

Introduction to Computer Networks

Fundamentos de Redes

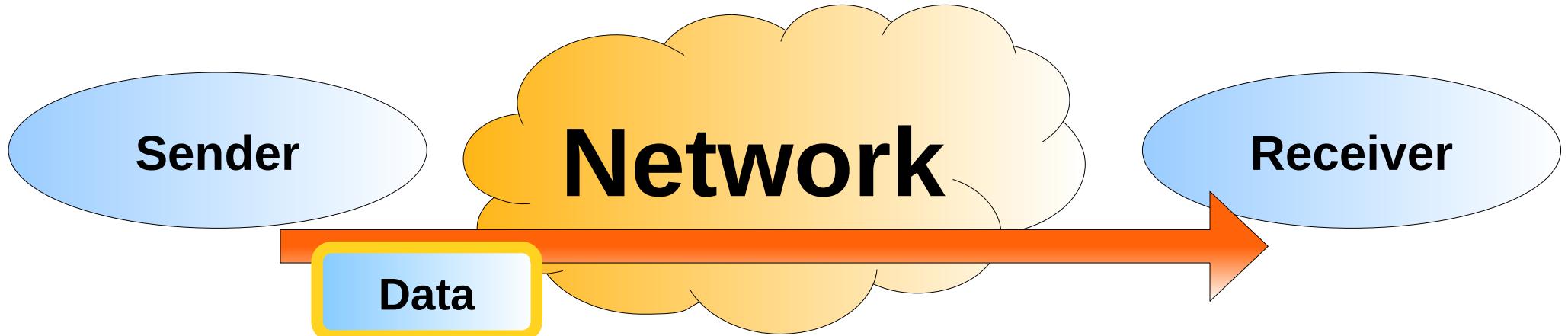
**Mestrado Integrado em
Engenharia de Computadores e Telemática
DETI-UA**



universidade de aveiro

deti.ua.pt

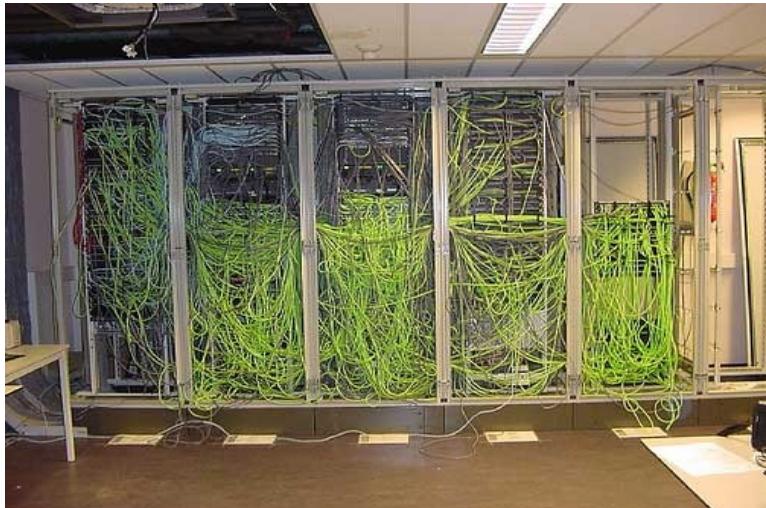
Computer Network



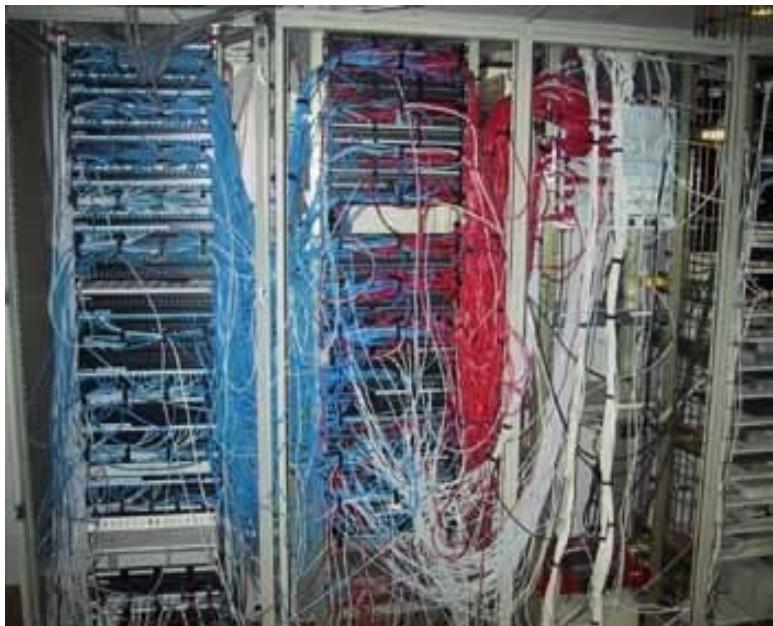
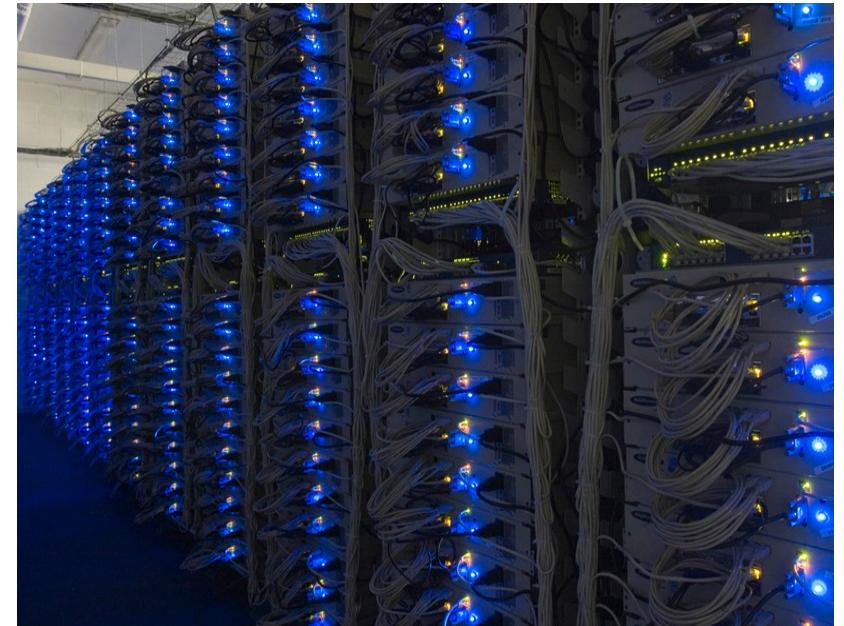
- Purpose: Transmit information/data from a sender to a receiver
 - ◆ Using multiple entities, equipment and services.
 - ◆ Constrained by the sender/receiver requirements
 - ◆ QoS, Security, ...



Different Implementations

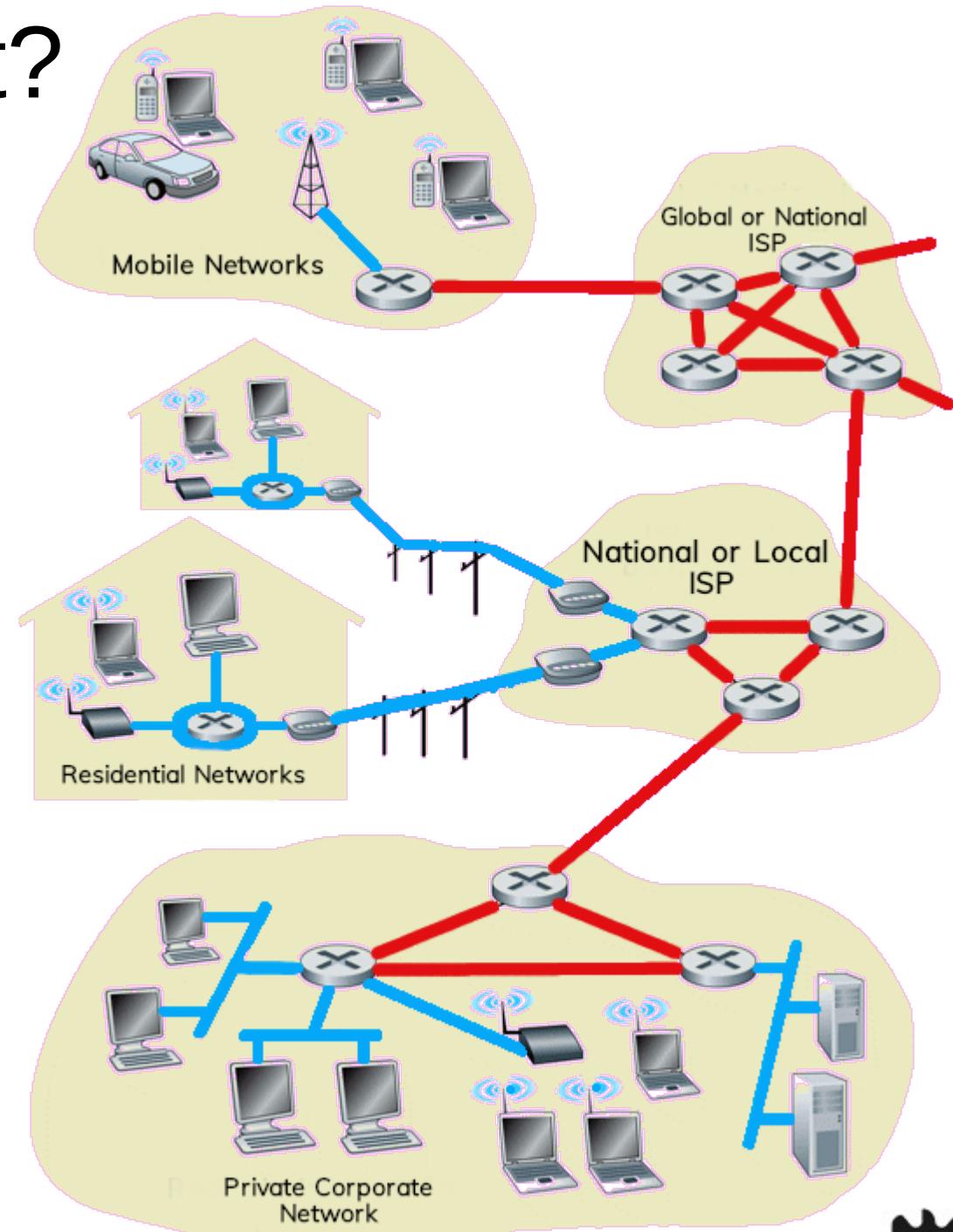


vs

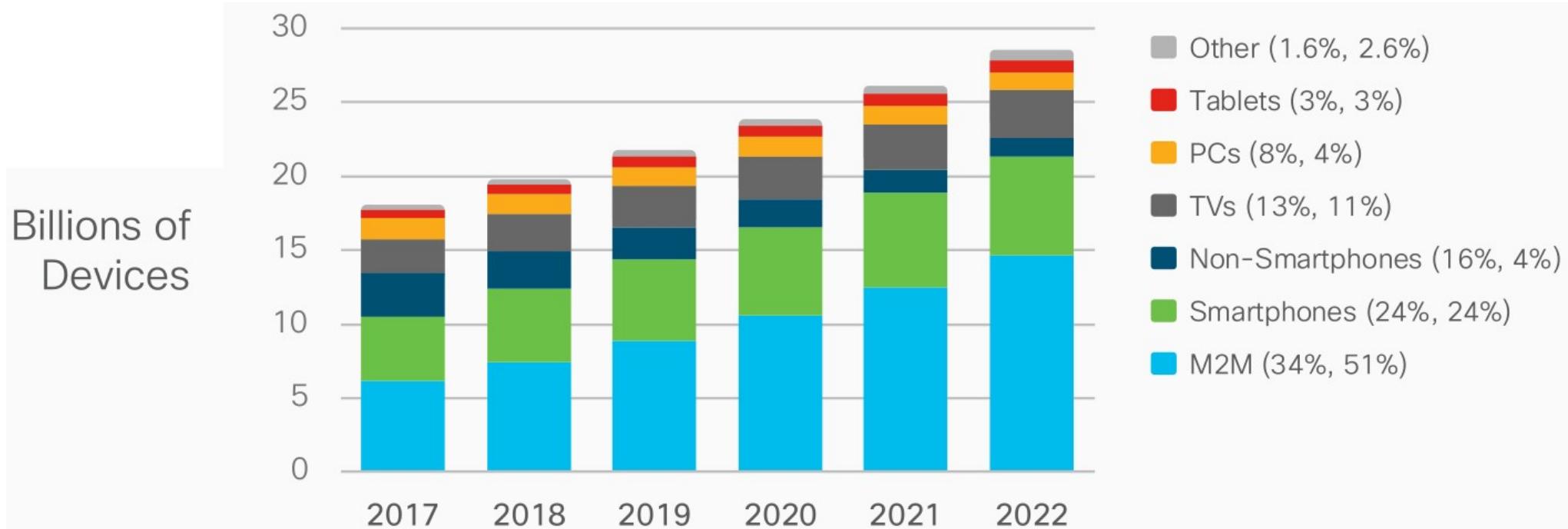


What's the Internet?

- Millions of interconnected devices: hosts or terminal
 - ◆ PCs, servers, phones, tablets, TVs, ...
 - ◆ Execute distributed actions.
- Physical connections
 - ◆ Optical fiber (light), copper (electrons), antennas/satellite (radio), ...
- Routers: devices that interconnect different networks.
- Protocols that control/define data exchange
 - ◆ e.g., TCP, IP, HTTP, FTP, PPP
- Internet: “*network of networks*”
 - ◆ Hierarchical (approximately)
 - ◆ Public vs. Private Internet
 - ◆ Internet neutrality (never existed!)
- Internet Standards
 - ◆ RFC: Request for comments
 - ◆ IETF: Internet Engineering Task Force



Global Devices Growth



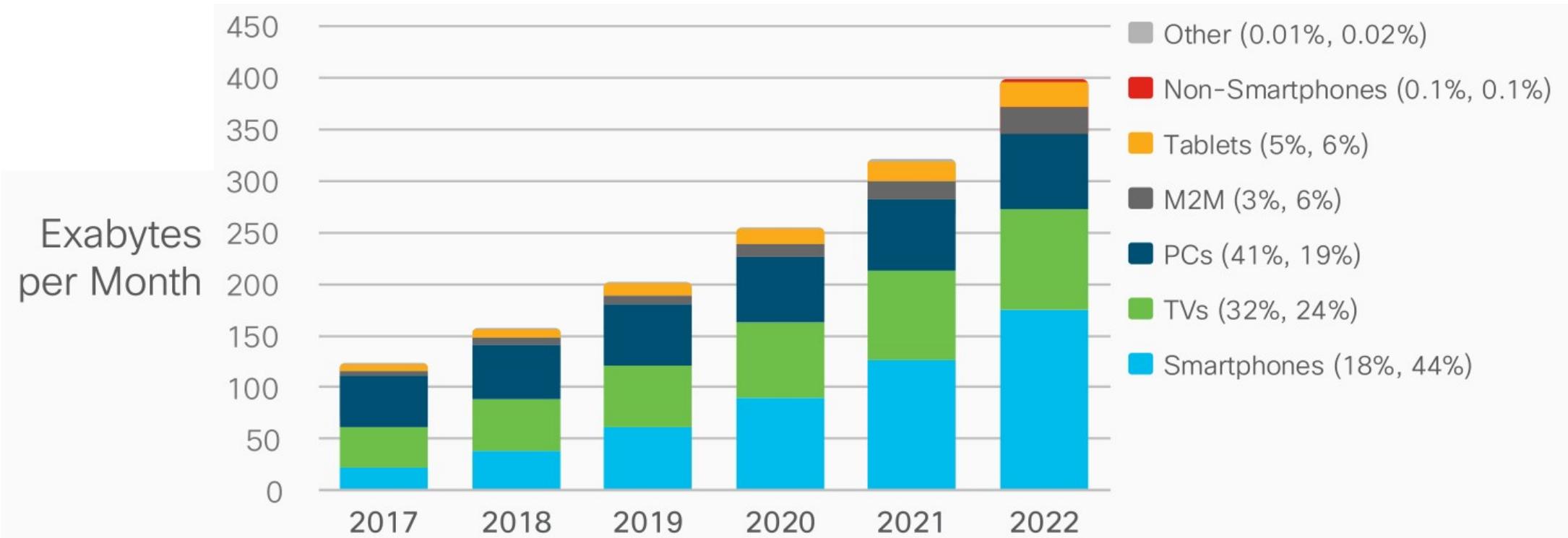
* Figures (n) refer to 2017, 2022 device share

Source: Cisco VNI Global IP Traffic Forecast, 2017-2022

- Device numbers are growing faster than both the population and Internet users.
- By 2022, M2M connections will be 51 percent of the total devices and connections.
 - ◆ Smart meters, video surveillance, healthcare monitoring, transportation, and package or asset tracking.



