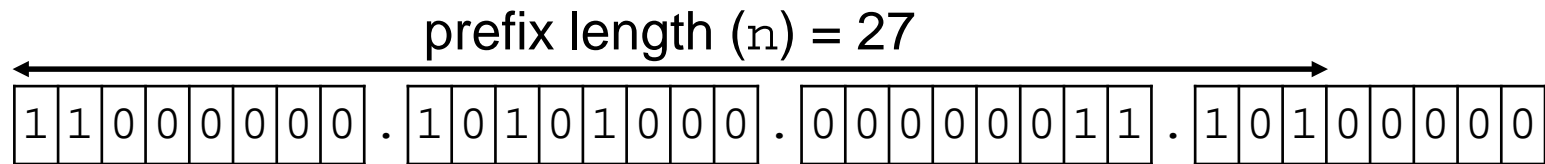
- Broadcast address (hostId = 1):



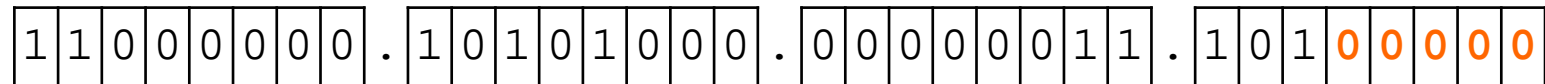
- Answer: 192.168.3.191
- Example of online service to carry out calculations about IPv4 addressing: <http://www.calculadora-redes.com/>

2. IPv4 - Addresses - Classless

- Example 2: What is the **network prefix** of the network 192.168.3.160/27?



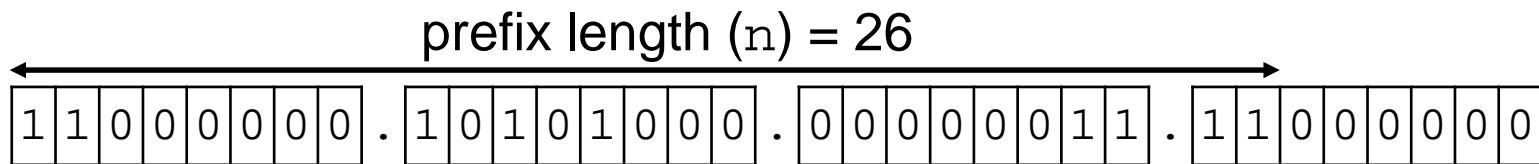
- Network address (hostId = 0):



- Answer: 192.168.3.160

2. IPv4 - Addresses - Classless

- Example 3: What is the **netmask** of the network 192.168.3.192/26?



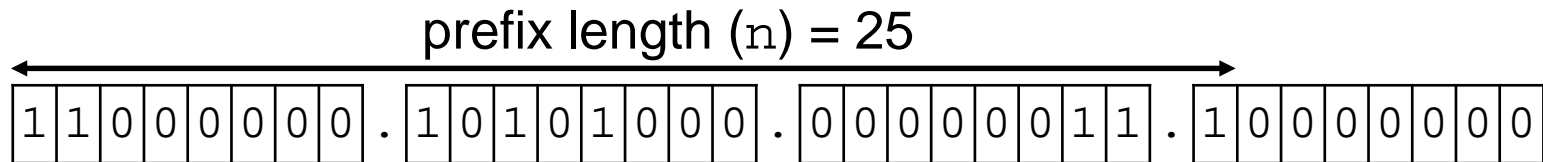
- Netmask (netId = 1; hostId = 0):



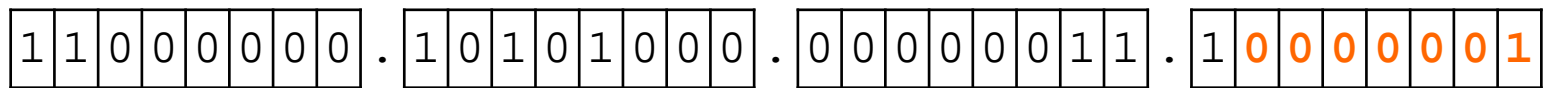
- Answer: 255.255.255.192

2. IPv4 - Addresses - Classless

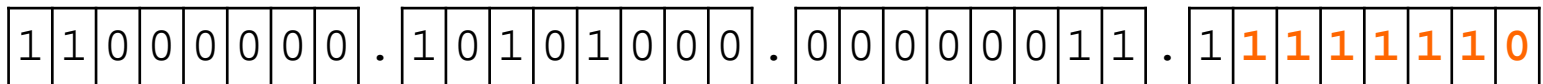
- Example 4: What is the **first and last IP addresses** of the network 192.168.3.192/25?



- First IP address (hostId = 000 ... 1):



- Last IP address (hostId = 111 ... 0):



- Answer: 192.168.3.129 and 192.168.3.254 (there are 126 possible IP addresses in this range, this is $2^7 - 2$)

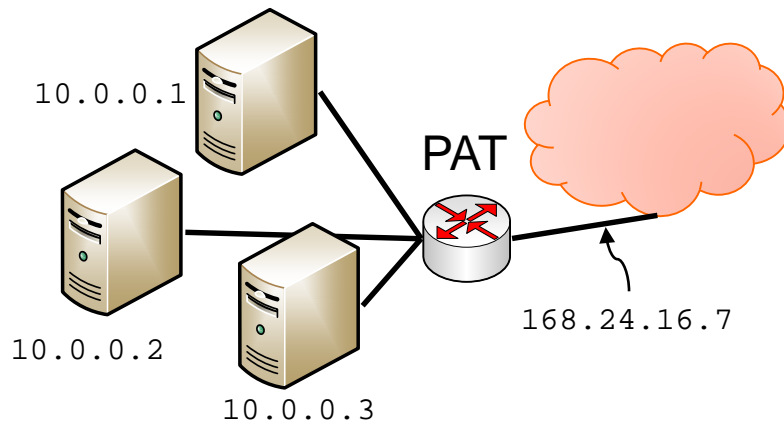
2. IPv4 - Addresses - NAT

- **NAT** (Network Address Translation) is a method which allows to assign one (or several) IP address(es) to a set of **private IP addresses**
 - NAT allows to preserve public IPv4 address space to solve the problem of IPv4 address exhaustion
 - NAT has been defined on RFCs [2663](#) and [3022](#)
- The private IP addresses ranges are well known ([RFC 1918](#))

| HostId | IP addresses range | Network |
|---------|-------------------------------|----------------|
| 24 bits | 10.0.0.0 – 10.255.255.255 | 10.0.0.0/8 |
| 20 bits | 172.16.0.0 – 172.31.255.255 | 172.16.0.0/12 |
| 16 bits | 192.168.0.0 – 192.168.255.255 | 192.168.0.0/16 |
| 16 bits | 169.254.0.0 – 169.254.255.255 | 169.254.0.0/16 |

2. IPv4 - Addresses - NAT

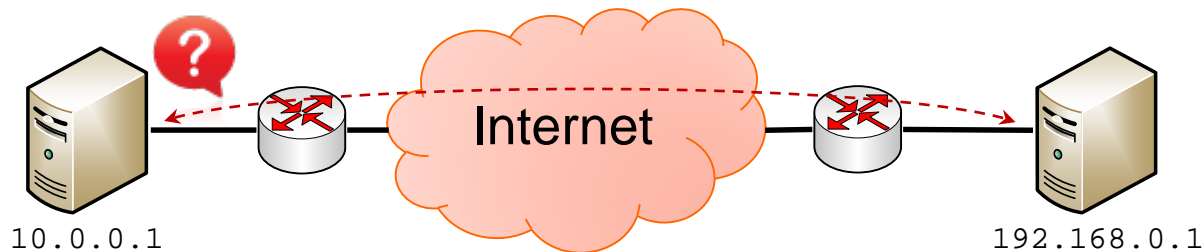
- The most common case is when a NAT device use a **single public IP** (this typically happens in home gateways). This is known as static PAT (Port Address Translation)
- PAT devices maintains a **NAT table** with information about internal connections (IP and port) and external connections
- External hosts to the network will be able to "traverse the NAT" only if the internal host has previously sent information



| Internal IP | Internal port | External IP | External port |
|-------------|---------------|-------------|---------------|
| 10.0.0.3 | TCP 53750 | 168.24.16.7 | TCP 4001 |
| ... | ... | ... | ... |

2. IPv4 - Addresses - NAT

- Address translation performed on NAT devices works well for client-server applications where the client initiates communication and the server is well known
- On the other hand, P2P (peer to peer) communication between hosts that are behind NAT devices can be complicated
- This occurs in file sharing applications, video conferencing systems, online games, etc.



2. IPv4 - Addresses - NAT

- There are two main ways to resolve the problem of host communication behind NAT devices:
 1. Manually, through **port forwarding** (port mapping)
 - Port forwarding is carried out by manually adding rules in the NAT table, so that the output ports of the NAT are the same as the host of the private network
 - The effect will be that the port is visible ("open") from outside the private network (as long as there are no Firewalls that introduce other traffic rules)
 2. Automatically, using **NAT traversal** techniques (hole punching, TURN, STUN, ...)
- There are different tools which can help in this domain:
 - What is my IP address? <https://www.whatismyip.com/>
 - Is a port open? <http://www.yougetsignal.com/tools/open-ports/>
 - How to open the ports of my router? <https://portforward.com/>
 - What type of NAT is in my network? <https://pypi.python.org/pypi/pystun>

2. IPv4 - Addresses - Routing table

- A host requires a proper network configuration to communicate with other using its network interface
- The typical network setup includes:
 - **IP address**: used as a unique identifier of my host in the whole network
 - **Netmask**: used to identify the hosts in my subnet (broadcast domain at network level) from the rest of hosts
 - Default gateway: used to communicate with other hosts outside my physical network
 - DNS servers: used to resolve domain names to IP addresses
- This setup can be:
 - Static (i.e. manual)
 - Dynamic (i.e. using DHCP)

2. IPv4 - Addresses - Routing table

- Each network device (i.e. hosts and routers) has a **routing table** which contains a list of routes to particular destinations (i.e. networks or hosts)
- Routing tables typically contains the following information:
 - **Destination**: Target host or network prefix
 - **Netmask**: Netmask of the target host or network
 - **Gateway**: Next hop
 - **Interface**: Name of interface for the route
 - **Metric**: Associated cost of using the route
 - **Flags**: U=Up (route is valid), G=Gateway (route is done through a Gateway instead of a direct connection)