Global IP Traffic by Devices



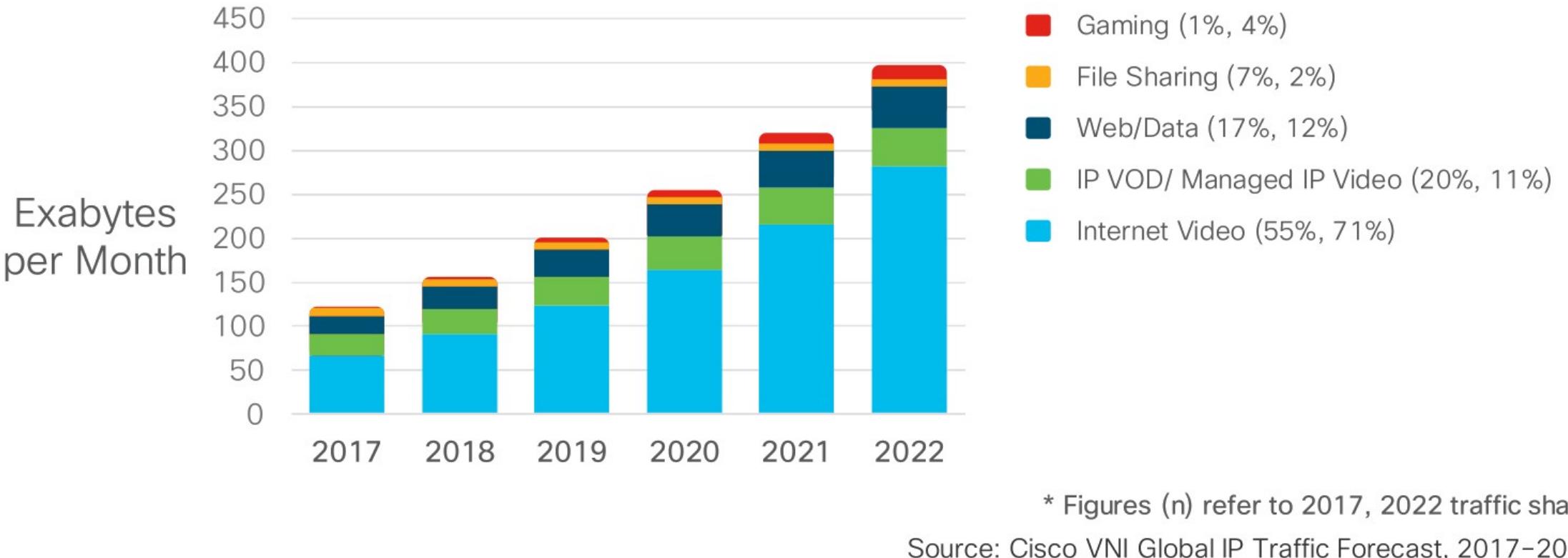
* Figures (n) refer to 2017, 2022 traffic share

Source: Cisco VNI Global IP Traffic Forecast, 2017–2022

- At the end of 2017, 59 percent of IP traffic originated from non-PC devices.
- By 2022, 81 percent of IP traffic will originate from non-PC devices.



IP Traffic by Application

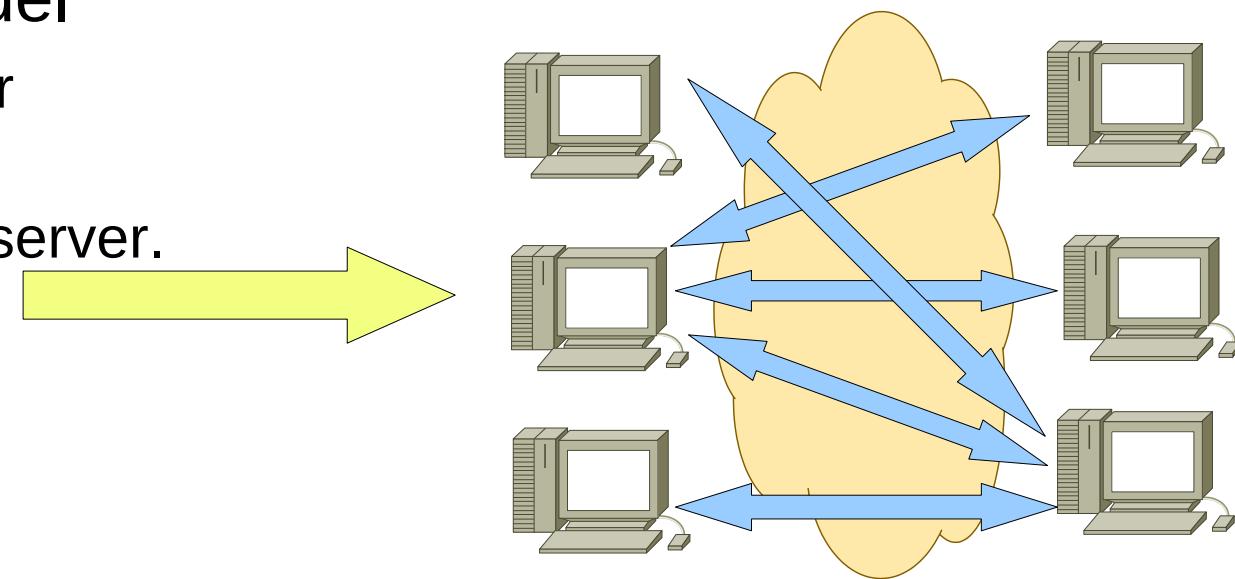
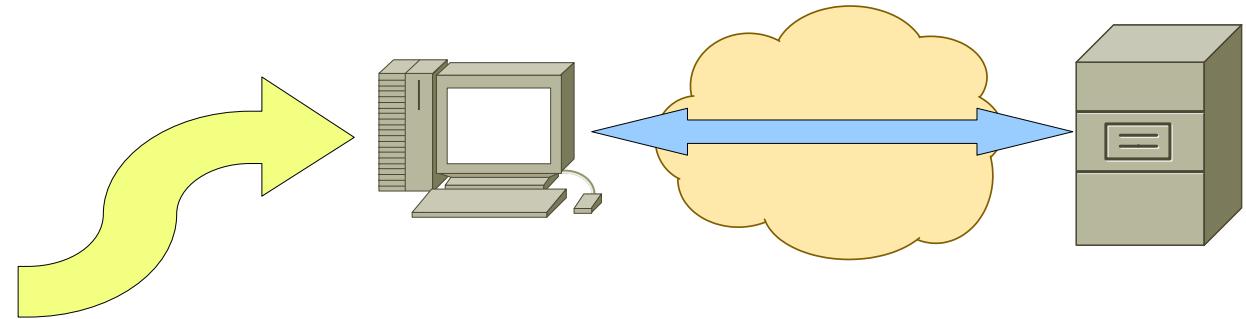


* Figures (n) refer to 2017, 2022 traffic share

Source: Cisco VNI Global IP Traffic Forecast, 2017–2022

Internet Periphery

- Terminals (Hosts)
 - ◆ Run applications.
- Client/Server Model
 - ◆ Host requests data from an always on server.
 - ◆ E.g., browser/Web server; Email client/server.
- Peer-to-Peer (P2P) Model
 - ◆ Minimal (or none) server utilization.
 - ◆ Hosts act as client and server.
 - ◆ E.g., eMule, BitTorrent.



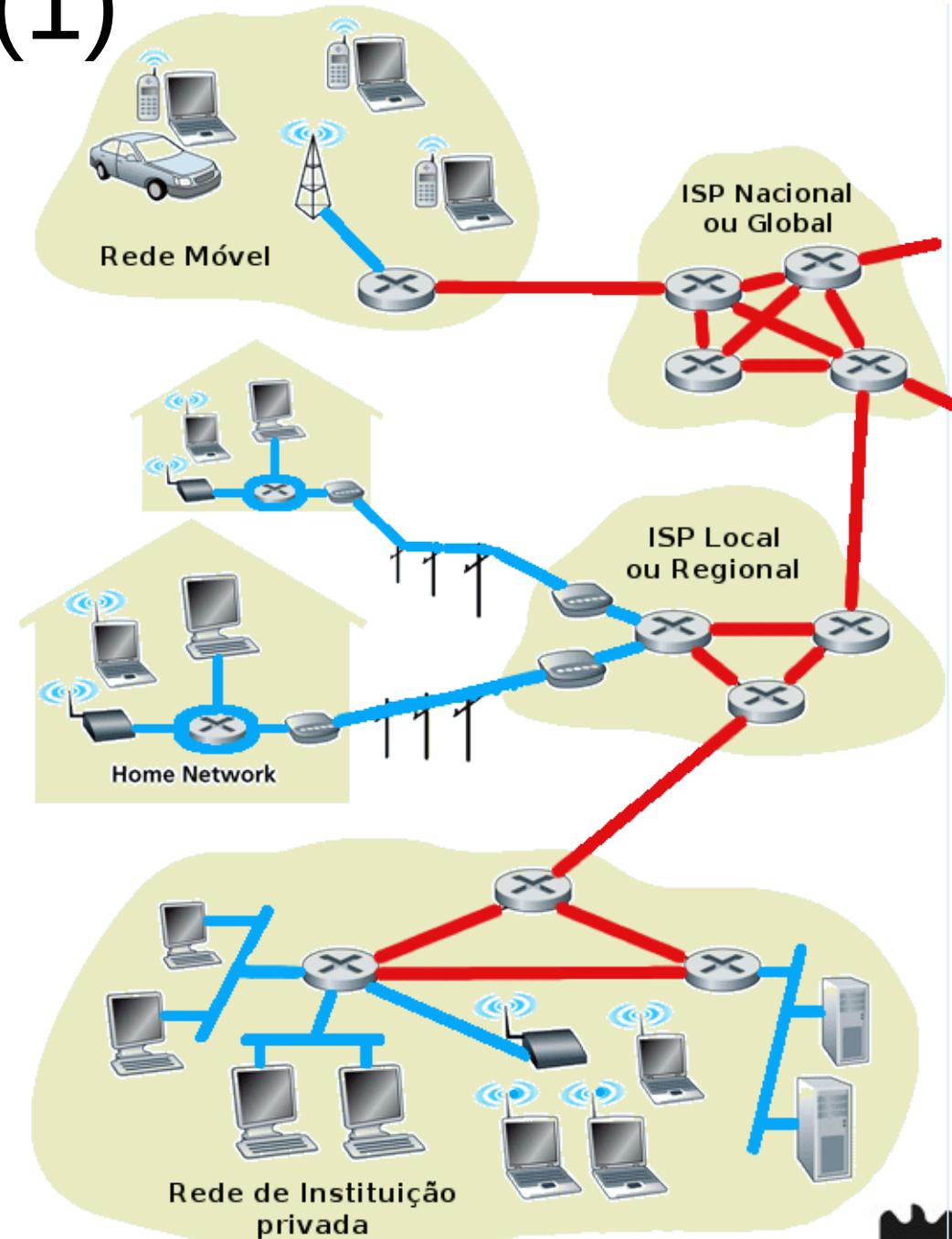
Internet Infrastructure (1)

- **Access Networks**

- ◆ Interconnects hosts to the boundary of Internet.
- ◆ May incorporate more than one technology.
 - ◆ e.g.,
Wireless+Ethernet+FFTB.

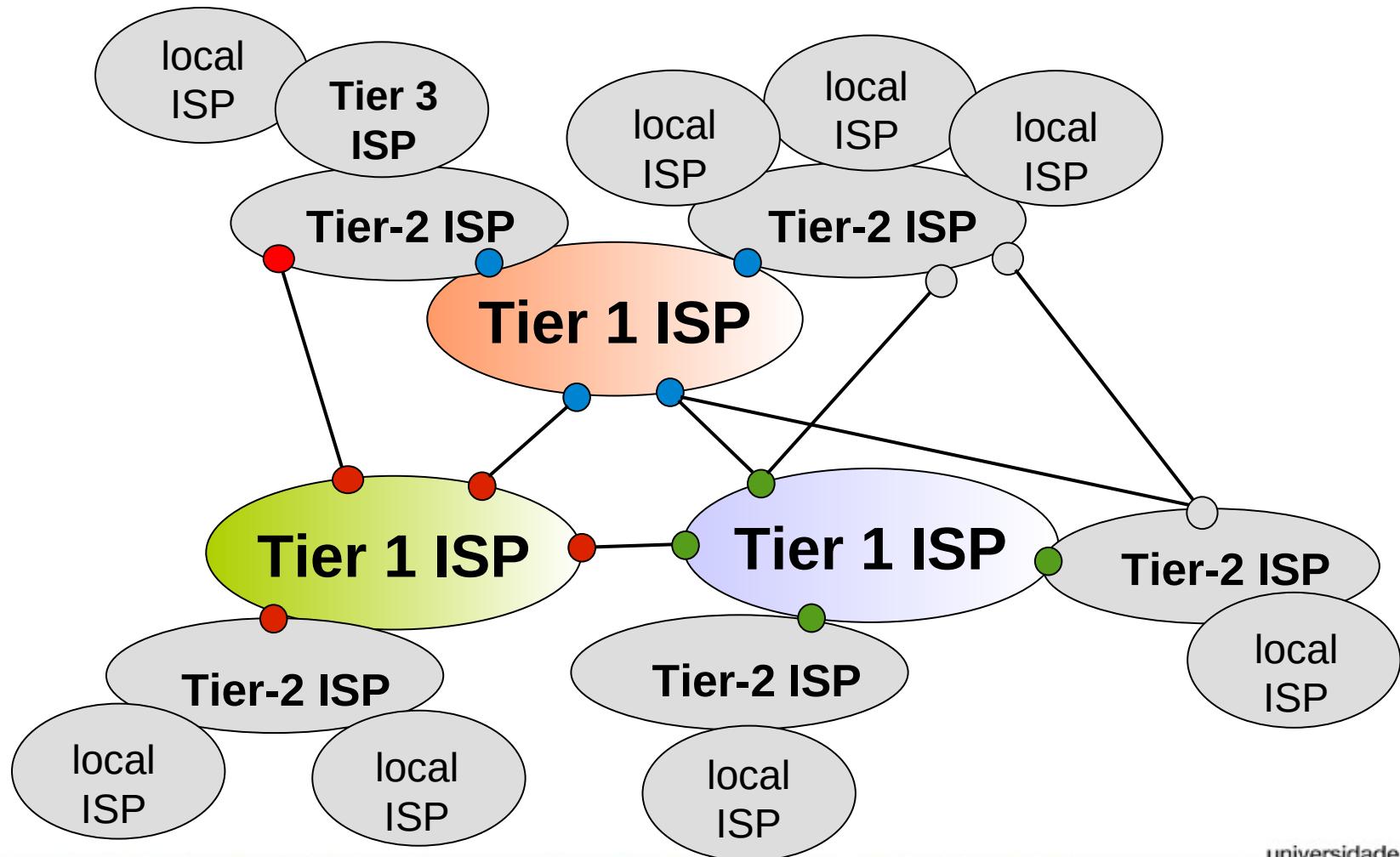
- **Core Network(s)**

- ◆ Routers interconnect multiple access networks.
- ◆ Multiple interconnected core networks are the Internet.



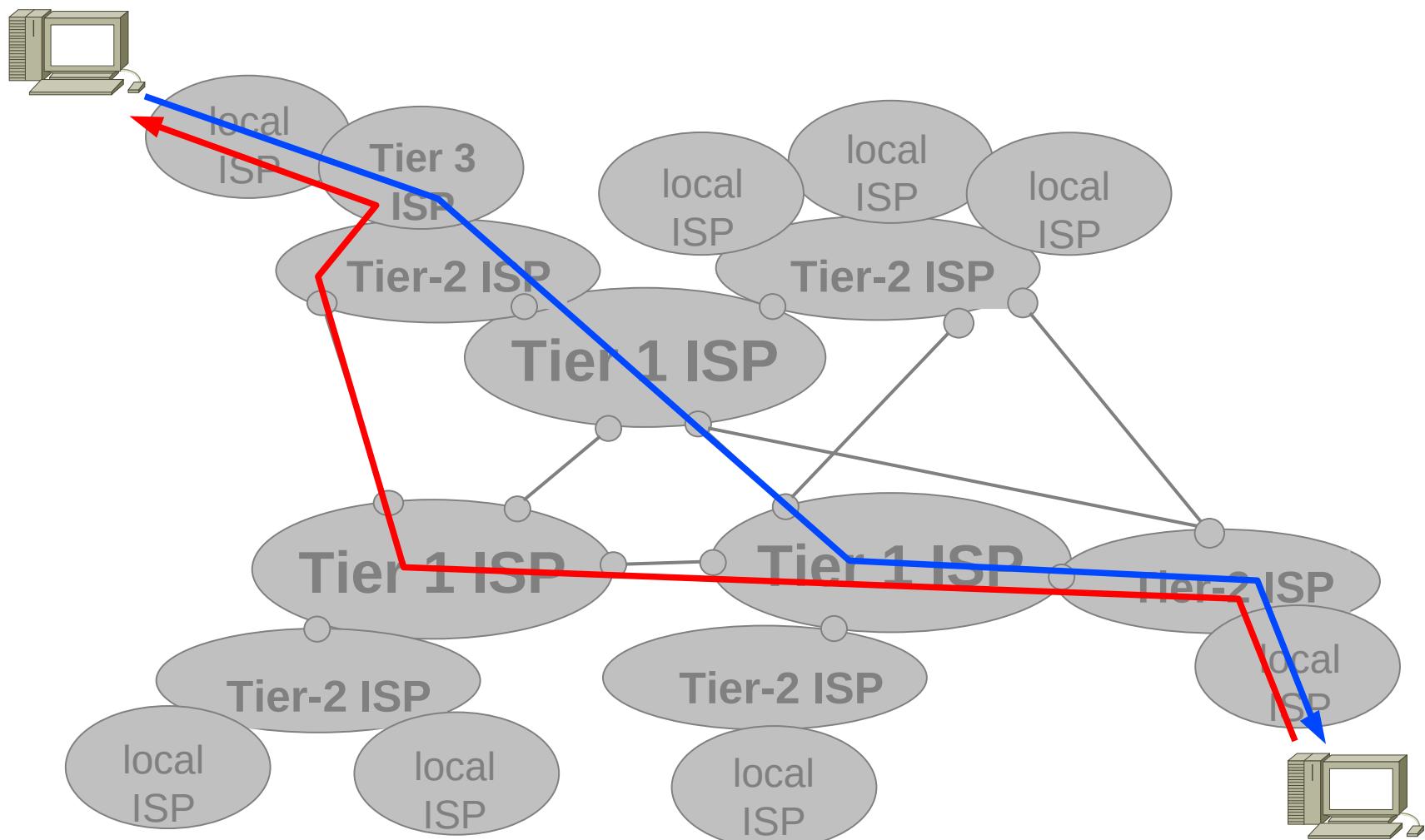
Internet Infrastructure (2)

- Hierarchical (approximately)
- Tier 1 ISP (Global ISPs, e.g Sprint e AT&T), Tier 2 ISP (smaller, nation or region wide), Tier 3 ISP and Local ISP (provide local accesses)



Internet Infrastructure (3)

- Data transverses multiple ISP.
- Paths may not be symmetric (usually are not).



Core Networks

- Datagram Networks

- Networks that provide only a connection-less service.
- Packets reach their intended destination in a different order in which they were sent.
- There is no reservation of resources as there is no dedicated path for a connection session.
- No easy way to guarantee QoS per client.

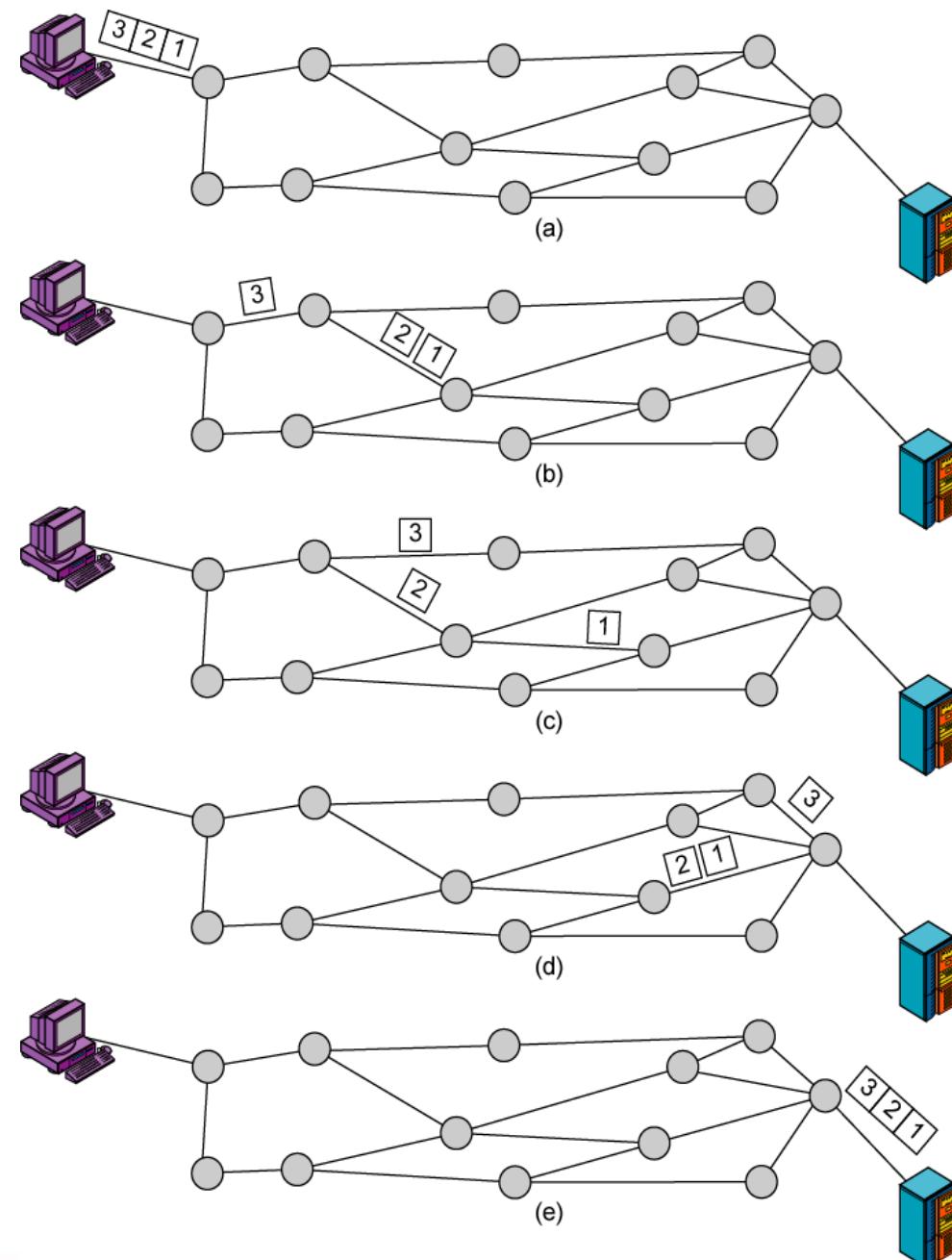
- Virtual Circuits Networks

- Networks that provide only a connection service at the network layer.
- Packets always reach their intended destination in the same order in which they were sent.
- There is a reservation of resources like buffers, CPU, bandwidth, etc for the time in which the VC is going to be used by a data transfer session.
- Evolved from the Phone call concept.
- Is being revived with the advent of virtualization of network functions and services.
- Allows to implement per-client QoS.
- Nowadays operate (virtually) over Datagram networks.

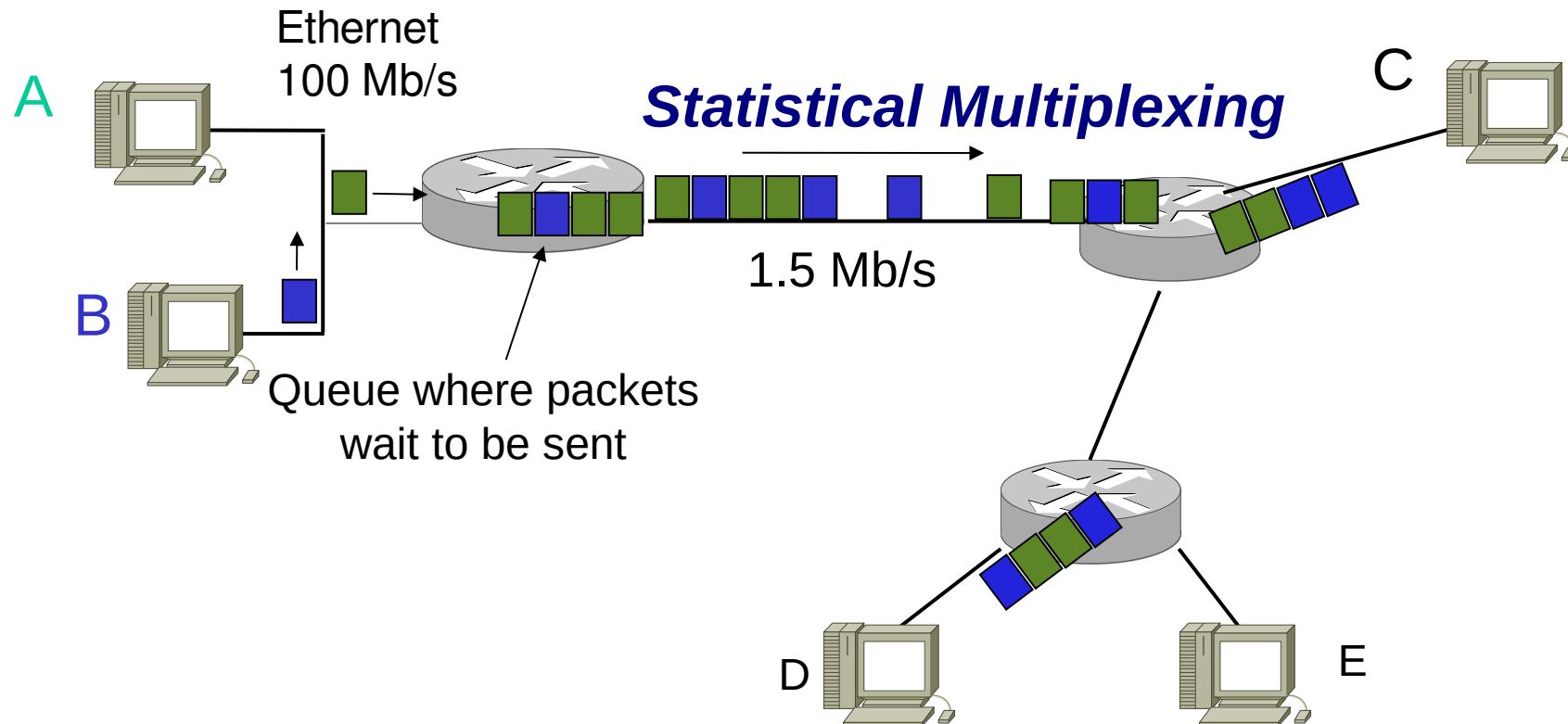


Routing in Datagram Networks

- Each packet is handled independently.
- Packets may take any route.
- Packets may arrive out of order.
- Packets may be lost.
- The receiver has the responsibility to order the packets.
- In some applications, the receiver has the responsibility to recover any lost packet.



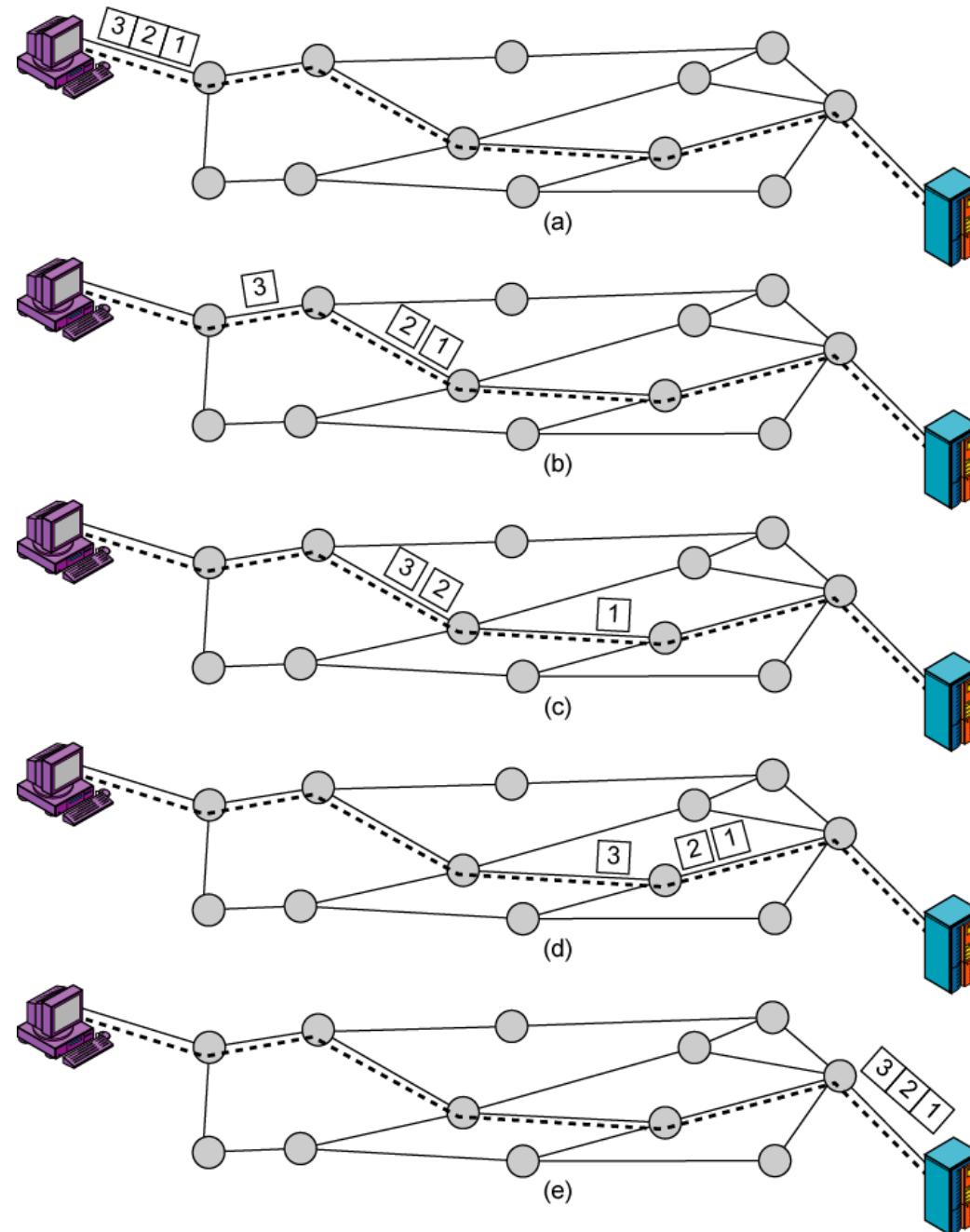
Datagram Networks: Statistical Multiplexing



The sequence of packets (A or B) does not have a fixed statistical multiplexing pattern

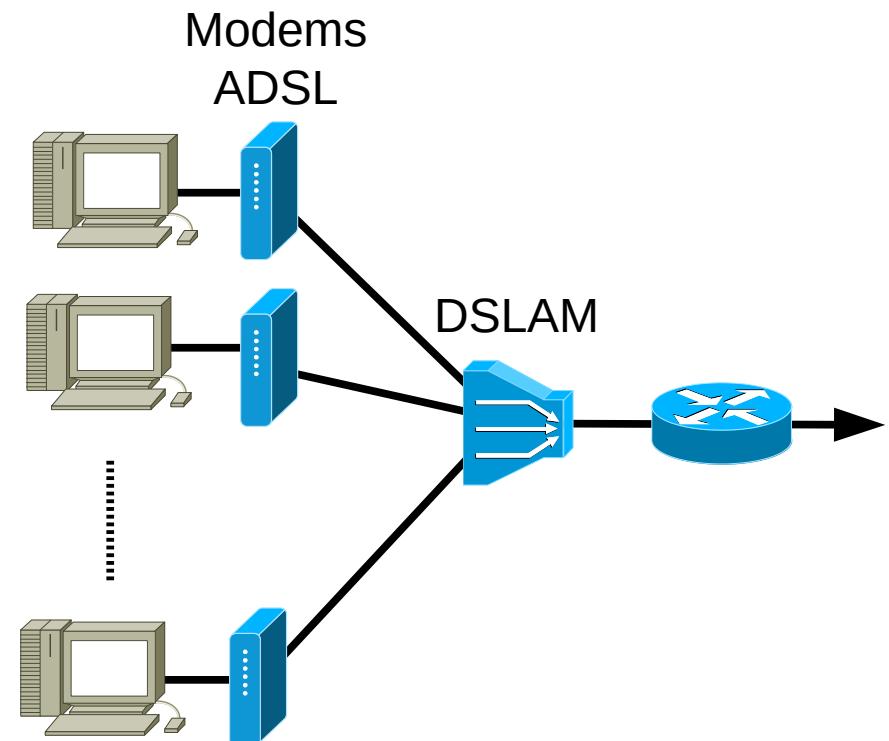


Routing in Virtual Circuits Networks



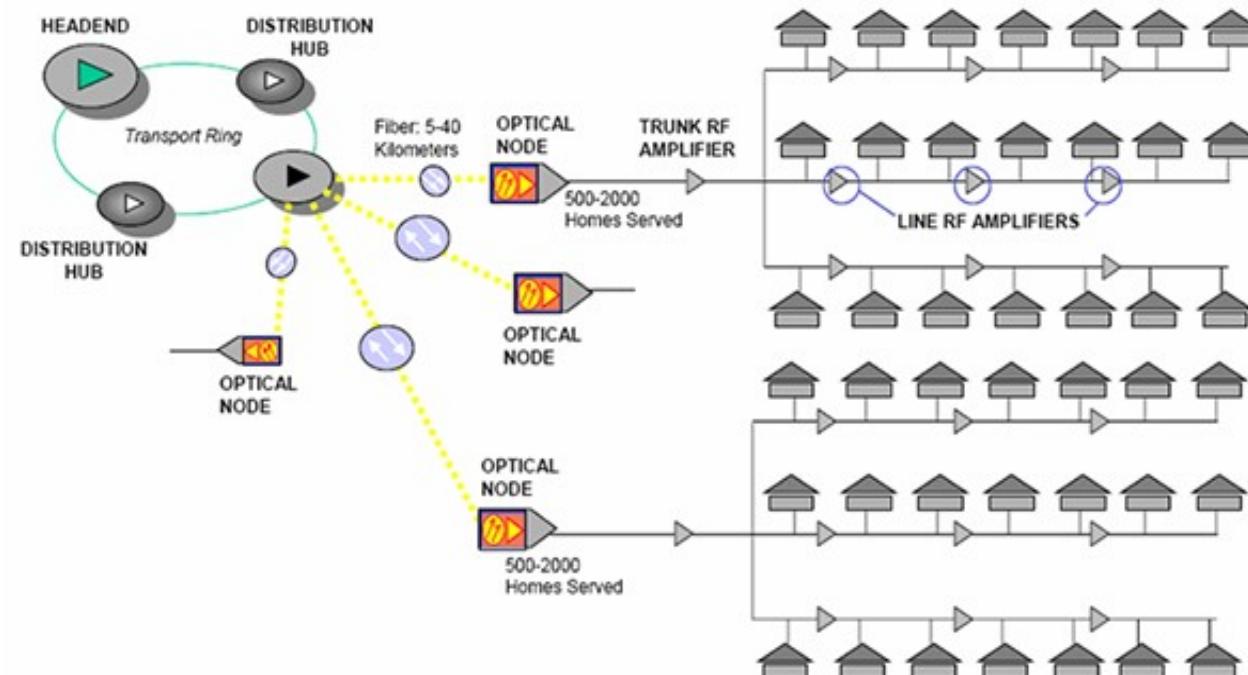
Residential Access Networks: Point-to-Point