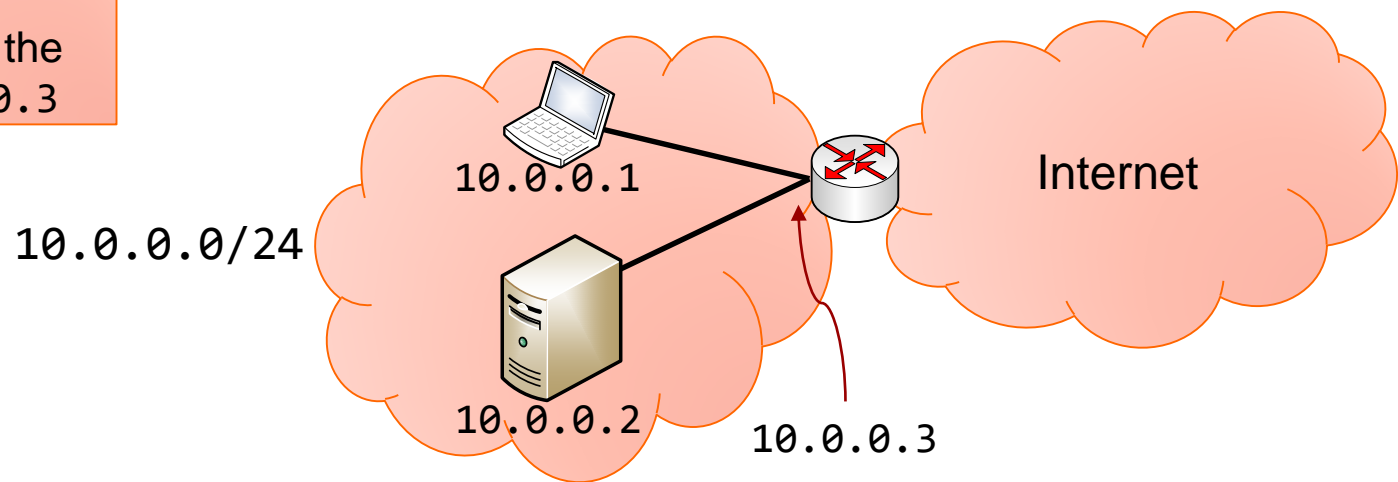
2. IPv4 - Addresses - Routing table

- For example:

| Destination | Netmask | Gateway | Interface |
|-------------|---------------|----------|-----------|
| 10.0.0.0 | 255.255.255.0 | 0.0.0.0 | eth0 |
| 0.0.0.0 | 0.0.0.0 | 10.0.0.3 | eth0 |

This entry means that gateway is not required to reach network 10.0.0.0/24 through eth0

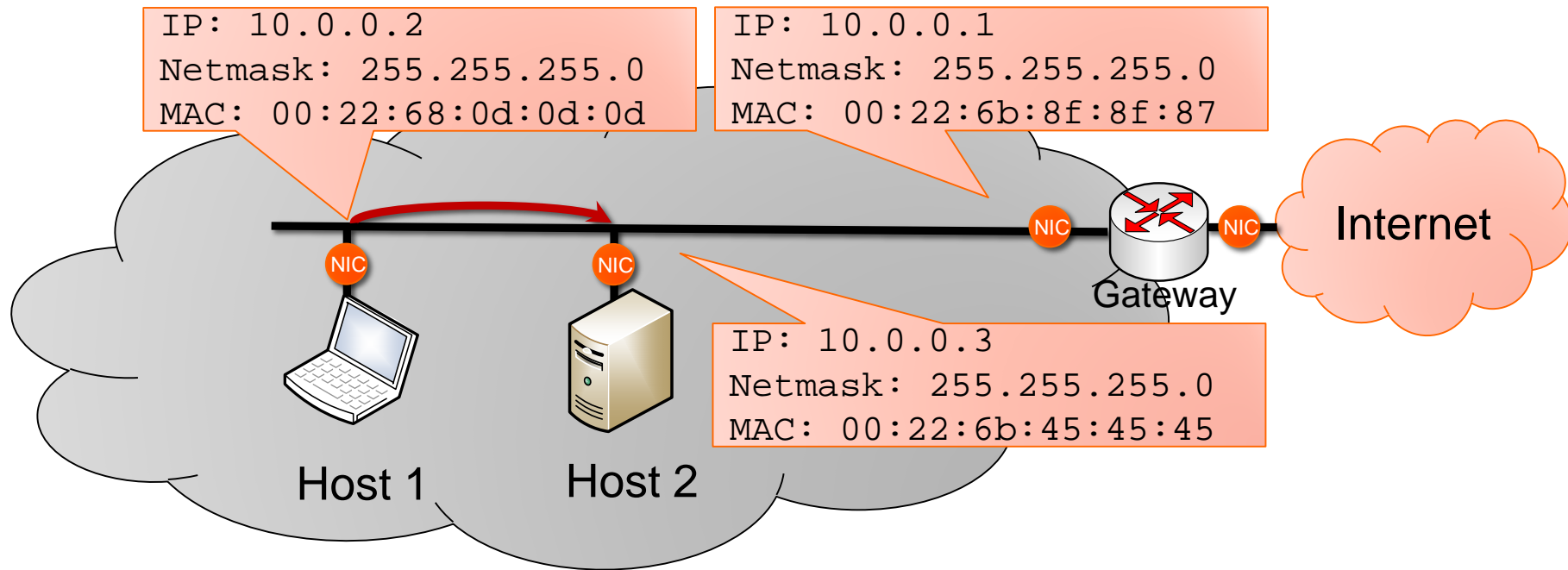
This entry means that the default gateway (i.e. every packet outside the rest routes) is 10.0.0.3



2. IPv4 - Addresses - Routing table

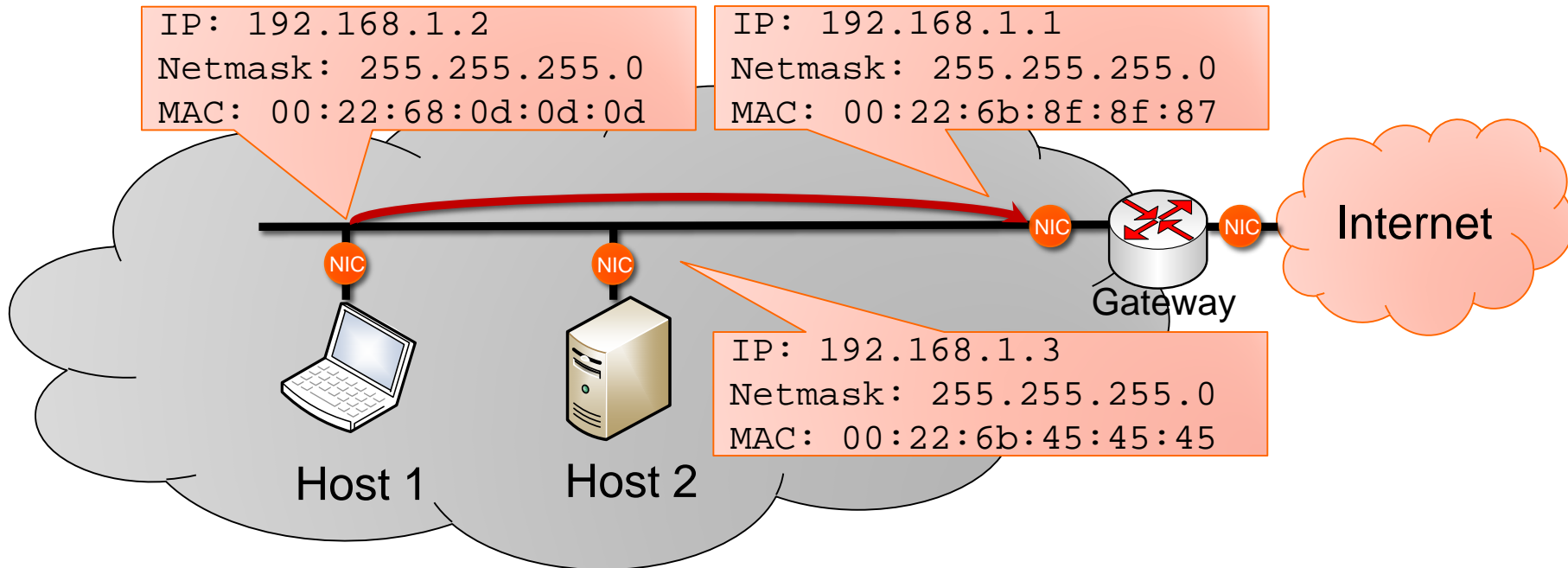
- A host which wants to send an IP packet must know the destination IP
- Using the destination IP and the **routing table**, the source host can find out if the destination host:
 - Case 1 (destination IP in the same subnet): the IP packet will be encapsulated in a frame addressed directly to the destination host
 - Case 2 (destination IP in a different subnet): the IP packet will be encapsulated in a frame addressed to the **gateway** (which provides connectivity to the outside of the network)

2. IPv4 - Addresses - Routing table



- Example 1: Source IP = 10.0.0.2 and destination IP = 10.0.0.3
- Supposing the same routing table of the previous slide, host 1 calculates:
 - Source network prefix = $10.0.0.1 \cdot 255.255.255.0 = 10.0.0.0$
 - Destination network prefix = $10.0.0.3 \cdot 255.255.255.0 = 10.0.0.0$
 - $10.0.0.0 = 10.0.0.0 \rightarrow$ Hosts are in the same subnet
- The packet is encapsulate into a frame which is sent directly to **destination host**