- PSTN Modem
 - ◆ Até 56Kb/s de acesso directo ao router
 - ◆ Não era possível telefonar e aceder à Internet ao mesmo tempo
- ADSL: asymmetric digital subscriber line
 - ◆ Até 8Mbps downstream/1Mbps upstream
 - ◆ FDM:
 - ◆ 50 kHz - 1 MHz para downstream
 - ◆ 4 kHz - 50 kHz para upstream
 - ◆ 0 kHz - 4 kHz para telefone tradicional
- ADSL2: 12Mbps/1Mbps
- ADSL2+: 24Mbps/1Mbps
- VDSL: 55Mbps/15Mbps
- VDSL2 (long range): 55Mbps/30Mbps
- VDSL2 (short range): 100Mbps/100Mbps

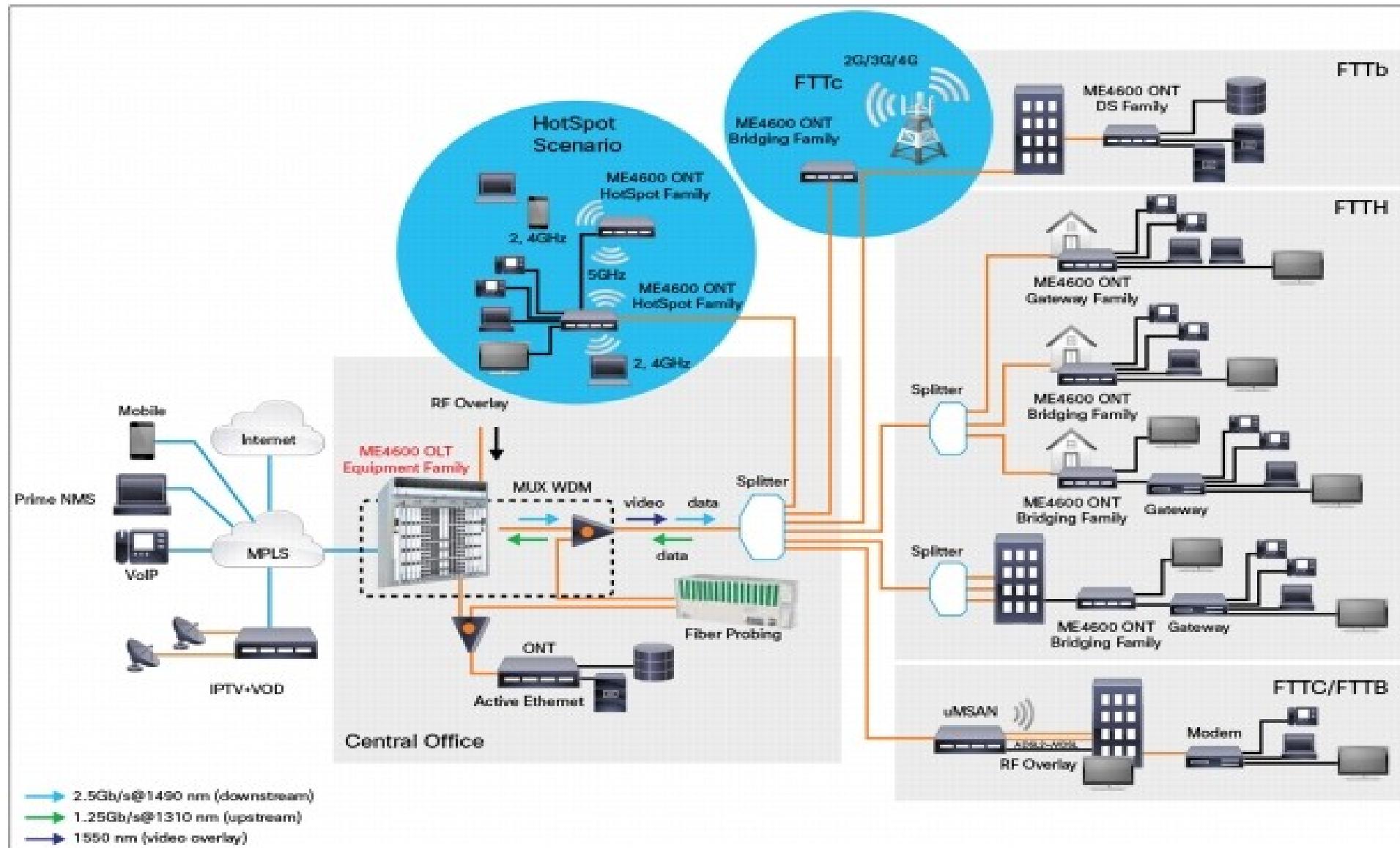


Residential Access Networks: CATV Network

- Rede de cabo e fibra liga habitações ao router do ISP
- HFC: Hybrid Fiber Coax
 - ◆ Assimétrico: até 10Mbps/1 Mbps
- DOCSIS: Data Over Cable Service Interface Specification
 - ◆ Versão 2 - assimétrico: até 50Mbps/27Mbps
 - ◆ Versão 3 (4 canais) - assimétrico: até 200Mbps/108Mbps



Residential and Corporate Access Networks: Fiber to the X (FTTx)

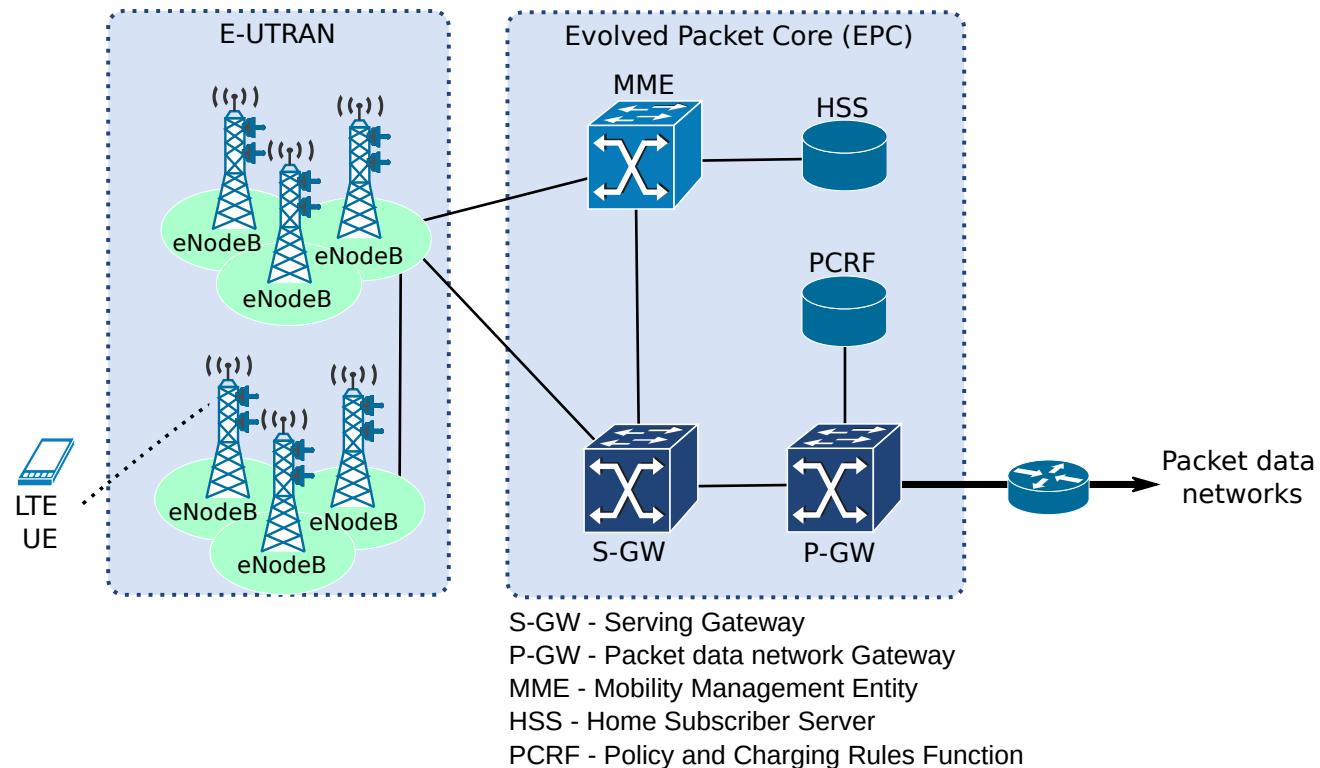


X=Terminal, Home, Building, Closet/Curb, ...



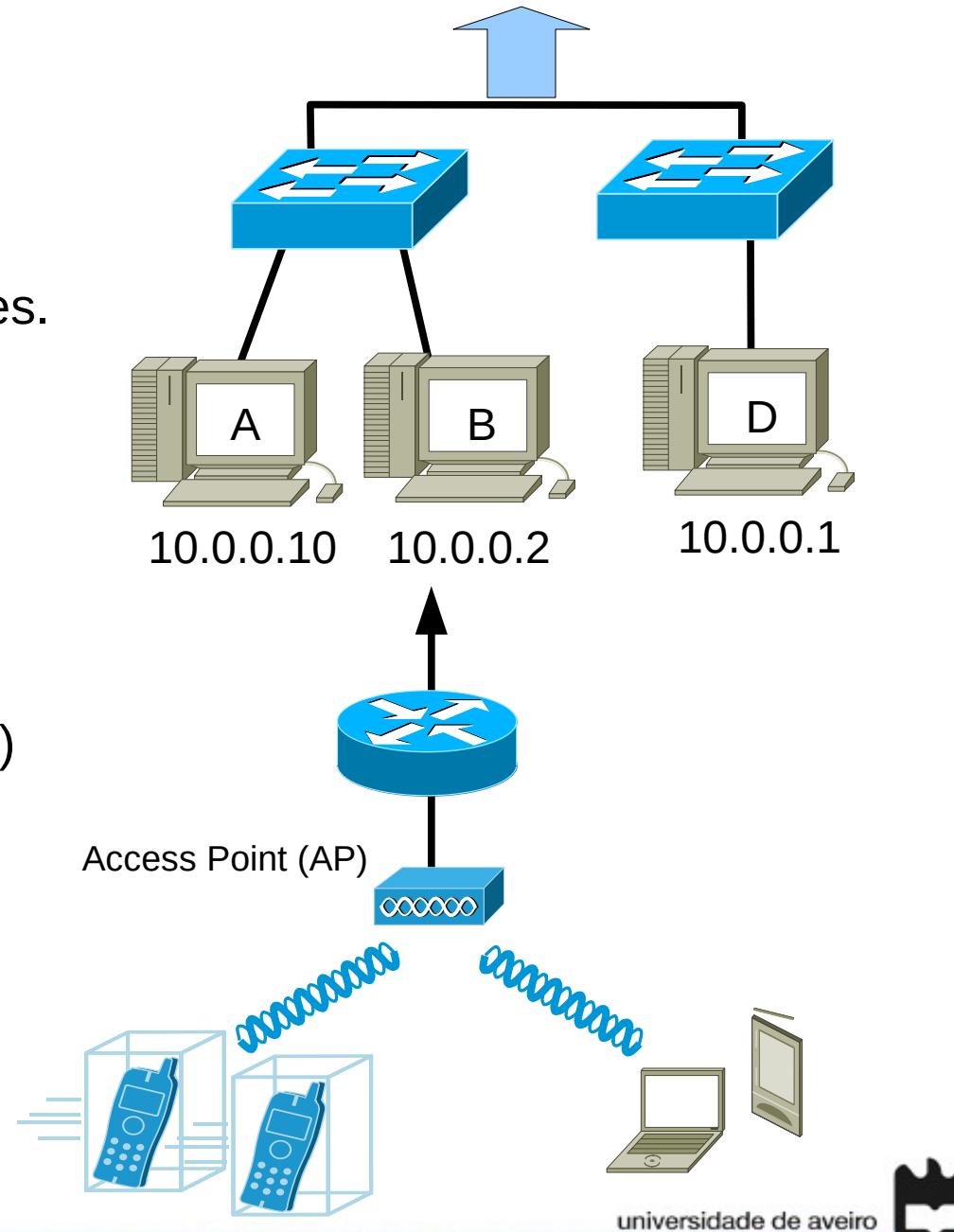
Mobile Access Network

- Provided by an ISP.
- Technologies:
 - ◆ LTE Advanced: até 1Gbps/500Mbps
 - ◆ LTE: até 100Mbps/50Mbps
 - ◆ WiMax: até 128Mbps/56Mbps
 - ◆ 3G HSPA+: até 42Mbps/11Mbps
 - ◆ 3G HSUPA: upload até 5.7Mbps
 - ◆ 3G HSDPA: download até 14.4Mbps
 - ◆ 3G UMTS: até 384kbps/384kbps
 - ◆ WAP/GPRS na Europa: até 114kbps



Local Area Network (LAN)

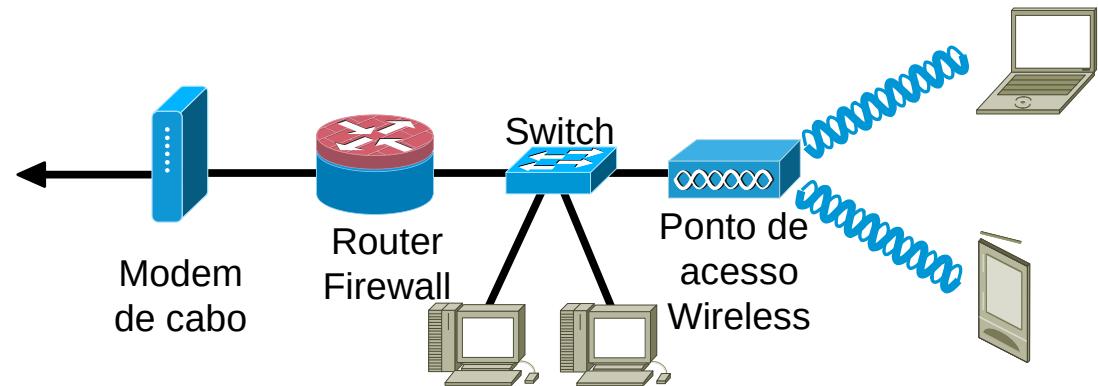
- Commonly implemented using:
 - ◆ Ethernet
 - ◆ Cabled technology.
 - ◆ Hosts interconnected using switches.
 - ◆ Wi-Fi (802.11)
 - ◆ Wireless technology
 - ◆ Hosts connect to Access Points (AP)
 - ◆ Versions:
 - 802.11b (WiFi 1): 11 Mbps
 - 802.11n (WiFi 2): 11 Mbps
 - 802.11g (Wi-Fi 3): 54 Mbps
 - 802.11n (Wi-Fi 4): ~300 Mbps
 - 802.11ac (Wi-Fi 5): ~1Gbps
 - 802.11ax (Wi-Fi 6): > 1Gbps



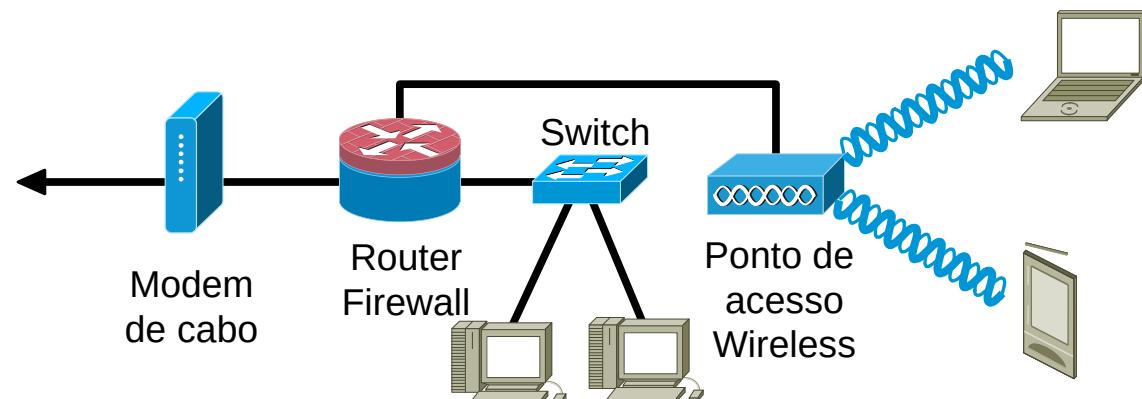
Residential Access Networks: Home LAN

Modem

- Fibre, 4G/5G Mobile network, CATV, ADSL, ...
- Router/firewall/NAT
- (Switched) Ethernet
- Wireless network (Wi-Fi)
 - Wireless access point.