2. IPv4 - Addresses - Routing table

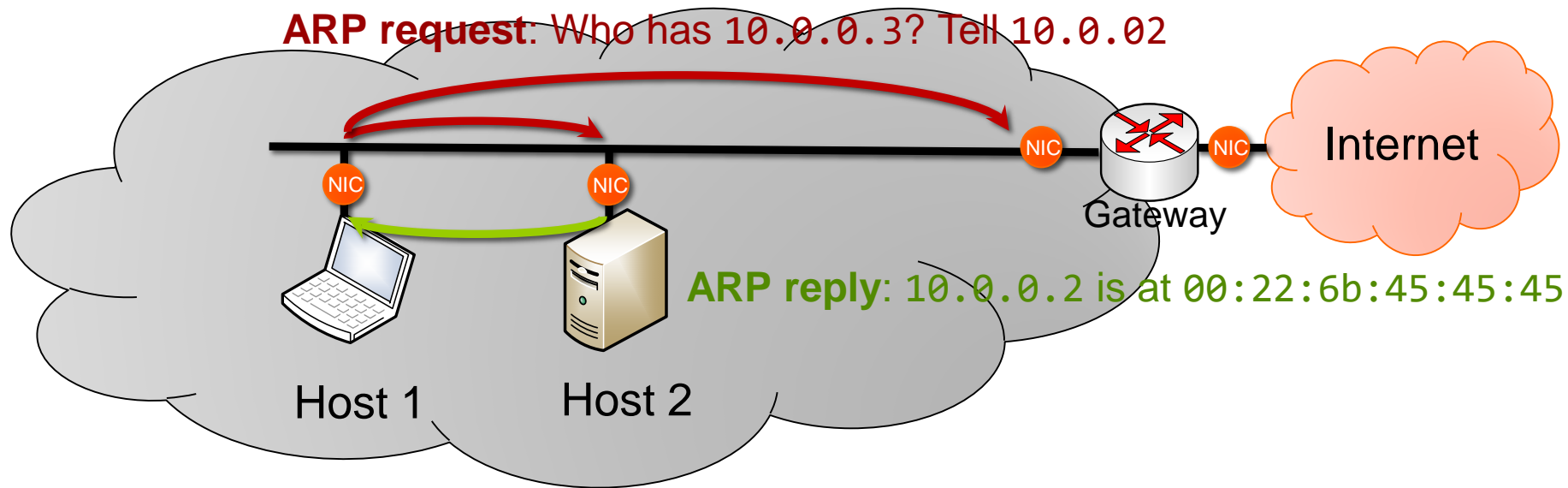


- Example 2: Source IP = 10.0.0.2 and destination IP = 216.58.210.227
- Supposing the same routing table of the previous slide, host 1 calculates:
 - Source network prefix = $10.0.0.1 \cdot 255.255.255.0 = 10.0.0.0$
 - Destination network prefix = $216.58.210.227 \cdot 255.255.255.0 = 216.58.210.0$
 - $10.0.0.0 \neq 216.58.210.0 \rightarrow$ Hosts are in different subnets
- The packet is encapsulate into a frame which is sent directly to the **gateway**

2. IPv4 - Addresses - ARP

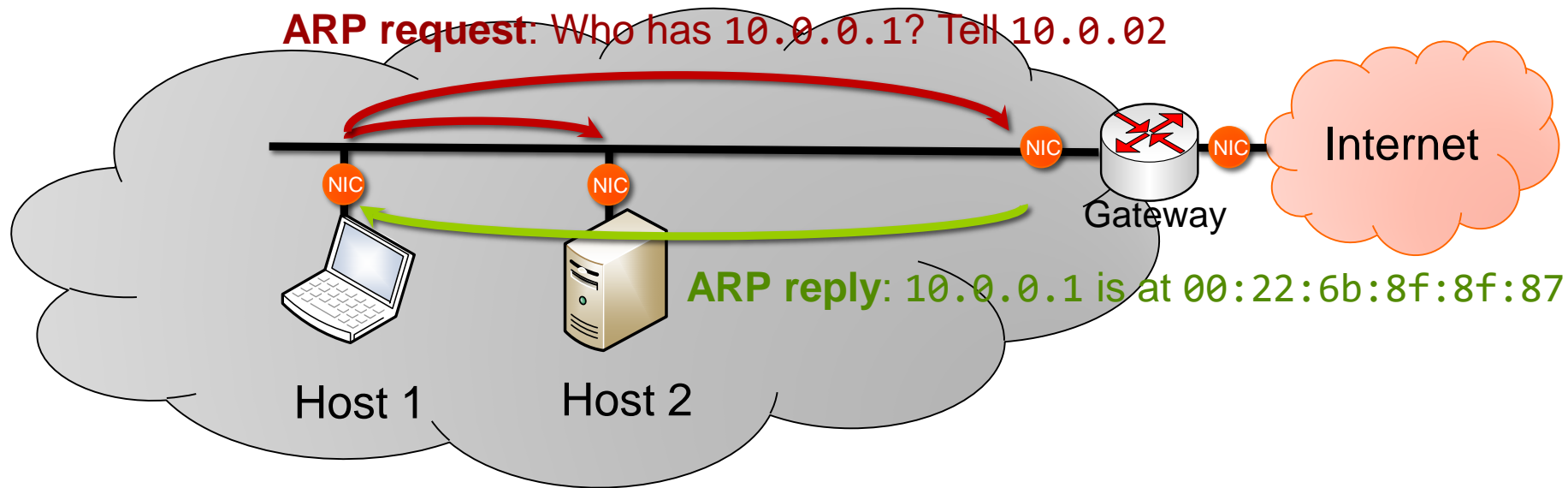
- **ARP** (Address Resolution Protocol) is a link-level protocol used to find the IP address of a certain MAC address
- To find out the MAC address of a given IP address, two types of ARP messages are involved:
 - Request (field operation code = 1)
 - Who has the IP x.x.x.x?
 - They are sent by the broadcast MAC address (FF-FF-FF-FF-FF-FF)
 - Reply (field operation code = 0)
 - Only the owner of that IP will respond
- The information about IP address and its corresponding MAC (physical) address is stored in the **ARP table**

2. IPv4 - Addresses - ARP



- Example 1: Host 1 (10.0.0.2) want to send a frame to Host 2 (10.0.0.3)
 - If MAC address of host 2 is unknown in host 1 ARP table, host 1 sends an ARP request to the whole network segment (MAC address FF:FF:FF:FF:FF:FF)
 - Only the owner of that IP (10.0.0.3) respond with its MAC address

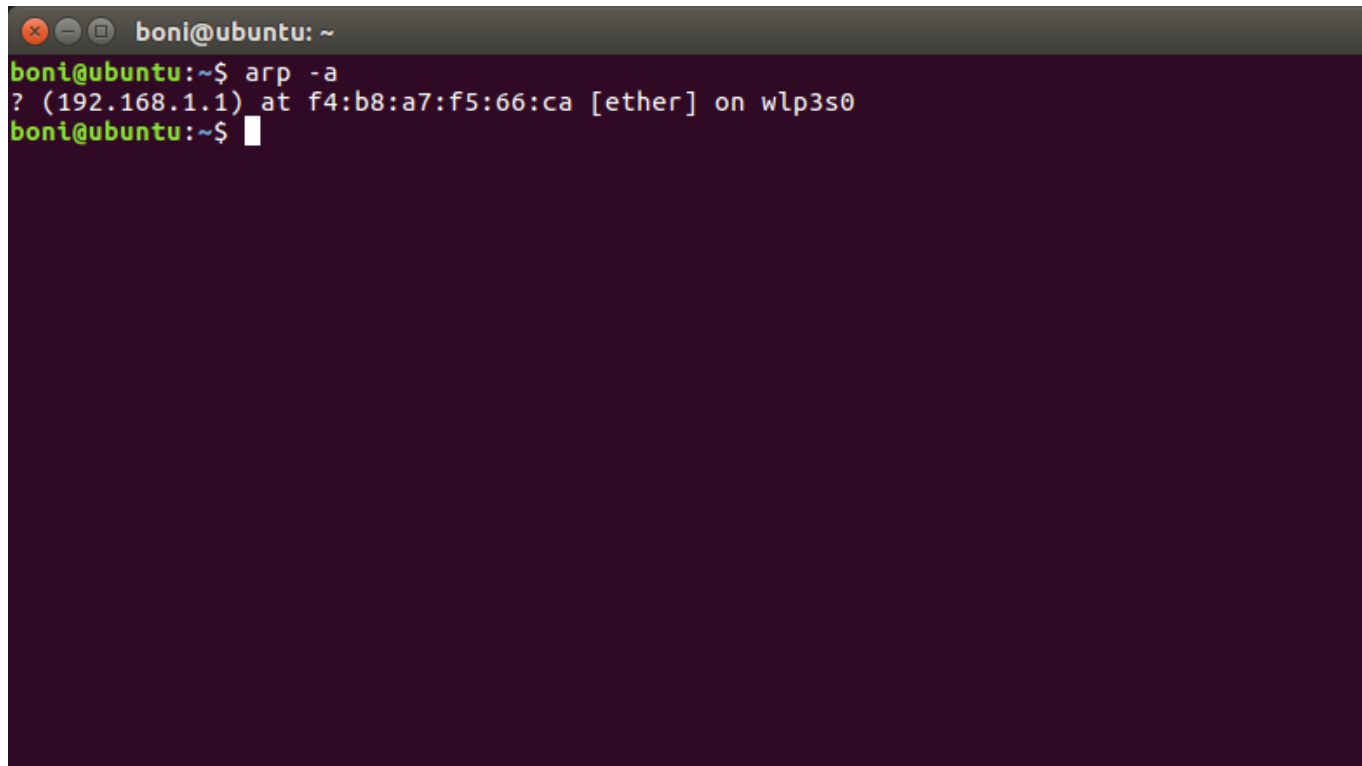
2. IPv4 - Addresses - ARP



- Example 2: Host 1 (10.0.0.2) want to send a frame to Gateway (10.0.0.1)
 - If MAC address of gateway is unknown in host 1 ARP table, host 1 sends an ARP request to the whole network segment (MAC address FF:FF:FF:FF:FF:FF)
 - Only the owner of that IP (10.0.0.1) respond with its MAC address

2. IPv4 - Addresses - ARP

- The value the ARP table can be managed using the command `arp` in the shell:

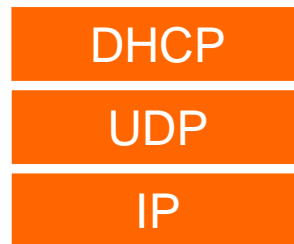
A terminal window with a dark purple background. The title bar shows 'boni@ubuntu: ~'. The prompt is 'boni@ubuntu:~\$'. The user enters 'arp -a'. The output is '? (192.168.1.1) at f4:b8:a7:f5:66:ca [ether] on wlp3s0'. The prompt returns to 'boni@ubuntu:~\$' with a cursor.

```
boni@ubuntu:~$ arp -a
? (192.168.1.1) at f4:b8:a7:f5:66:ca [ether] on wlp3s0
boni@ubuntu:~$
```

2. IPv4 - Addresses - DHCP

DHCP

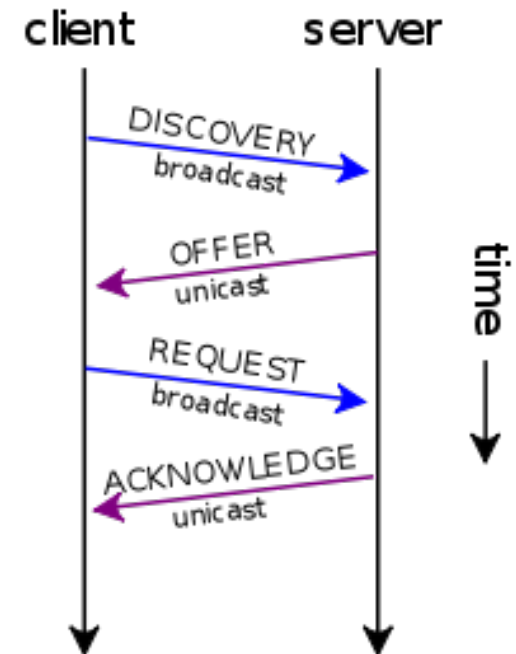
- DHCP = Dynamic Host Configuration Protocol ([RFC 2131](#))
- Client/server application protocol that automates the allocation of network parameters automatically
- Extension to BOOTP (Bootstrap Protocol), which worked statically based on link-level addresses (MAC)
- It works over UDP
 - Client: port 68
 - Server: port 67



UDP is used instead of TCP because DHCP needs to be able to send messages through broadcast traffic, and this type of traffic is not allowed in TCP (a TCP connection is established host-to-host)

2. IPv4 - Addresses - DHCP

- DHCP message exchange:
 1. DHCP Discovery: Broadcast request (to all computers in a network) to identify the DHCP server. If the host that wants to obtain the network configuration does not know the network mask, it will send this message to the generic broadcast address 255.255.255.255
 2. DHCP Offer: DHCP server(s) on the network (there may be several) offer an available IP address
 3. DHCP Request: The client accepts the offer of the IP address through this type of message, again sent by broadcast
 4. DHCP Ack: The assignment is completed by sending a message of assent from the server to the client



2. IPv4 - Addresses - DHCP

- In the DHCP offer, the server provides different types of network configuration parameters, typically:
 - IP address: YIAddr field (your IP address)
 - Network mask: option code 1
 - Gateway: option code 3
 - DNS server(s): option code 6
 - Domain name: option code 15

2. IPv4 - Addresses - IP configuration in hosts



- In GNU/Linux:

Static network setup

```
pc1:~# nano /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.1.39
    netmask 255.255.255.0
    gateway 192.168.1.255
```

Dynamic network setup (DHCP)

```
pc1:~# nano /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

2. IPv4 - Addresses - IP configuration in hosts

- In Windows:



Static network setup

Propiedades: Protocolo de Internet versión 4 (TCP/IPv4)

General

Puede hacer que la configuración IP se asigne automáticamente si la red es compatible con esta funcionalidad. De lo contrario, deberá consultar con el administrador de red cuál es la configuración IP apropiada.

Obtener una dirección IP automáticamente

Usar la siguiente dirección IP:

Dirección IP: 192 . 168 . 1 . 50

Máscara de subred: 255 . 255 . 255 . 0

Puerta de enlace predeterminada: 192 . 168 . 1 . 1

Obtener la dirección del servidor DNS automáticamente

Usar las siguientes direcciones de servidor DNS:

Servidor DNS preferido: 80 . 58 . 61 . 250

Servidor DNS alternativo: 80 . 58 . 61 . 254

Validar configuración al salir

Opciones avanzadas...

Aceptar Cancelar

Dynamic network setup (DHCP)

Propiedades: Protocolo de Internet versión 4 (TCP/IPv4)

General Configuración alternativa

Puede hacer que la configuración IP se asigne automáticamente si la red es compatible con esta funcionalidad. De lo contrario, deberá consultar con el administrador de red cuál es la configuración IP apropiada.

Obtener una dirección IP automáticamente

Usar la siguiente dirección IP:

Dirección IP: . . .

Máscara de subred: . . .

Puerta de enlace predeterminada: . . .

Obtener la dirección del servidor DNS automáticamente:

Usar las siguientes direcciones de servidor DNS:

Servidor DNS preferido: . . .

Servidor DNS alternativo: . . .

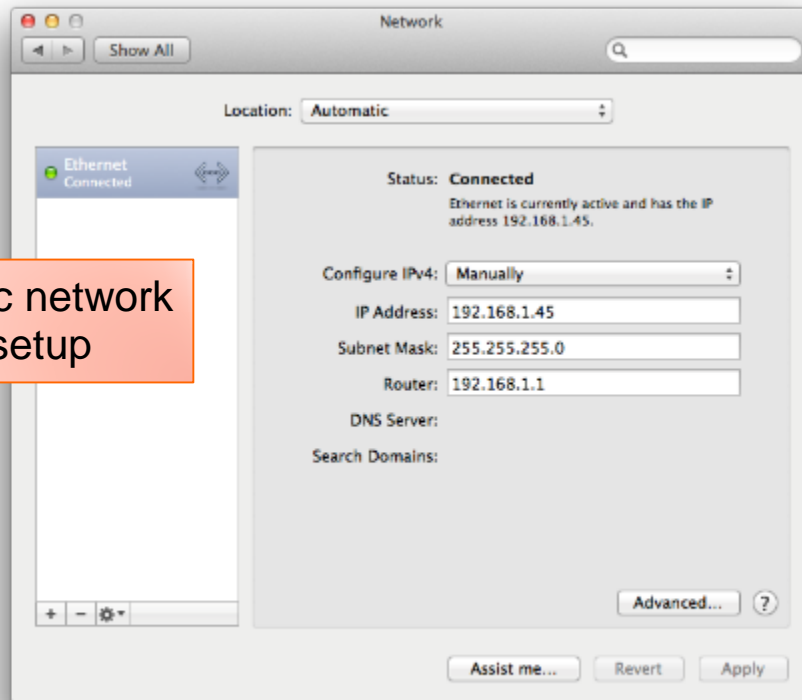
Validar configuración al salir

Opciones avanzadas...

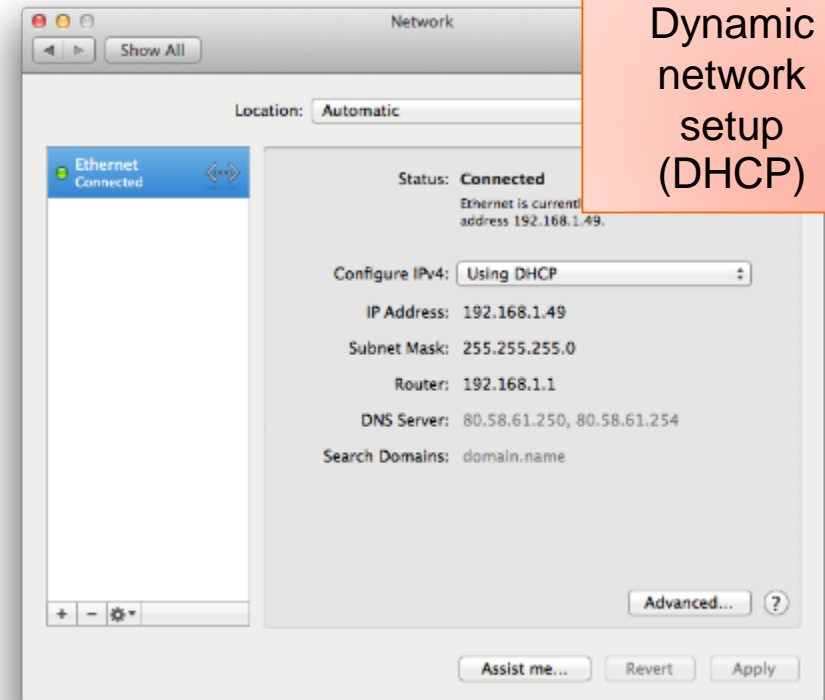
Aceptar Cancelar

2. IPv4 - Addresses - IP configuration in hosts

- In Mac OS:



Static network setup



Dynamic network setup (DHCP)

2. IPv4 - Addresses - IP configuration in hosts

- In Android:



Static network
setup

YOURNETWORK

IP settings

Static

IP address
192.168.1.44

Gateway
192.168.1.1

Network prefix length
24

DNS 1
8.8.4.4

DNS 2
8.8.8.8

Cancel Save

Dynamic
network setup
(DHCP)

YOURNETWORK

WPA/WPA2 PSK

IP address
192.168.1.44

Password
(unchanged)

Show password

Show advanced options

Proxy settings
None

IP settings
DHCP

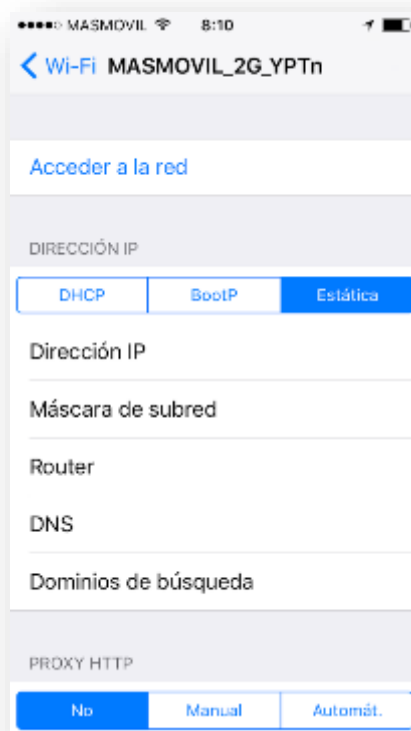
Cancel Save

2. IPv4 - Addresses - IP configuration in hosts

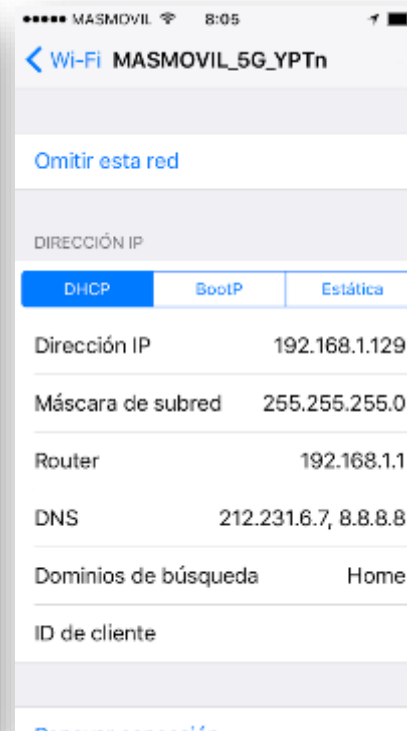
- In iOS:



Static network setup

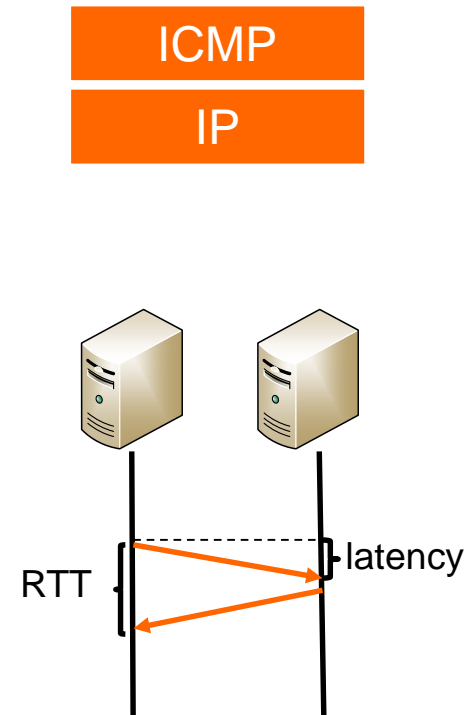


Dynamic network setup (DHCP)



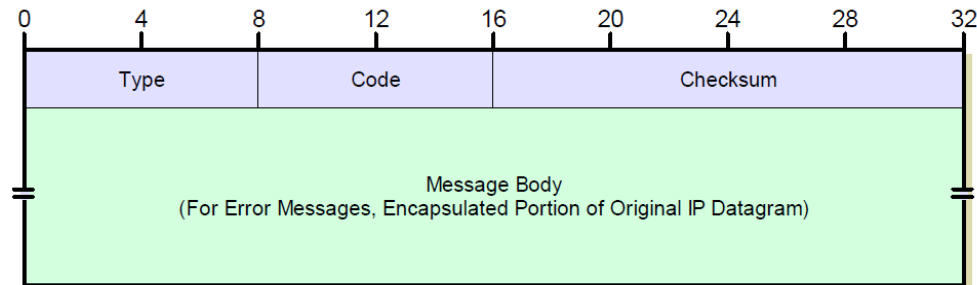
2. IPv4 - Addresses - ICMP

- ICMP (Internet Control Message Protocol) is the control and error notification protocol for IP ([RFC 792](#))
- Several network diagnostic tools use ICMP:
 - Ping: Tool to check connectivity between 2 hosts and calculate the time it takes for the packages to arrive
 - RTT (Round Trip Time) is the time it takes to get a response to a request
 - Network latency \approx RTT / 2
 - Traceroute: Tool that allows you to find out the path (routers / hosts) through which IP packets have passed to reach a destination. In each jump the RTT is calculated



2. IPv4 - Addresses - ICMP

- ICMP message format:

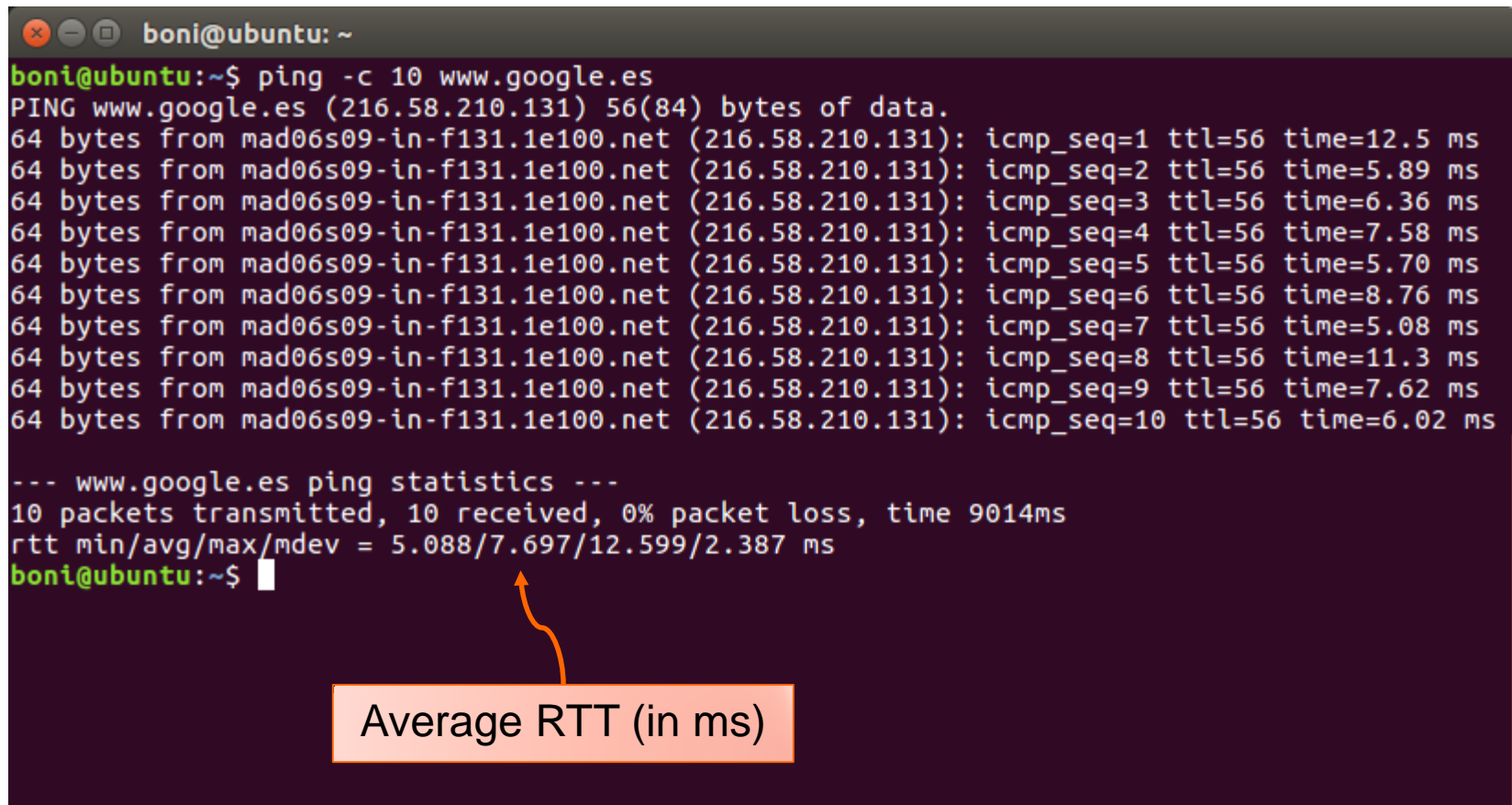


- Some examples of important types:
 - 8: Echo Request: Request echo (ping)
 - 0: Echo Reply: Answer by echo (ping)
 - 11: Time Exceeded:
 - Code 0: TTL exceeded in transit
 - 3: Destination Unreachable: Sent in several situations in which the destination is not reachable. For example:
 - Code 3: Unreachable port
 - Code 4: IP packet too big. Fragmentation is required but the package comes with DF = 1

2. IPv4 - Addresses - ICMP

- Ping:

```
boni@ubuntu: ~  
boni@ubuntu:~$ ping -c 10 www.google.es  
PING www.google.es (216.58.210.131) 56(84) bytes of data.  
64 bytes from mad06s09-in-f131.1e100.net (216.58.210.131): icmp_seq=1 ttl=56 time=12.5 ms  
64 bytes from mad06s09-in-f131.1e100.net (216.58.210.131): icmp_seq=2 ttl=56 time=5.89 ms  
64 bytes from mad06s09-in-f131.1e100.net (216.58.210.131): icmp_seq=3 ttl=56 time=6.36 ms  
64 bytes from mad06s09-in-f131.1e100.net (216.58.210.131): icmp_seq=4 ttl=56 time=7.58 ms  
64 bytes from mad06s09-in-f131.1e100.net (216.58.210.131): icmp_seq=5 ttl=56 time=5.70 ms  
64 bytes from mad06s09-in-f131.1e100.net (216.58.210.131): icmp_seq=6 ttl=56 time=8.76 ms  
64 bytes from mad06s09-in-f131.1e100.net (216.58.210.131): icmp_seq=7 ttl=56 time=5.08 ms  
64 bytes from mad06s09-in-f131.1e100.net (216.58.210.131): icmp_seq=8 ttl=56 time=11.3 ms  
64 bytes from mad06s09-in-f131.1e100.net (216.58.210.131): icmp_seq=9 ttl=56 time=7.62 ms  
64 bytes from mad06s09-in-f131.1e100.net (216.58.210.131): icmp_seq=10 ttl=56 time=6.02 ms  
  