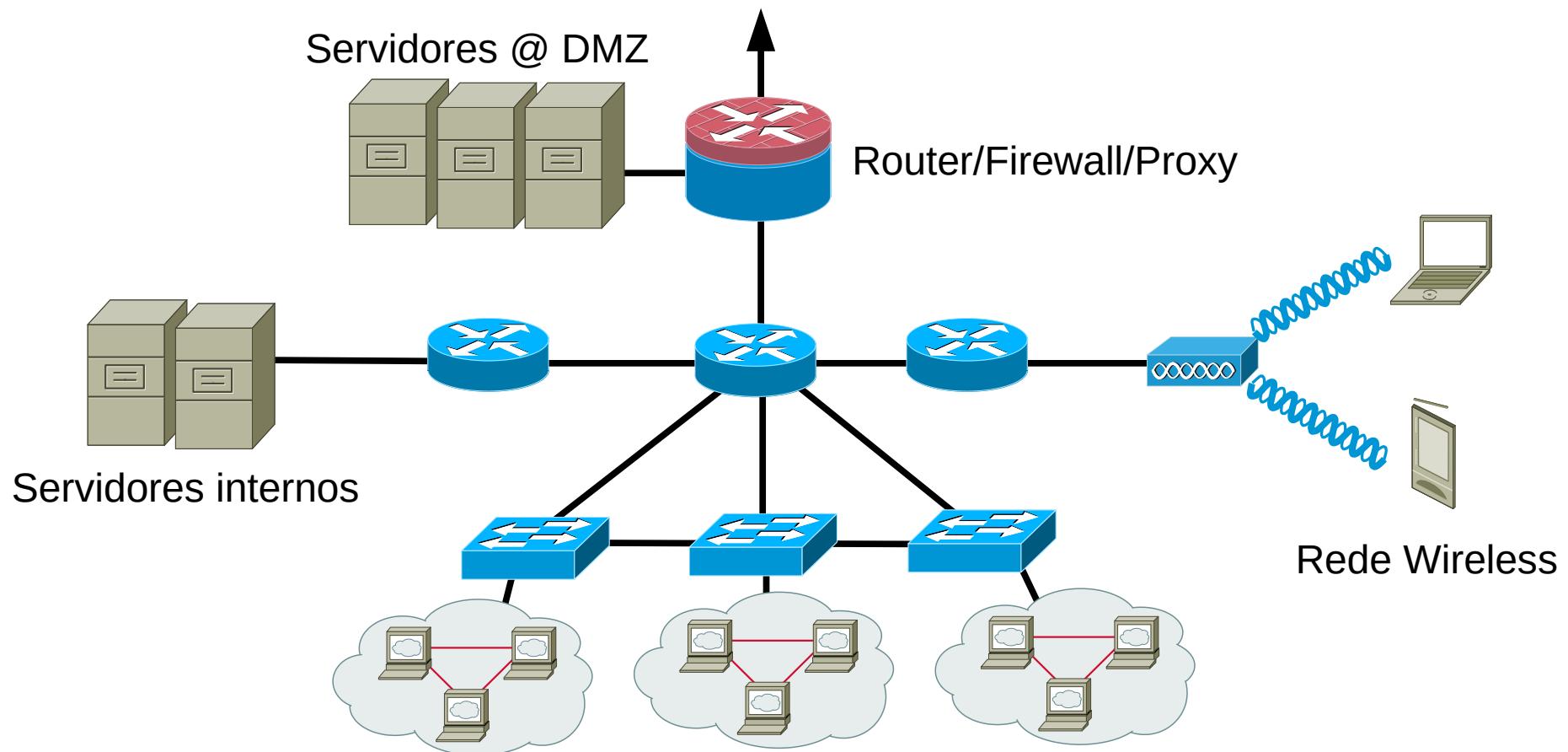
Ethernet (cabled) and Wireless networks
on the same IP network



Ethernet (cabled) and Wireless networks
on different IP networks

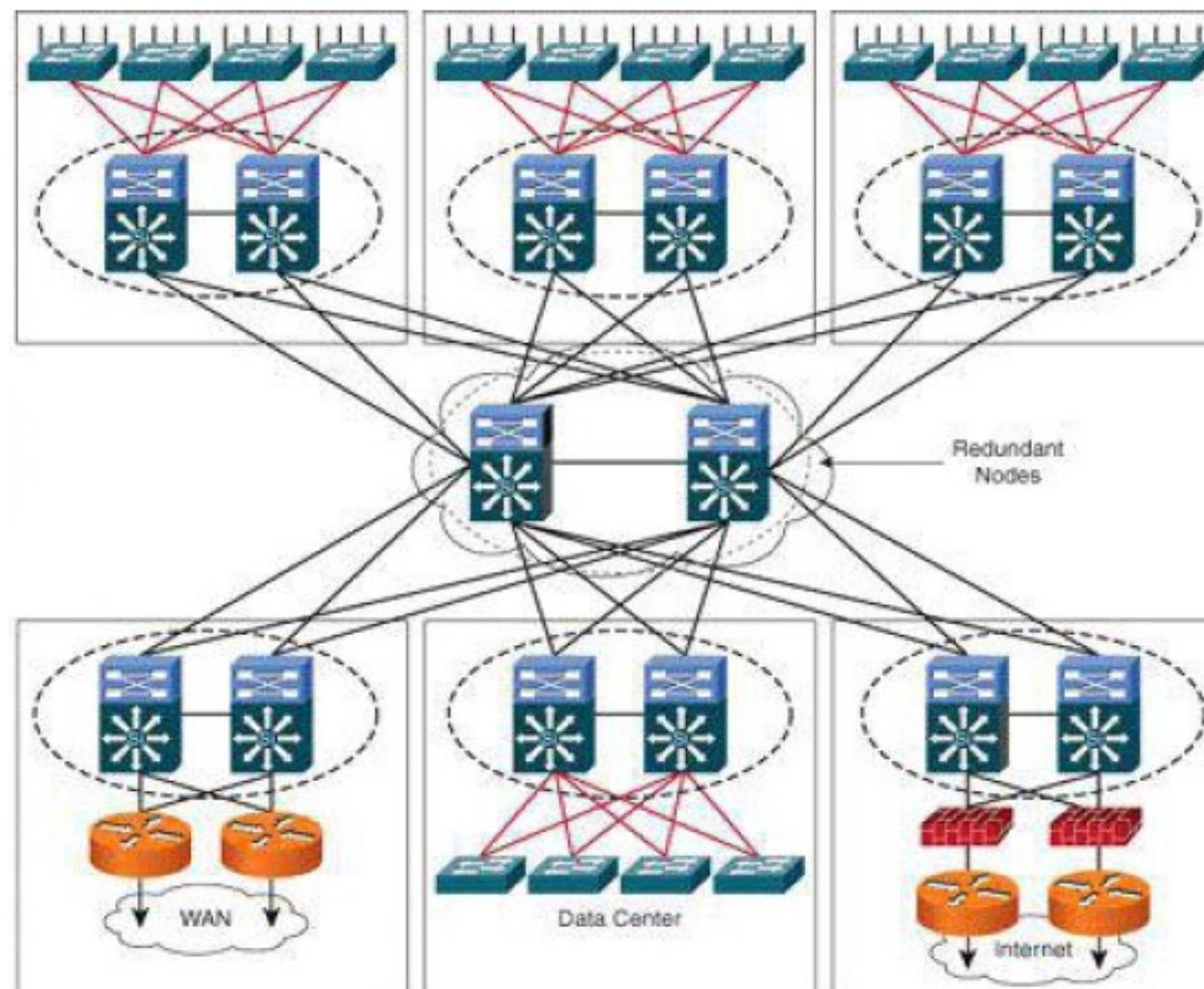


Corporate Access Networks: Small LAN



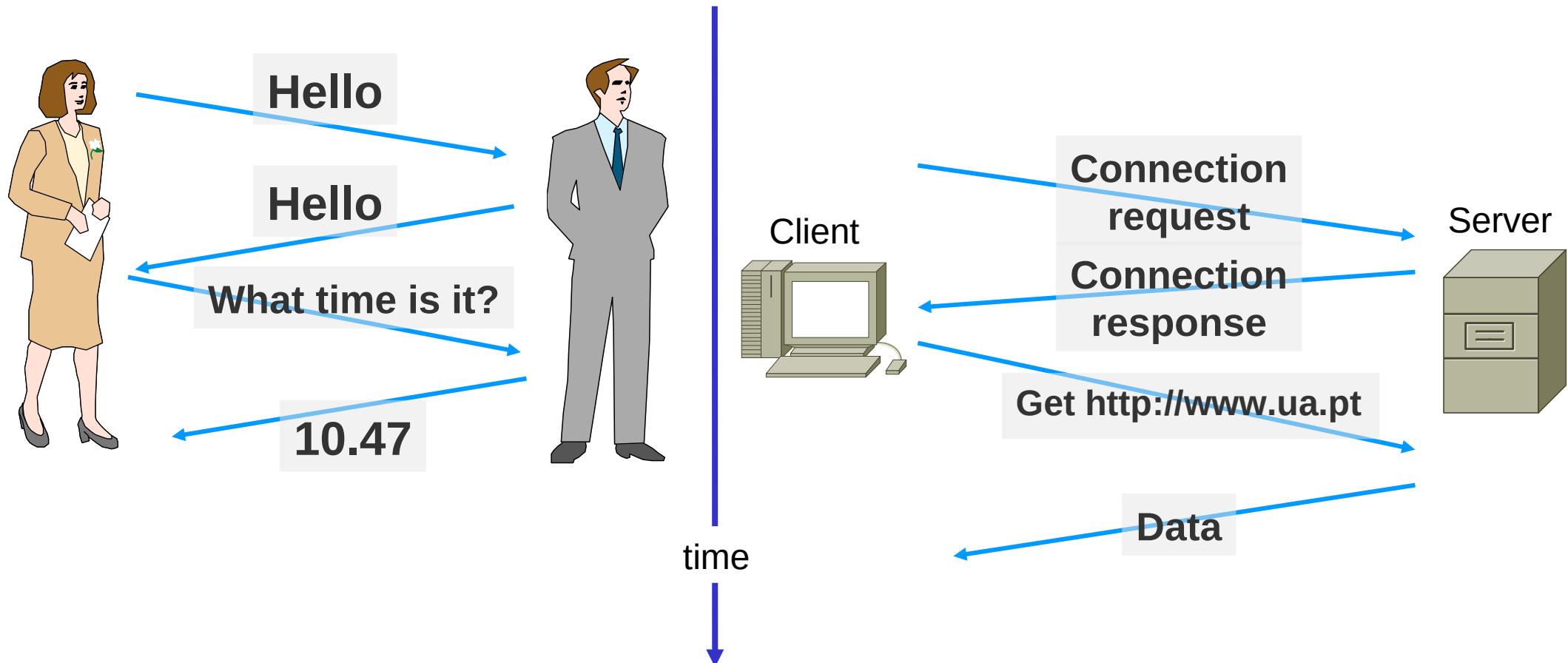
Corporate Access Networks: Medium/Large LAN

- Hierarchical architecture



What's a Protocol?

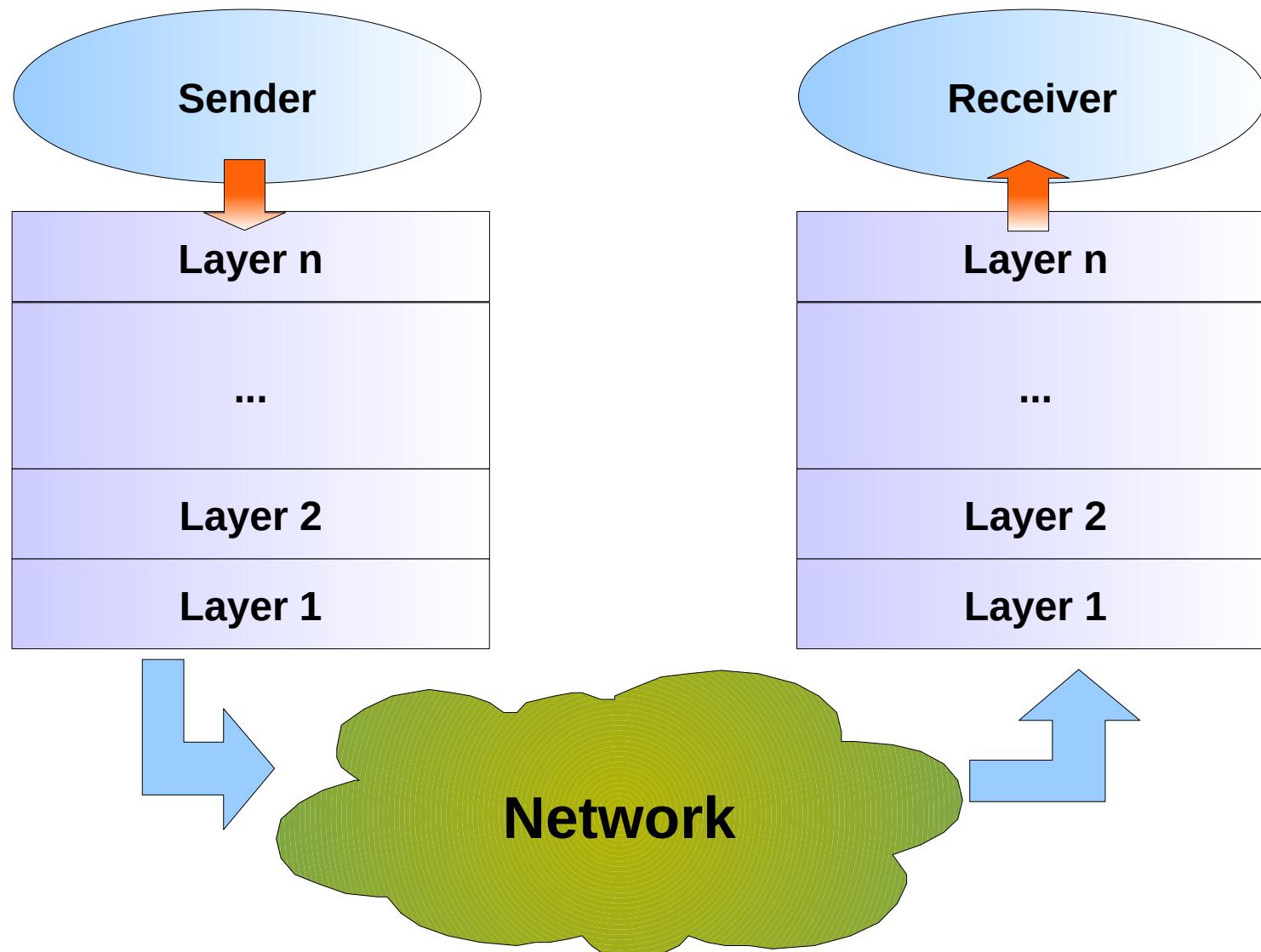
- Human protocol vs. Network protocol



- Network protocols define:
 - ◆ Format and order of the messages,
 - ◆ Actions to execute on sending, receiving and relaying messages.



Functionalities are Organized in Layers

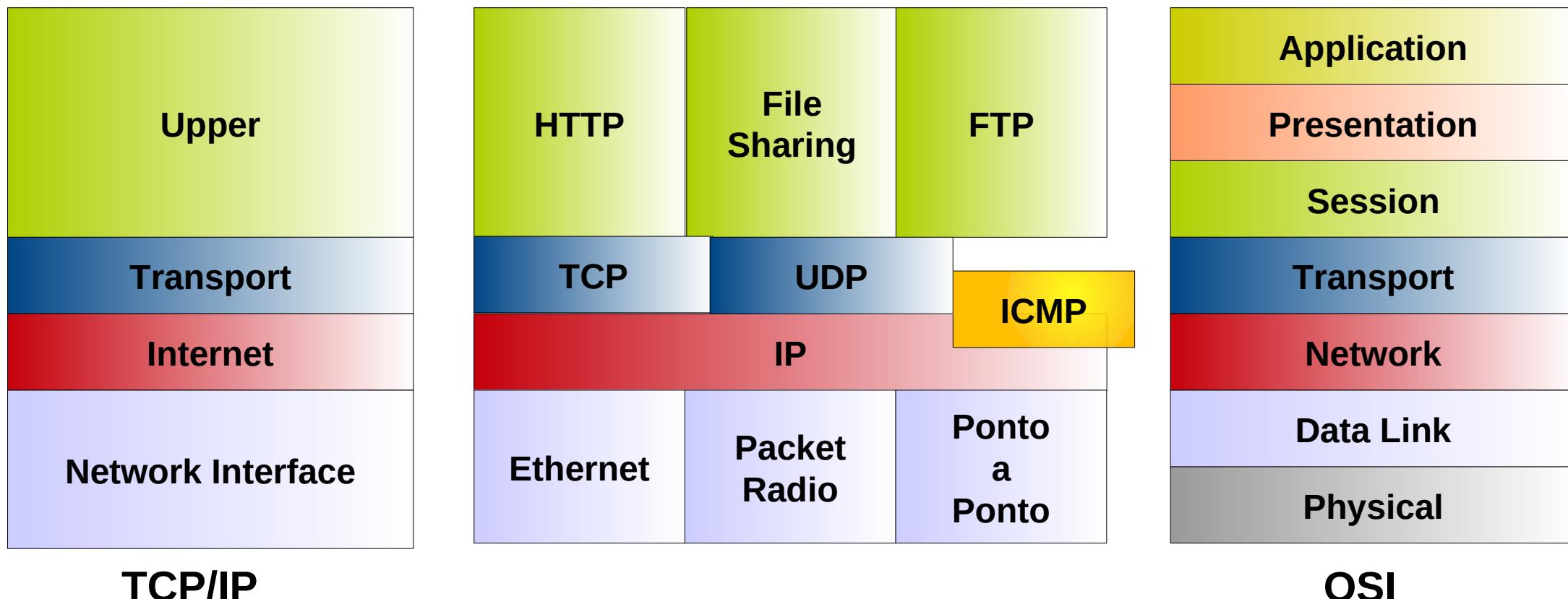


Modelo OSI (Open Systems Interconnection)

| | | |
|---------|--------------|--|
| Layer 7 | Application | Application/Service |
| Layer 6 | Presentation | Definition, manipulation and encoding of information |
| Layer 5 | Session | Establishing and maintaining sessions |
| Layer 4 | Transport | End-to-end communication |
| Layer 3 | Network | Addressing and routing |
| Layer 2 | Data Link | Local communication and medium sharing |
| Layer 1 | Physical | Physical signal transmission |