--- www.google.es ping statistics ---  
10 packets transmitted, 10 received, 0% packet loss, time 9014ms  
rtt min/avg/max/mdev = 5.088/7.697/12.599/2.387 ms  
boni@ubuntu:~$
```



Average RTT (in ms)

2. IPv4 - Addresses - ICMP

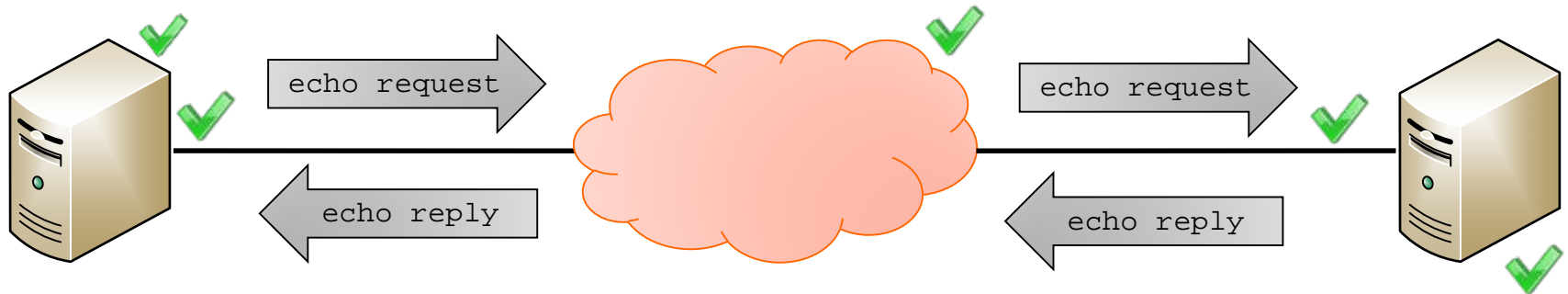
- Traceroute:

```
boni@ubuntu: ~  
boni@ubuntu:~$ traceroute www.google.es  
traceroute to www.google.es (216.58.210.131), 30 hops max, 60 byte packets  
1  * * *  
2  * * *  
3  * * *  
4  * * *  
5  72.14.195.234 (72.14.195.234)  14.291 ms  14.292 ms  13.202 ms  
6  * * *  
7  74.125.253.198 (74.125.253.198)  5.539 ms  72.14.233.124 (72.14.233.124)  61.676 ms  
209.85.142.146 (209.85.142.146)  10.795 ms  
8  108.170.237.167 (108.170.237.167)  5.061 ms  5.524 ms  8.159 ms  
9  mad06s09-in-f131.1e100.net (216.58.210.131)  5.385 ms  44.707 ms  4.658 ms  
boni@ubuntu:~$
```

This tool is called `traceroute` in the Linux/Unix shell and `tracert` in the Windows shell

2. IPv4 - Addresses - ICMP

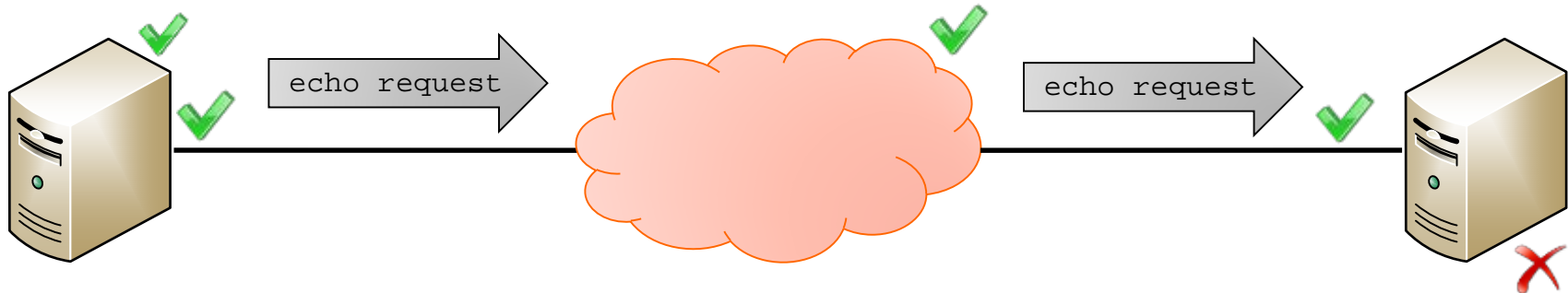
- Ping scenarios:



1. PING success scenario
(remote host is active, so echo
reply arrives at the origin)

2. IPv4 - Addresses - ICMP

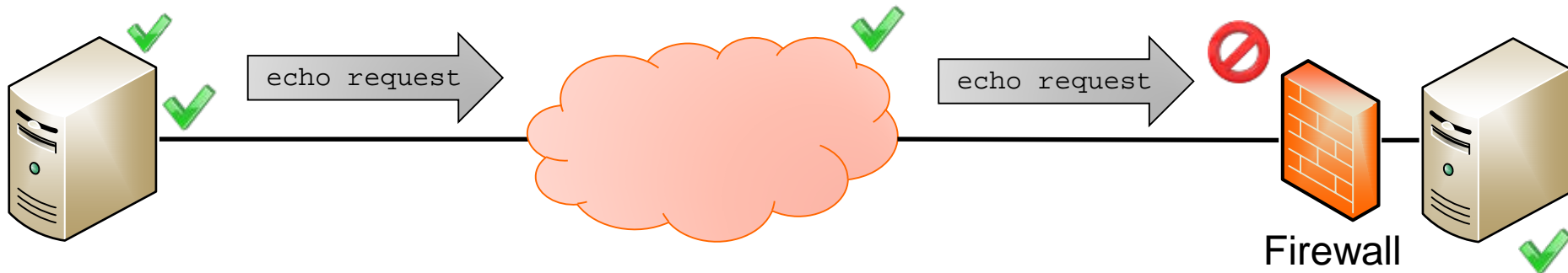
- Ping scenarios:



2. PING failure scenario
(remote host is offline)

2. IPv4 - Addresses - ICMP

- Ping scenarios:



3. Remote host is behind a firewall (software or hardware) that limits ICMP traffic. In this case, we do not get any echo reply

Table of contents

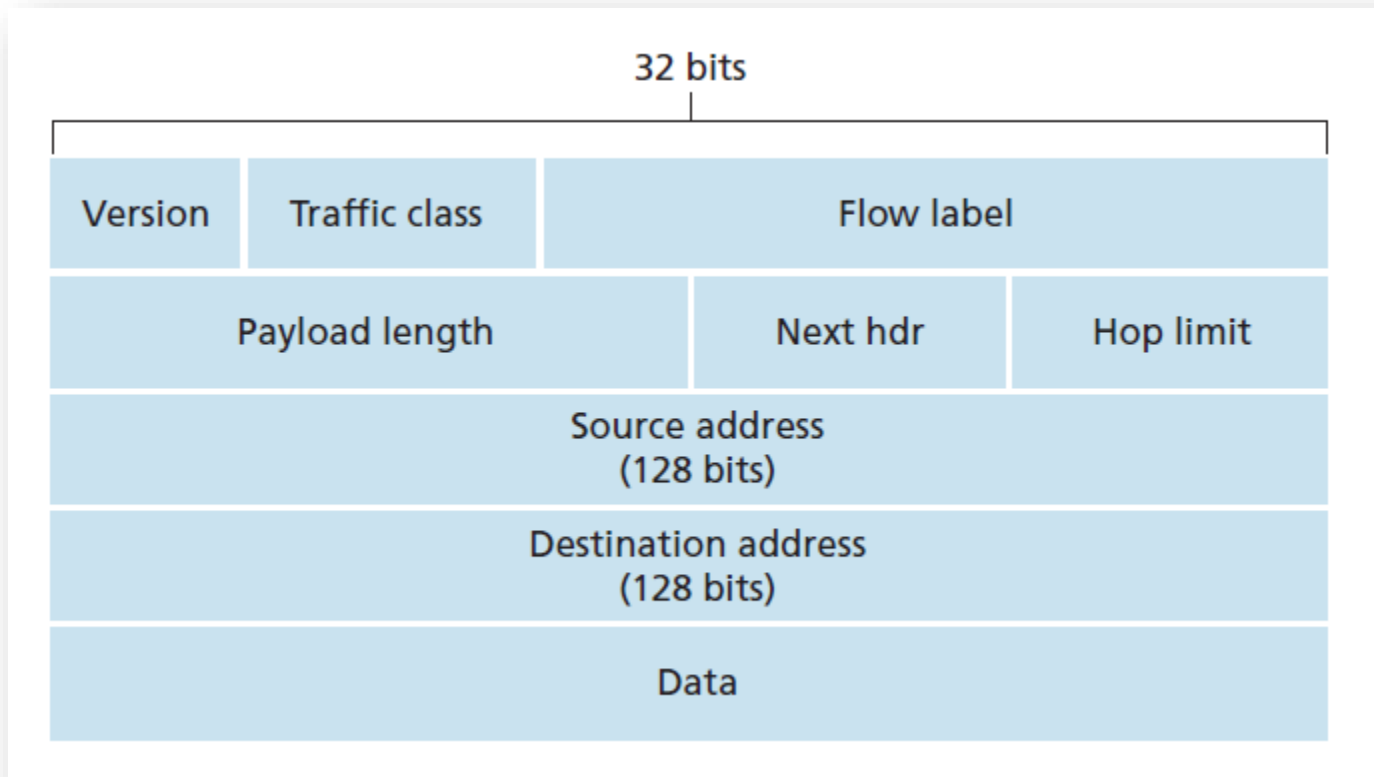
1. Introduction
2. IPv4
- 3. IPv6**
 - I. Differences with IPv4**
 - II. Packet format**
 - III. Addresses**
 - IV. Transition IPv4-IPv6**
4. Routing in Internet
5. Takeaways

3. IPv6 - Differences with IPv4

- Addresses in IPv6 are made up by **128 bits** (16 bytes) instead of 32 bits (4 bytes) in IPv4
 - The total number of IPv6 addresses is 2^{128} addresses (340 billion billion billion addresses)
 - NAT is not required with IPv6 (real end-to-end connectivity is possible)
- Several **simplifications** from IPv6 with respect IPv4:
 - IPv6 header has a fixed size (40 bytes)
 - Fragmentation is only done by the source node (unlike IPv4, in which any intermediate router can fragment a packet)
 - Checksum is not calculated
 - Disappear broadcast addresses (multicast addresses will be used directed to the group formed by all the hosts of a network)

IPv4 and IPv6 are **incompatible** protocols (communication between IPv4 and IPv6 hosts require protocol conversion)

3. IPv6 - Packet format



3. IPv6 - Packet format

- Version (4 bits): 0x06
- Traffic class (8 bits): It allows to set different priority for packets, distinguish multimedia streams, etc.
- Flow label (20 bits): Linked to traffic class. Everything at 0 indicates a best-effort service. Can be used to reserve bandwidth (for example, for streaming)
- Length of the data field (16 bits): Maximum length = 65536 bytes = 64 KB
- Next header (8 bits): Indicates the type of protocol that encapsulates the packet. Follow the same format as IPv4 (0x06 = TCP, 0x11 = UDP ...)
- Limit of jumps (8bits): Same concept as TTL (each router decrements this value and when it reaches 0 the package is discarded)
- Source and destination address (128 bits)
- Data (payload)

3. IPv6 - Addresses

- IPv6 addresses are written as 8 groups of four hexadecimal digits, that is, 16-bit words separated by colons. For example

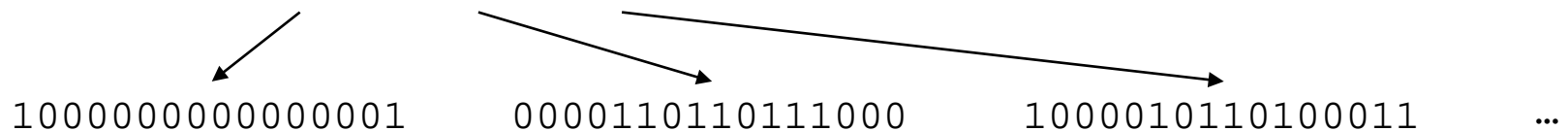
2001:0db8:85a3:08d3:1319:8a2e:0370:7334

- A group of four digits can be compressed when empty (groups of 0). For example:

2001:0db8:85a3:0000:1319:8a2e:0370:7344 =

2001:0db8:85a3:0:1319:8a2e:0370:7344 =

2001:0db8:85a3::1319:8a2e:0370:7344



3. IPv6 - Addresses

- The use of such large addresses allows hierarchical routing to be used
- An IPv6 network uses a group of contiguous IPv6 addresses
- The initial part of the addresses are identical for all hosts in a network, and is called network address or routing prefix
- Network addresses are written in **CIDR** notation. For example:

2001:db8:1234::/48

... starts at the address:

2001:0db8:1234:0000:0000:0000:0000:0001

... and ends in:

2001:0db8:1234:ffff:ffff:ffff:ffff:ffff

3. IPv6 - Addresses

- The address with all zeros is used to indicate the absence of address (it cannot be assigned to a host):

`::/128`

- The loopback address is an address that a node can use to send packets to itself (127.x.x.x from IPv4). It can not be assigned to any physical interface:

`::1/127`

- Multicast addresses The prefix of multicast is:

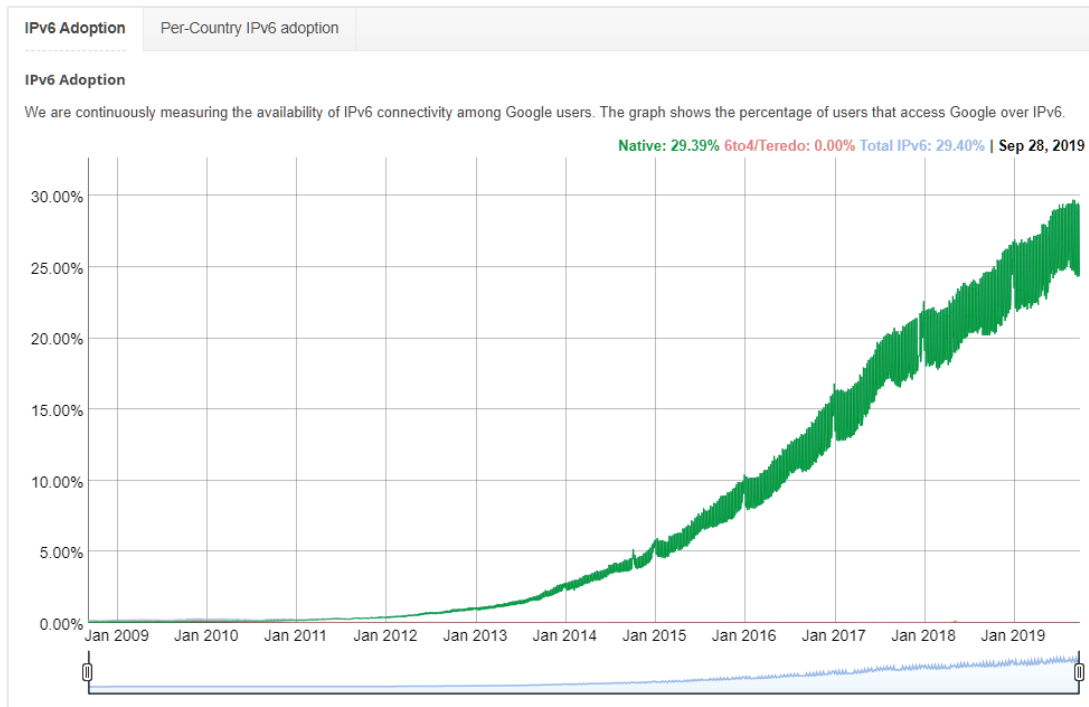
`ff00::/8`

- The mapped IPv4 address is used as a transition mechanism in dual terminals (last 32 bits correspond to IPv4):

`::ffff:192.0.2.128`

3. IPv6 - Transition IPv4-IPv6

- The gradual increase of mobile devices and sensors with Internet connectivity (IoT, Internet of Things) is providing a boost for the implementation of IPv6



<https://www.google.com/intl/en/ipv6/statistics.html>

Although IPv6 was born in 1998, most of the Internet traffic today is still using IPv4

The forecasts suggest that both protocols will coexist for at least 5 to 10 more years

The complete migration to IPv6 will involve updating the Internet infrastructure at its different levels (access network, backbone network) as well as the network configuration of hosts (in other words, hardware and software)

Table of contents

1. Introduction
2. IPv4
3. IPv6
- 4. Routing in Internet**
 - I. Autonomous Systems**
 - II. BGP**
 - III. RIP**
 - IV. OSPF**
5. Takeaways

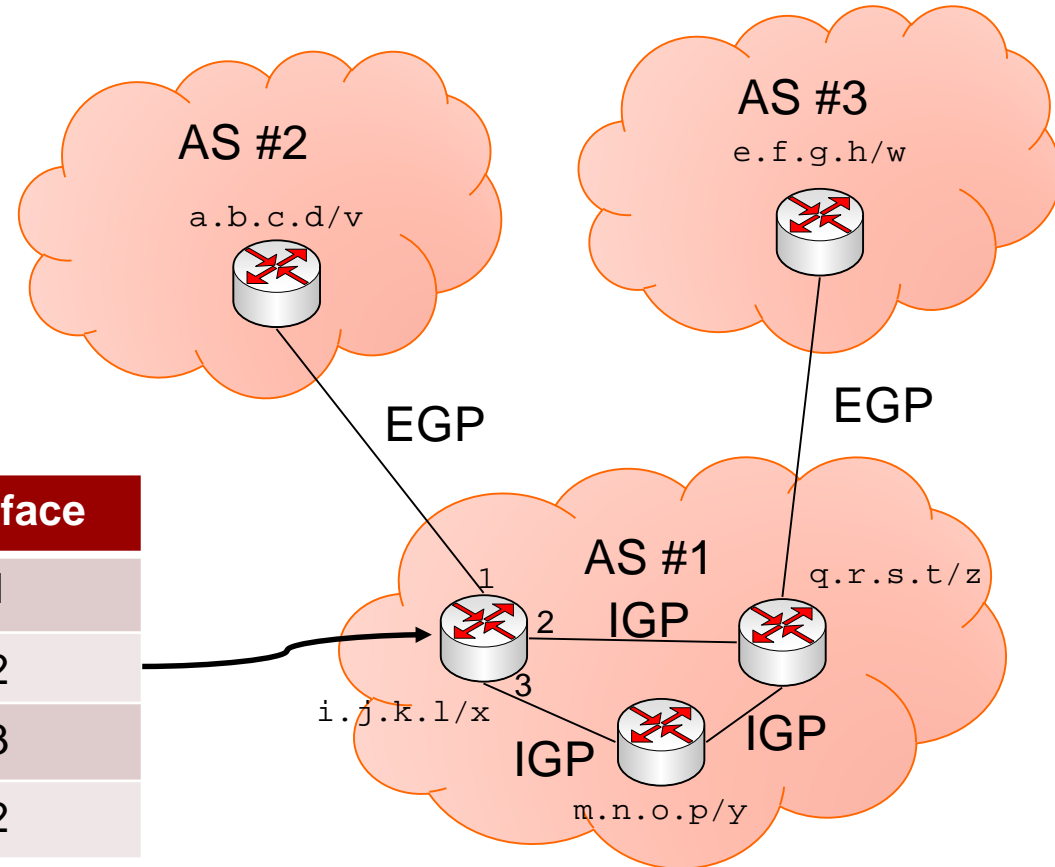
4. Routing in Internet - Autonomous Systems

- As we already know, the **Internet** is the global system of interconnected computer networks that use the Internet protocol suite (TCP/IP) to connect hosts worldwide
- Structurally, the Internet is hierarchical in three levels:
 - Tier-1: Internet backbones
 - Tier-2: Regional ISPs (Internet service providers)
 - Tier-3: ISPs that provide Internet access to final users
- To carry out the routing of IP packets, a different structure is followed, based on **Autonomous Systems (AS)**

4. Routing in Internet - Autonomous Systems

- Autonomous Systems (AS) are a set of subnets and routers managed by a single authority
- Each AS manages a number of networks (classless notation)
- There are different routing protocols that are used to define the contents of the routers' routing tables. These routing protocols are divided into two types:
 - Interior Gateway Protocol (IGP): within an AS:
 - **RIP** (Routing Information Protocol)
 - **OSPF** (Open Shortest Path First)
 - Exterior Gateway Protocol (EGP): between different ASs:
 - **BGP** (Border Gateway Protocol)

4. Routing in Internet - Autonomous Systems



Routing table

| Destination | Gateway | Interface |
|-------------|---------|-----------|
| a.b.c.d/v | a.b.c.d | 1 |
| q.r.s.t/z | q.r.s.t | 2 |
| m.n.o.p/y | m.n.o.p | 3 |
| e.f.g.h/w | q.r.s.t | 2 |

4. Routing in Internet - Autonomous Systems

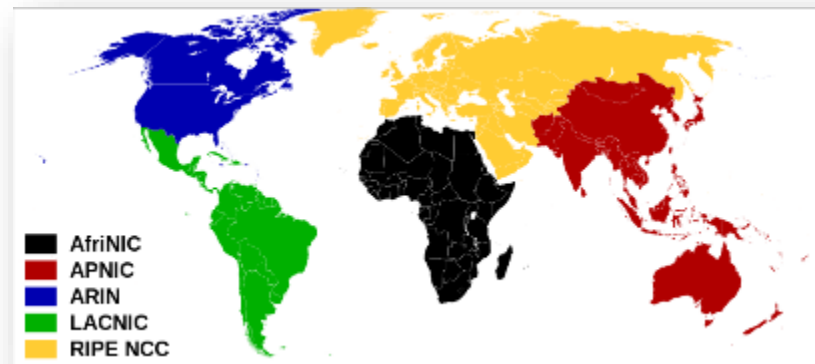
- Each AS has an unique identifier (ASN, Autonomous System Number)
- By mid-2016, 54,000 AS's had been counted on the Internet
- Each Internet service provider (ISP) usually consists of one or more ASs. For example, in Spain:

| ASN | Name | Number of IPs |
|---------|--|---------------|
| AS3352 | Telefónica de España | 10,699,520 |
| AS12479 | France Telecom España | 4,228,864 |
| AS12430 | Vodafone España | 3,389,440 |
| AS6739 | Vodafone Ono | 3,152,128 |
| AS766 | RedIRIS (Entidad Publica Empresarial Red.es) | 1,572,352 |

<https://www.iana.org/assignments/as-numbers/as-numbers.xhtml>

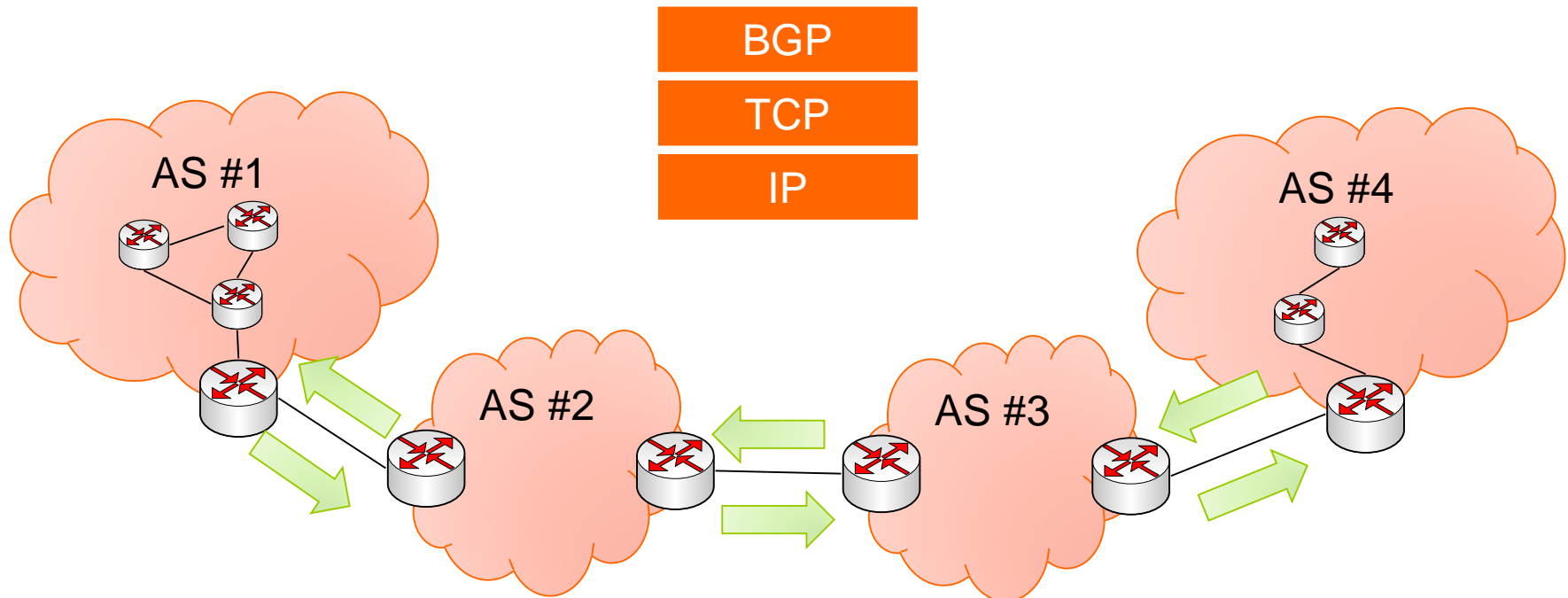
4. Routing in Internet - Autonomous Systems

- At the organizational level, ASs are managed by organizations called Regional Internet Registry (RIR):
 - American Registry for Internet Numbers (ARIN): Anglo-Saxon America
 - RIPE Network Coordination Center (RIPE NCC): Europe, the Middle East and Central Asia
 - Asia-Pacific Network Information Center (APNIC): Asia and the Pacific region
 - Latin American and Caribbean Internet Address Registry (LACNIC): Latin America and the Caribbean
 - African Network Information Center (AfrinIC): Africa



4. Routing in Internet - BGP

- BGP (Border Gateway Protocol) is an external routing protocol (EGP), that is, it exchanges routing information between different ASs
- Each AS must know the subnets that it has inside
- BGP is used for ASs to **announce** network prefixes that are handled within the AS to other ASs



4. Routing in Internet - BGP

- BGP is a critical protocol for the proper functioning of the Internet
 - Incident of the AS 7007:
 - It was a problem that occurred in the year 1997 and that was about to collapse all Internet traffic
 - Due to a bug, the AS 7007 started publishing many more network prefixes than it actually managed
 - This caused that all the Internet traffic of the moment was directed to a 45Mbps connection for a few hours
 - Arab Spring:
 - In 2010 there were demonstrations against the Hosni Mubarak regime in Egypt
 - The Egyptian government ordered all access providers operating in the country to sever their international connections to completely silence the wave of protest
 - The Egyptian routers stopped announcing 3500 BGP routes, leaving the rest of the routers without the necessary information to exchange traffic with Egyptian servers

RIP

UDP

IP

4. Routing in Internet - RIP

- RIP (Routing Information Protocol) is an internal routing protocol (IGP), that is, it manages the information of an AS
- It was one of the first IGP protocols and is still widely used
- RIP allows to **update the routing tables** of the routers of an AS using:
 - The number of **hops** to reach a given destination as metric
 - The **Bellman-Ford** algorithm to obtain the shortest path in the network (graph using hops as weight)
- Pros:
 - Easy to configure
 - It is supported by most manufacturers
- Cons:
 - The use of number of jumps as metric is too simplistic, since it does not take into account other variables such as bandwidth, load of nodes, etc.
 - The maximum jump limit (15) is too small a number that limits the size of the AS

4. Routing in Internet - OSPF

OSPF

IP

- OSPF (Open Shortest Path First) is an internal routing protocol (IGP)
- Same as RIP, OSPF allows to **update the routing tables** of the routers of an AS:
 - The metric used in OSPF is called **cost**, and takes into account various parameters such as bandwidth and latency (calculated RTT)
 - OSPF uses the **Dijkstra** algorithm to obtain the shortest path in the network (graph using cost as weight)
- When ASs are large, OSPF allows to split the AS into smaller groups called areas

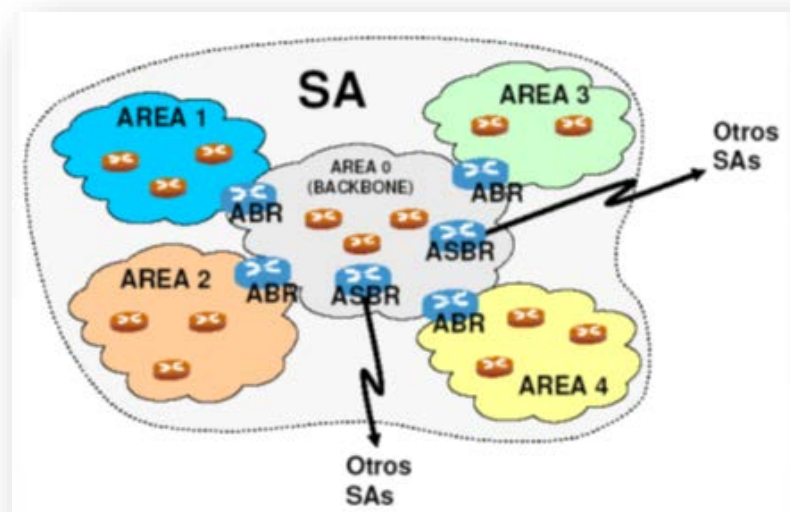


Table of contents

1. Introduction
2. IPv4
3. IPv6
4. Routing in Internet
- 5. Takeaways**

5. Takeaways

- **IP** (Internet Protocol) is the network layer protocol of the TCP/IP protocol stack (Internet)
 - IPv4 (32 bits for addresses) is the most used version nowadays
 - IPv6 (128 bits for addresses) is replacing IPv4
- The **classless** address notation to identify an IP network given its network prefix and the prefix length (w.x.y.z/n) and allows to find out the network identifier (netId) and host identifier (hostId)
 - Netmask \rightarrow netId = 1...1 and hostId = 0...0
 - Broadcast address \rightarrow netId = ... and hostId = 0...0
- **ICMP** is the control and error notification protocol for IP used for diagnostic tools such as ping or traceroute
- **ARP** is a link-level protocol used to find the IP address of a certain MAC address