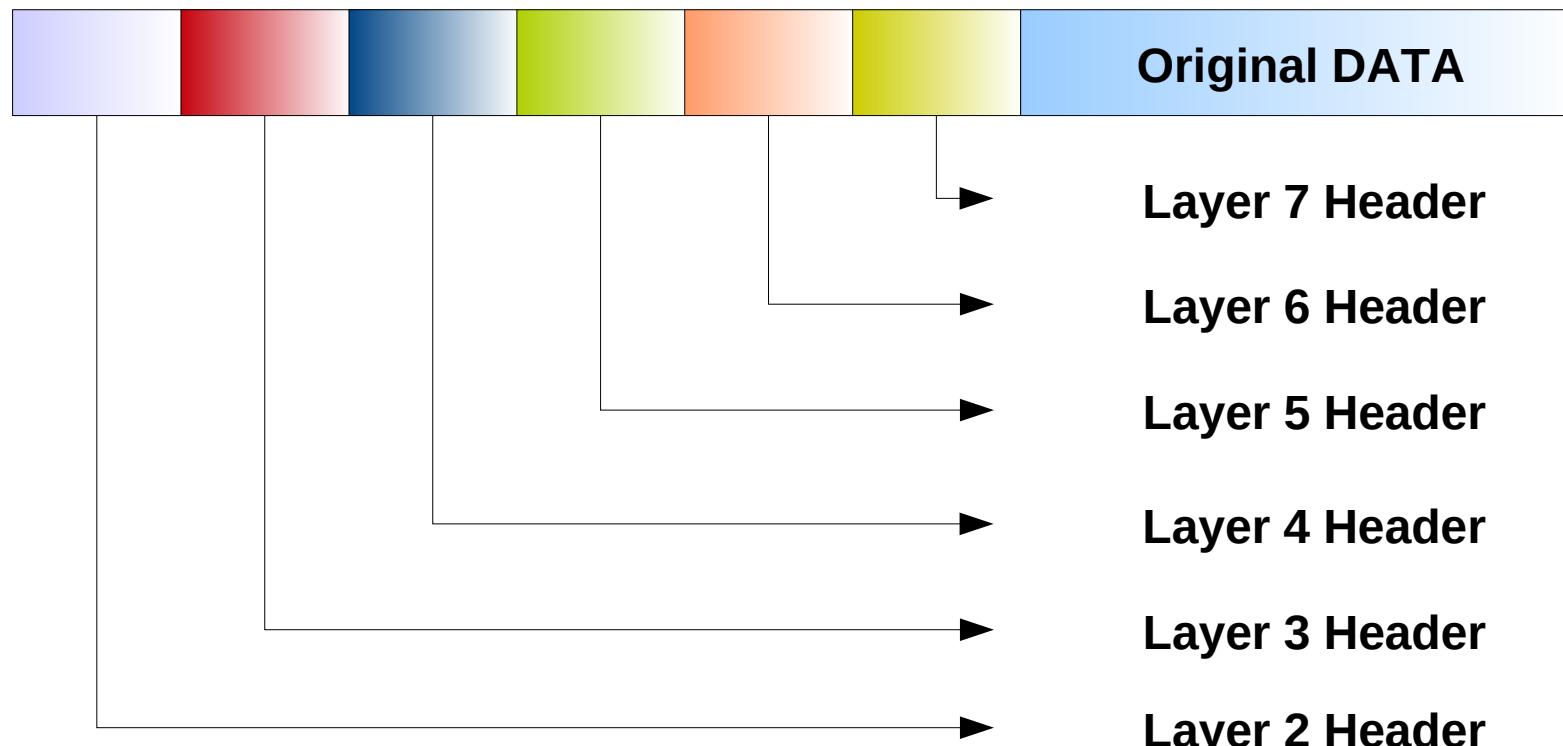


TCP/IP Reference Model



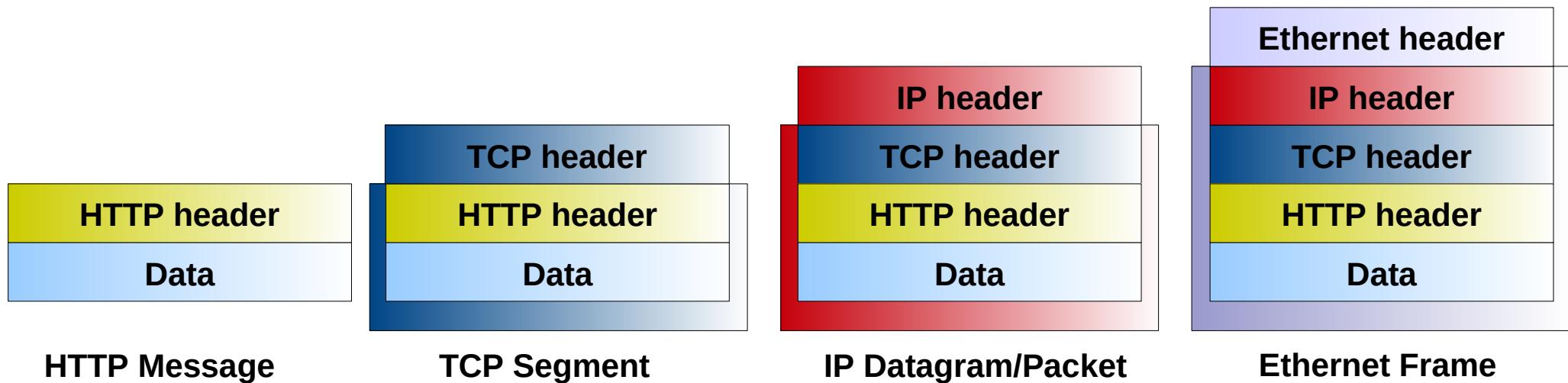
Header Concatenation

- Packets that travel through a network may have multiple concatenated headers
 - ◆ May be from protocols from all OSI layers.
 - ◆ May have more than one header from the same OSI layer.



Example

HTTP (HyperText Transfer Protocol)



HTTP Example with Wireshark

Wireshark screenshot showing an HTTP session between 192.168.91.102 and 193.136.92.50.

Selected frame 215 (HTTP response) details:

- Frame 215: 1837 bytes on wire (14696 bits), 1837 bytes captured (14696 bits) on interface enp0s20f0u1, id 0
- Ethernet II, Src: SuperMic_77:1e:ff (ac:1f:6b:77:1e:ff), Dst: RealtekS_68:06:bc (00:e0:4c:68:06:bc)
- Internet Protocol Version 4, Src: 193.136.92.50, Dst: 192.168.91.102
- Transmission Control Protocol, Src Port: 80, Dst Port: 43048, Seq: 2897, Ack: 440, Len: 1771
- [2 Reassembled TCP Segments (4667 bytes): #203(2896), #215(1771)]

Selected frame 215 (HTTP response) content:

```
HTTP/1.1 200 OK\r\nContent-Type: text/html\r\nServer: Microsoft-IIS/7.0\r\nX-Powered-By: PHP/5.3.6\r\nX-Powered-By: ASP.NET\r\nDate: Tue, 22 Sep 2020 16:47:08 GMT\r\nContent-Length: 4489\r\n\r\n[HTTP response 1/3]\n[Time since request: 0.018011071 seconds]\n[Request in frame: 136]\n[Next request in frame: 278]\n[Next response in frame: 280]\n[Request URI: http://www.av.it.pt/salvador/newHeaderSal.png]\nFile Data: 4489 bytes
```

Selected frame 215 (HTTP response) hex dump:

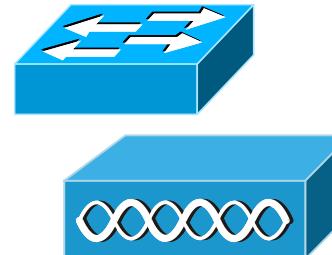
| | | |
|------|---|--------------------|
| 0000 | 48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f 4b 0d | HTTP/1.1 200 OK· |
| 0010 | 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 74 | ·Content -Type: t |
| 0020 | 65 78 74 2f 68 74 6d 6c 0d 0a 53 65 72 76 65 72 | ext/html ··Server |
| 0030 | 3a 20 4d 69 63 72 6f 73 6f 66 74 2d 49 49 53 2f | : Micros oft-IIS/ |
| 0040 | 37 2e 30 0d 0a 58 2d 50 6f 77 65 72 65 64 2d 42 | 7.0··X-P owered-B |
| 0050 | 79 3a 20 50 48 50 2f 35 2e 33 2e 36 0d 0a 58 2d | y: PHP/5 .3.6··X- |
| 0060 | 50 6f 77 65 72 65 64 2d 42 79 3a 20 41 53 50 2e | Powered- By: ASP. |
| 0070 | 4e 45 54 0d 0a 44 61 74 65 3a 20 54 75 65 2c 20 | NET··Dat e: Tue, |
| 0080 | 32 32 20 53 65 70 20 32 30 32 30 20 31 36 3a 34 | 22 Sep 2 020 16:4 |
| 0090 | 37 3a 30 38 20 47 4d 54 0d 0a 43 6f 6e 74 65 6e | 7:08 GMT ··Conten |
| 00a0 | 74 2d 4c 65 6e 67 74 68 3a 20 34 34 38 39 0d 0a | t-Lengt h : 4489·· |
| 00b0 | 0d 0a 3c 21 44 4f 43 54 59 50 45 20 68 74 6d 6c | ··<!DOCT YPE html |
| 00c0 | 3e 0a 3c 68 74 6d 6c 3e 0a 3c 68 65 61 64 3e 0a | >·<html> ·<head> |
| 00d0 | 3c 74 69 74 6c 65 3e 50 61 75 6c 6f 20 53 61 6c | <title>P aulo Sal |
| 00e0 | 76 61 64 6f 72 20 7c 20 50 65 72 73 6f 6e 61 6c | vador Personal |
| 00f0 | 20 48 6f 6d 65 20 50 61 67 65 3c 2f 74 69 74 6c | Home Pa ge</titl |
| 0100 | 65 3e 0a 3c 6d 65 74 61 20 68 74 74 70 2d 65 71 | e>·<meta http-ed |



Equipment Types

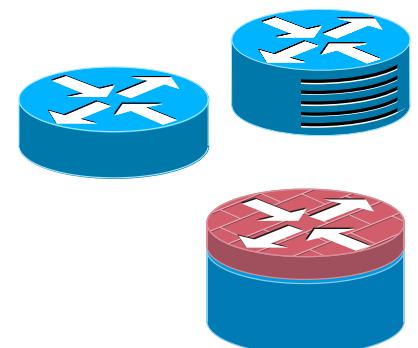
Switch

- ◆ OSI Layer 2 inter-connection,
- ◆ Implements VLAN,
- ◆ Forwarding based on Spanning-tree,
 - ◆ STP, RSTP, MSTP
- ◆ Wireless access points (AP).



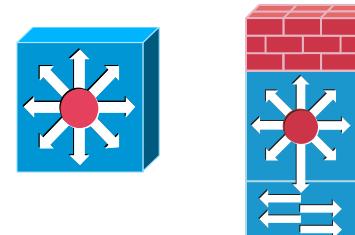
Router

- ◆ OSI Layer 3 inter-connection
- ◆ Has extra functionalities such as: QoS, Security, VPN gateway, monitoring, etc...



L3 Switch

- ◆ Switch+Router.
- ◆ Limited routing functionalities (lower/medium end models).
- ◆ Full routing functionalities (high end models).
- ◆ Many have Layer 3 dedicated hardware for switching.

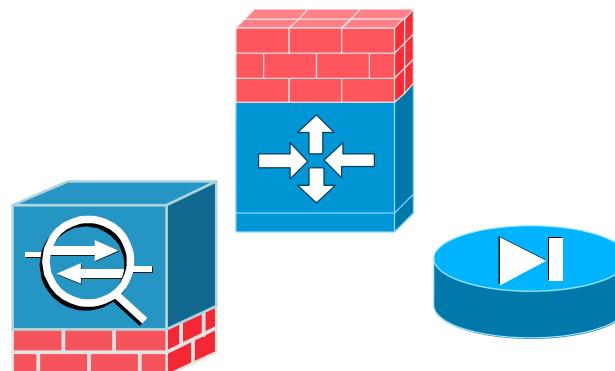


Router with switching modules

- ◆ Full Layer 3 functionalities
- ◆ Limited Layer 2 functionalities.

Security Appliance

- ◆ Firewall
- ◆ IDS/IPS (Intrusion Detection/Prevention System)
- ◆ NAT/PAT
- ◆ VPN Gateway



Introduction to Network Addresses

• Physical (MAC) in Ethernet and Wi-Fi

- ◆ MAC (Physical, Ethernet or LAN) Address:
- ◆ Function: Allow the exchange of data between network interfaces connected using a Layer 2 network.
- ◆ Have 6 bytes/48 bits.
- ◆ Are unique.
- ◆ Each network card has its own address.
- ◆ Defined by manufacturer
- ◆ Some hardware allows change.
- ◆ First 24-, 28-, or 36-bits assign to manufacturer.

Hexadecimal notation

Broadcast: FF-FF-FF-FF-FF-FF

```
3: wlp59s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN  
    link/ether 6e:c2:ce:46:d1:9b brd ff:ff:ff:ff:ff:ff permaddr 9c:b6:d0:c1:c0:  
59: enp0s20f0u1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP  
    link/ether 00:e0:4c:68:06:bc brd ff:ff:ff:ff:ff:ff  
    inet 192.168.91.102/25 brd 192.168.91.127 scope global dynamic noprefixroute  
      valid_lft 38824sec preferred_lft 38824sec  
    inet6 fe80::6583:5450:64c3:94a3/64 scope link noprefixroute  
      valid_lft forever preferred_lft forever
```

• IPv4

- ◆ 4 bytes = 32 bits
- ◆ 4 decimal numbers separated by dots (.)
 - ◆ e.g.: 10.0.0.1, 192.156.1.4, 253.1.3.7
 - ◆ 1 byte: 0 to 255

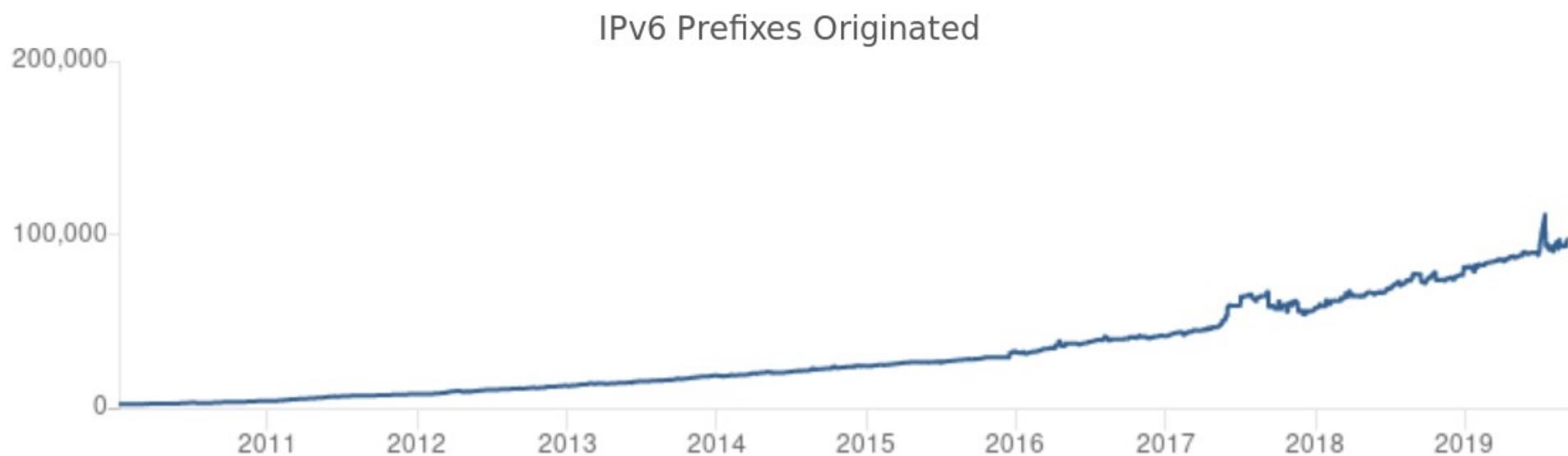
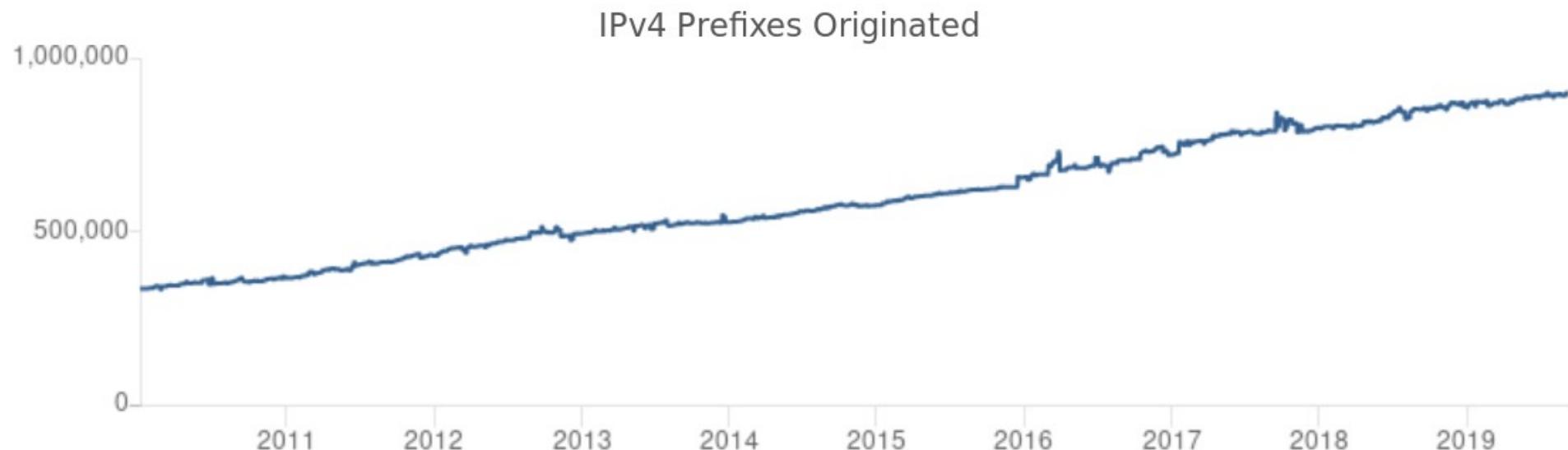
• IPv6

- ◆ 16 bytes = 128 bits
- ◆ 8 groups of 4 hexadecimal digits separated by colons (:)
 - ◆ Can be simplified, by merging sequential zeros!
 - ◆ e.g.: 2001:ABCD:1346:0011:ABFE:3478:A4B5:CC10
 - ◆ 2 hexadecimal digits represent a byte: 00 to FF

```
Ethernet adapter Ethernet:  
  
Connection-specific DNS Suffix . :  
Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter  
Physical Address . . . . . : 08-00-27-F7-59-07  
DHCP Enabled. . . . . : Yes  
Autoconfiguration Enabled . . . . . : Yes  
Link-local IPv6 Address . . . . . : fe80::c84:e2a0:88ad:a538%10(Preferred)  
IPv4 Address. . . . . : 10.0.2.15(Preferred)  
Subnet Mask . . . . . : 255.255.255.0  
Lease Obtained. . . . . : Tuesday, September 15, 2020 2:56:00 PM  
Lease Expires . . . . . : Wednesday, September 23, 2020 4:28:24 PM  
Default Gateway . . . . . : 10.0.2.2  
DHCP Server . . . . . : 10.0.2.2  
DHCPv6 IAID . . . . . : 34078759  
DHCPv6 Client DUID. . . . . : 00-01-00-01-22-0B-6E-F8-08-00-27-F7-59-07  
DNS Servers . . . . . : 193.136.92.73  
                                         193.136.92.74  
NetBIOS over Tcpip. . . . . : Enabled
```



IPv4/IPv6 Prefixes

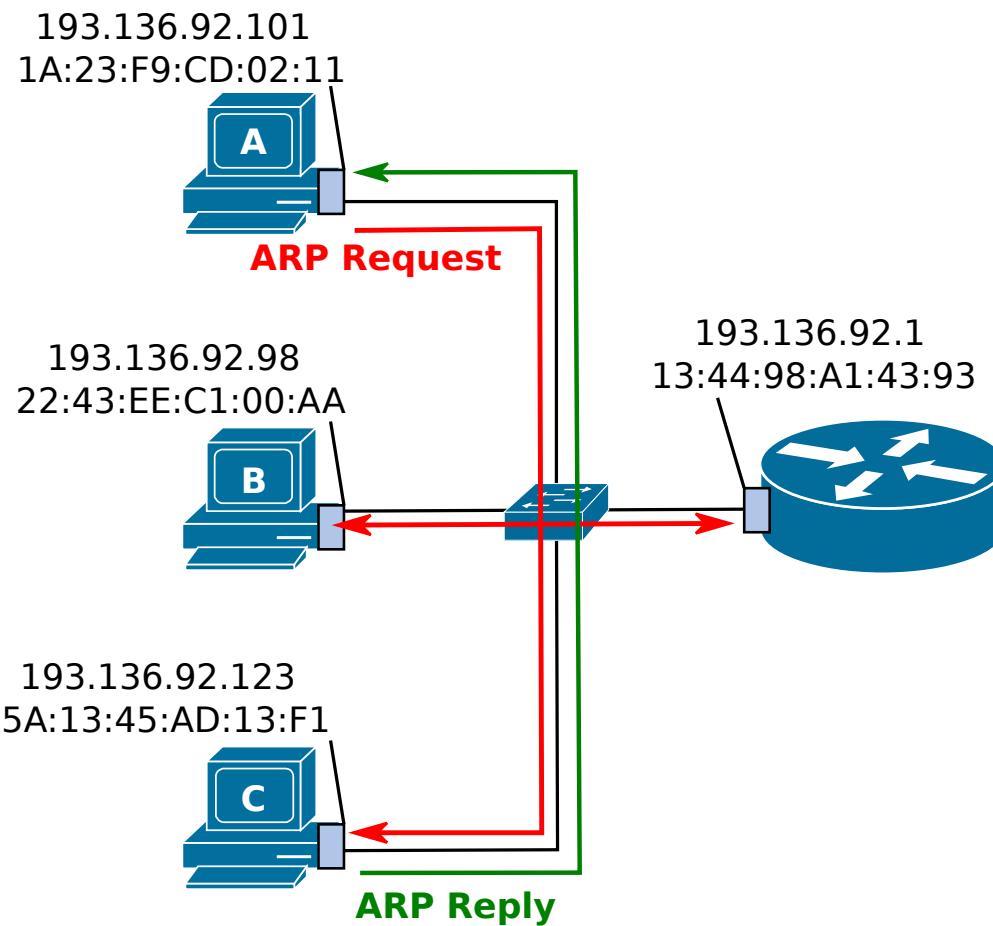


Updated 10 Sep 2019 20:45 PST © 2019 Hurricane Electric



Resolution of Physical Addresses

- IPv4: Address Resolution Protocol (ARP)
- IPv6: Neighbor Discovery (ICMPv6)
- ARP Example:
 - ◆ When “A” wants to contact “C” by IPv4:
 - ◆ “A” requires “C” MAC address.
 - ◆ Only knows IPv4 address.
 - ◆ If “C” IPv4 address is not present in the ARP table, then:
 - “A” send an “ARP Request” in broadcast to the local network (destination MAC: FF:FF:FF:FF:FF:FF) with the IPv4 address of “C”,
 - All machines receive this packet,
 - “C” verifies that is IPv4 address is on the the “ARP request”, responds directly to “A” with a “ARP reply” (destination MAC==MAC of “A”) with it’s own MAC address.
 - ◆ MAC address resolution only happens in the local network.
 - ◆ ARP packets do not pass through routers.



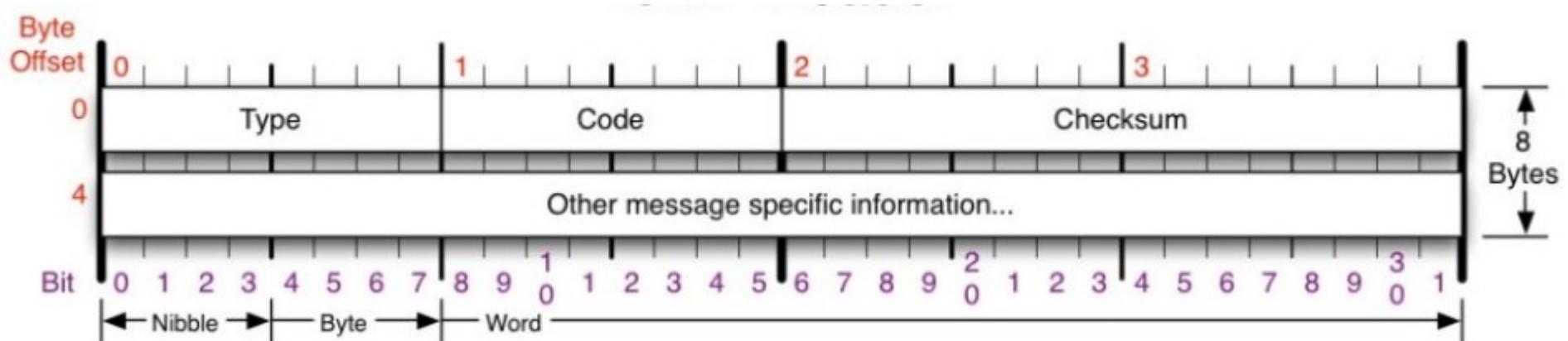
Hosts Connectivity

- Two hosts are considered connected if both can send packets to the other, and the packets are correctly received.
 - ◆ This is called **Full Connectivity**.
 - ◆ It is possible to measure the quality of the connectivity:
 - ◆ Measuring the number of **lost packets**,
 - ◆ Measuring **Round Trip Time (RTT)**.
 - Time it takes for a network packet to go from a starting point to a destination and back again to the starting point.
 - Does not require clock synchronization
 - ◆ Lack of full connectivity may be caused by routing problems, lack of connections between sender and receiver, and/or security constraints.
 - When connectivity is only achieved in one direction (not both ways), this is called partial connectivity.

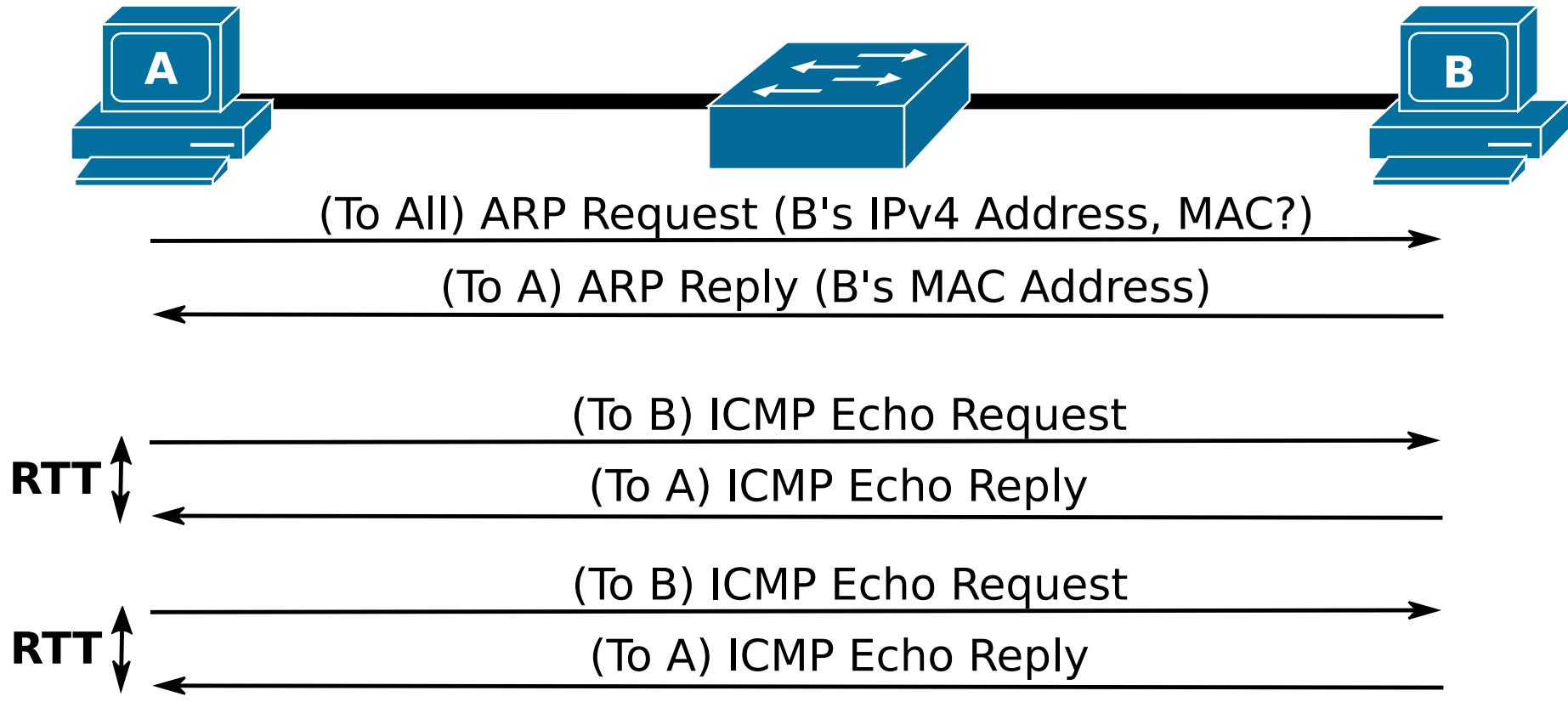


Internet Control Message Protocol (ICMP)

- Used to notify events and perform network operations
 - Notification of unreachable network,
 - Notification of unavailable UDP ports,
 - Routing redirection,
 - Connectivity tests and path identification,
 - Etc...
- Header with a fixed size of 8 bytes:



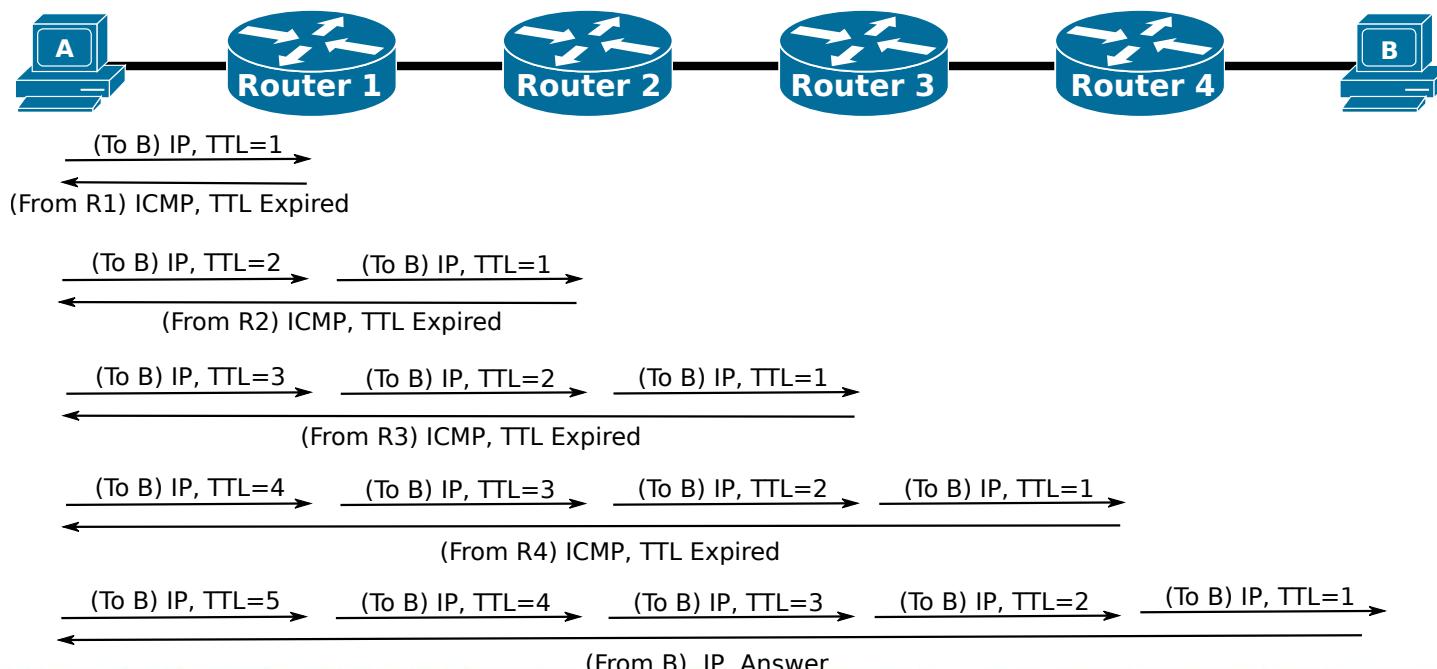
IPv4 Connectivity Test with PING



RTT: Round Trip Time

End-to-End Path Identification

- Usually called “Trace Route” or “Trace Path”
- Rely on the usage of the IP TTL header field
 - ◆ Uses ICMP, UDP or TCP packets
- TTL is reduced by one when reaches a router
 - ◆ If reaches 0 the packet is discarded,
 - ◆ And, the router notifies the sender with a ICMP “TTL expired in transit” message.
- Sender starts with TTL equal to 1, and progressively increases the sent TTL, until it reaches the destination.
 - ◆ TTL=1 → packet “expires” in first router on path, sender discovers first router.
 - ◆ TTL=2 → packet “expires” in second router on path, sender discovers second router.
 - ◆ And so on... until the sender receives an answer from the destination.



Local Area Networks (LAN)

Introduction to Switching, IPv4 and Routing

Fundamentos de Redes

Mestrado Integrado em
Engenharia de Computadores e Telemática
DETI-UA

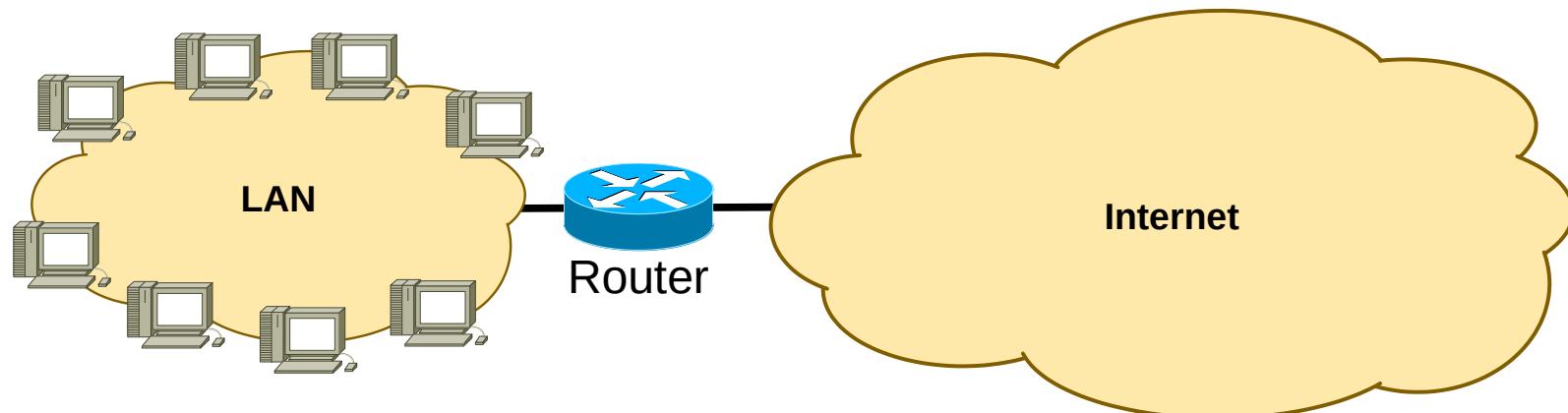


universidade de aveiro

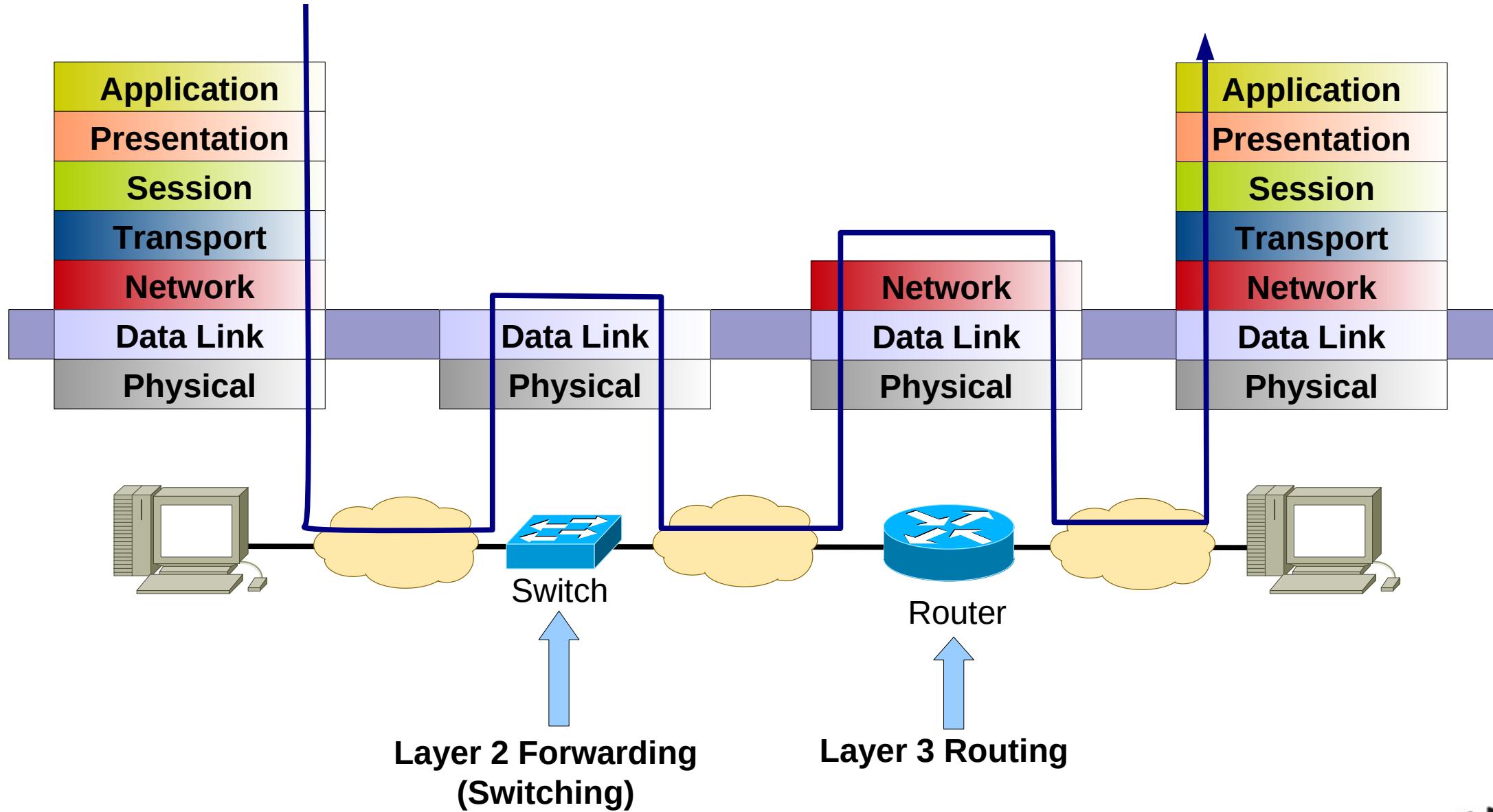
deti.ua.pt

Local Area Network (LAN)

- Is a computer network within a small geographical area.
 - ◆ Home, school, room, office building or group of buildings.
- Is composed of inter-connected hosts capable of accessing and sharing data, network resources and Internet access.
 - ◆ Host refers generically to a PC, server, or any other terminal.
- Technologies
 - ◆ Current: Ethernet, 802.11 (Wi-Fi)
 - Legacy: Token Ring, FDDI, ...

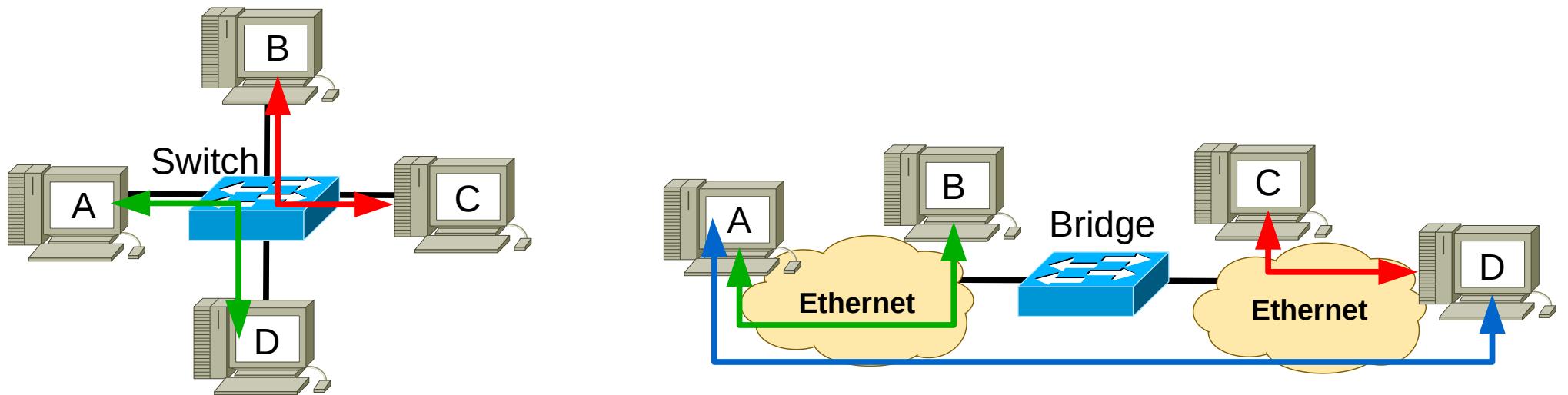


Local Area Network (LAN)

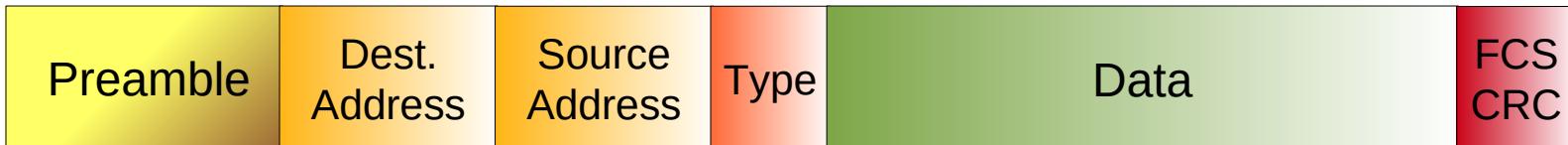


Switching

- With Switches/Bridges
 - Interconnection done at OSI Layer 2.
 - Hosts can transmit simultaneously.
 - A network of Switches is a **Broadcast Domain**
 - An Ethernet frame with destination FF:FF:FF:FF:FF:FF (Broadcast) will reach all connected switches and hosts.



Ethernet Frame

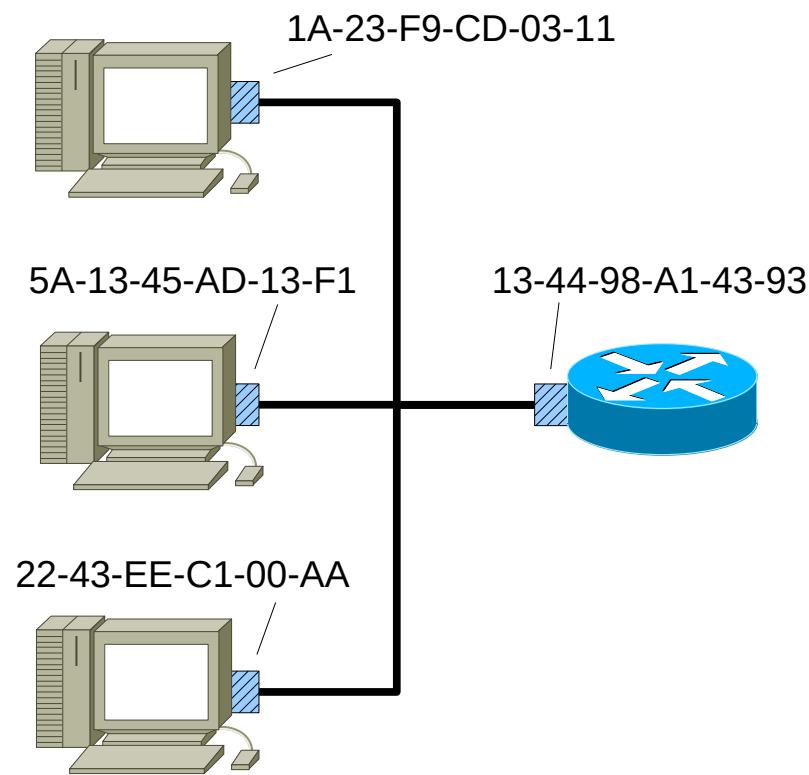


- The sender's network card encapsulates an IP datagrama (or any other network protocol) in an Ethernet frame.
- Preamble:
 - ◆ 7 bytes with pattern 10101010 followed by one byte with pattern10101011.
 - ◆ Used to sincronize the sending and receiving clocks.
- Destination and Source addresses: 6 bytes Physical (MAC) address
 - ◆ If the network card receives a frame with destination equal to its own address or its the broadcast address, it will pass data to the network level process.
 - ◆ If not, drops the frame.
- Type defines which protocol is encapsulated in the frame (usually IPv4 or IPV6).
- The frame check sequence (FCS) is a four-octet cyclic redundancy check (CRC) that allows detection of corrupted data within the entire frame as received on the receiver side.



MAC Addresses

- MAC (Physical, Ethernet or LAN) Address:
 - ◆ Function: Allow the exchange of data between network interfaces connected using a Layer 2 network.
 - ◆ Have 6 bytes/48 bits.
 - ◆ Are unique.
 - ◆ Each network card has its own address.
 - ◆ Defined by manufacturer
 - ◆ Some hardware allows change.
 - ◆ First 24-, 28-, or 36-bits assign to manufacturer.
 - ◆ Hexadecimal notation
 - ◆ Broadcast: FF-FF-FF-FF-FF-FF



Ethernet Frame Minimum Size

- Historically there were Ethernet technologies that allowed collisions and a collision detection mechanism had to be present (CSMA/CD).
- Depending on the technology and maximum cable size, the Ethernet frame had to be big enough to allow the collision detection mechanism to detect a frame being transmitted before the last frame byte leaving the source host.
- By legacy (it is possible to merge different Ethernet technologies) the **minimum frame size is 64 bytes**.
- If the frame's header plus data do not reach 64 bytes, a set of zeros must be added to the end of the frame to reach 64 bytes.
 - ◆ This is called **padding**.



Switches Basic Operations

- Switches have a **Forwarding Table**.
- When a switch receives an Ethernet frame:
 - ◆ Registers an entry at the Forwarding Table the frame's source MAC address and the port where the frame was received.
 - If no frames are received from that MAC address after some time (**aging time**) the entry is removed.
 - ◆ Searches the Forwarding Table for the frame's destination MAC address and forwards the packet according:
 - **Forwarding** mechanism:
 - If the frame's destination MAC address exists in the table, the switches forwards the frame through the port associated with that MAC address.
 - **Flooding** mechanism:
 - If the frame's destination MAC address DOES NOT exist in the table, the switches forwards the frame through all active ports (except the one where it was received).
 - » Note: Just within the same VLAN (more details later).

| MAC | Porta |
|-------------------|-------|
| 00:11:11:11:11:11 | 1 |
| 00:22:22:22:22:22 | 1 |
| A1:33:33:33:33:33 | 2 |
| 44:44:44:44:44:44 | 3 |
| 55:55:55:00:00:55 | 3 |



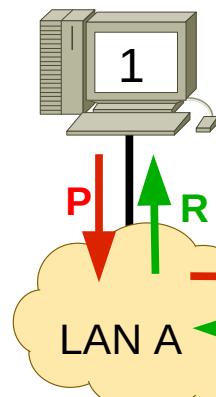
Learning, Flooding and Forwarding

Frame P

Dest. = MAC2 Source = MAC1

Frame R (Answer to P)

Dest. = MAC1 Source = MAC2



| Forwarding Table | |
|------------------|--|
| MAC1 – Port 1 | |
| MAC2 – Port 2 | |

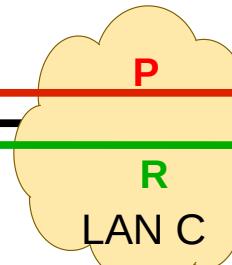
Switch 1

2

1

3

- (1). Switch 1 learns host 1 MAC address in port 1, from frame P source address (learning).
- (2). Switch 1 does not have frame's P destination (MAC 2) in the table, sends frame P to all ports except port 1 (flooding).



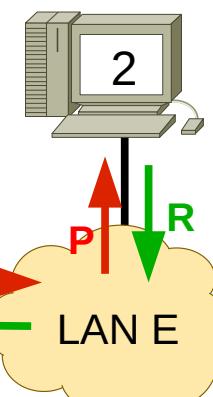
| Forwarding Table | |
|------------------|--|
| MAC1 – Port 1 | |
| MAC2 – Port 3 | |

Switch 2

3

2

1



- (5). Switch 2 learns host 2 MAC address in port 3, from frame R source address (learning).
- (6). Switch 2 have frame's R destination (MAC 1) in the table, sends frame R to port port 1 (forwarding).

- (7). Switch 1 learns host 2 MAC address in port 2, from frame R source address (learning).
- (8). Switch 2 have frame's R destination (MAC 1) in the table, sends frame R to port port 1 (forwarding).

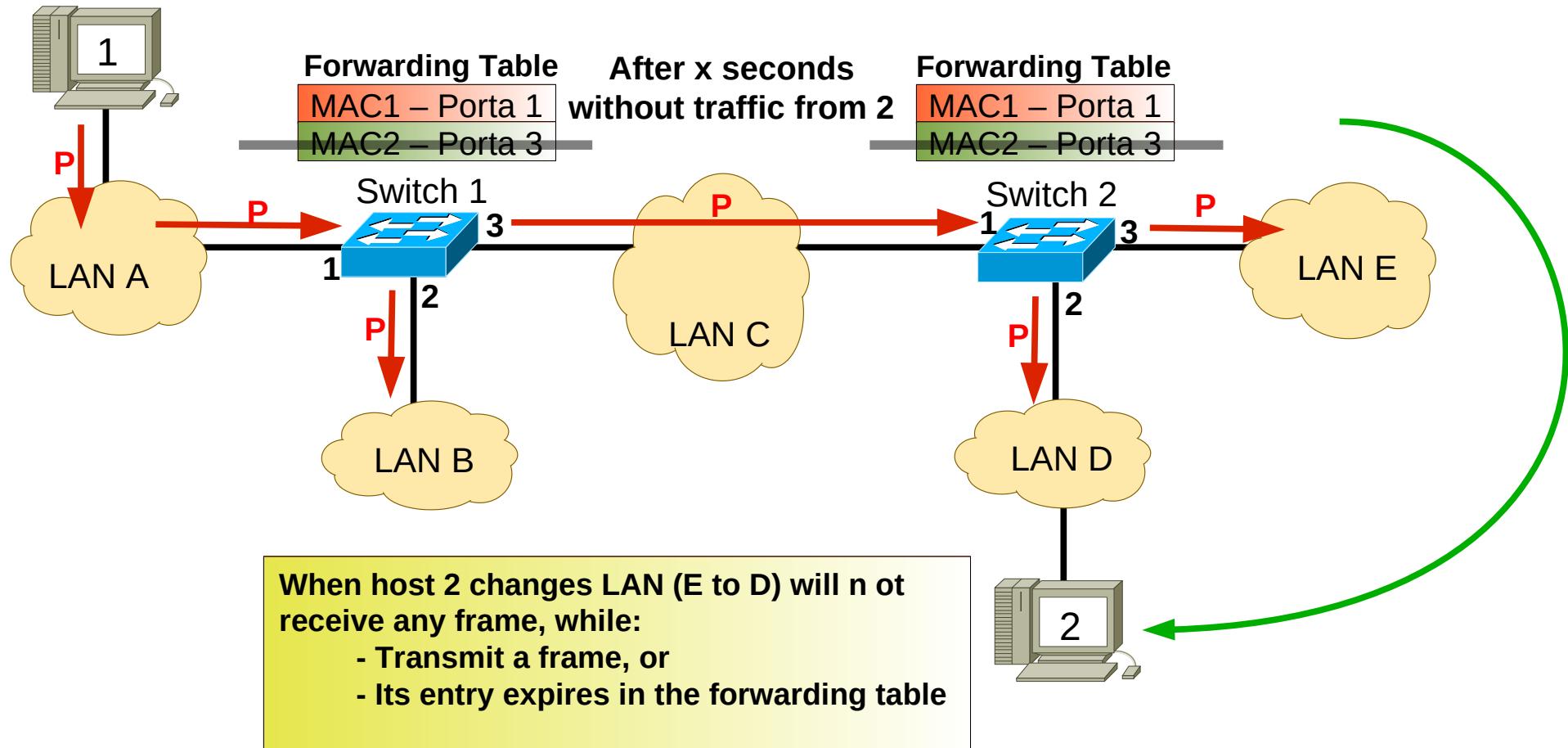
- (3). Switch 2 learns host 1 MAC address in port 1, from frame P source address (learning).
- (4). Switch 2 does not have frame's P destination (MAC 2) in the table, sends frame P to all ports except port 1 (flooding).



Forwarding Table Aging Time

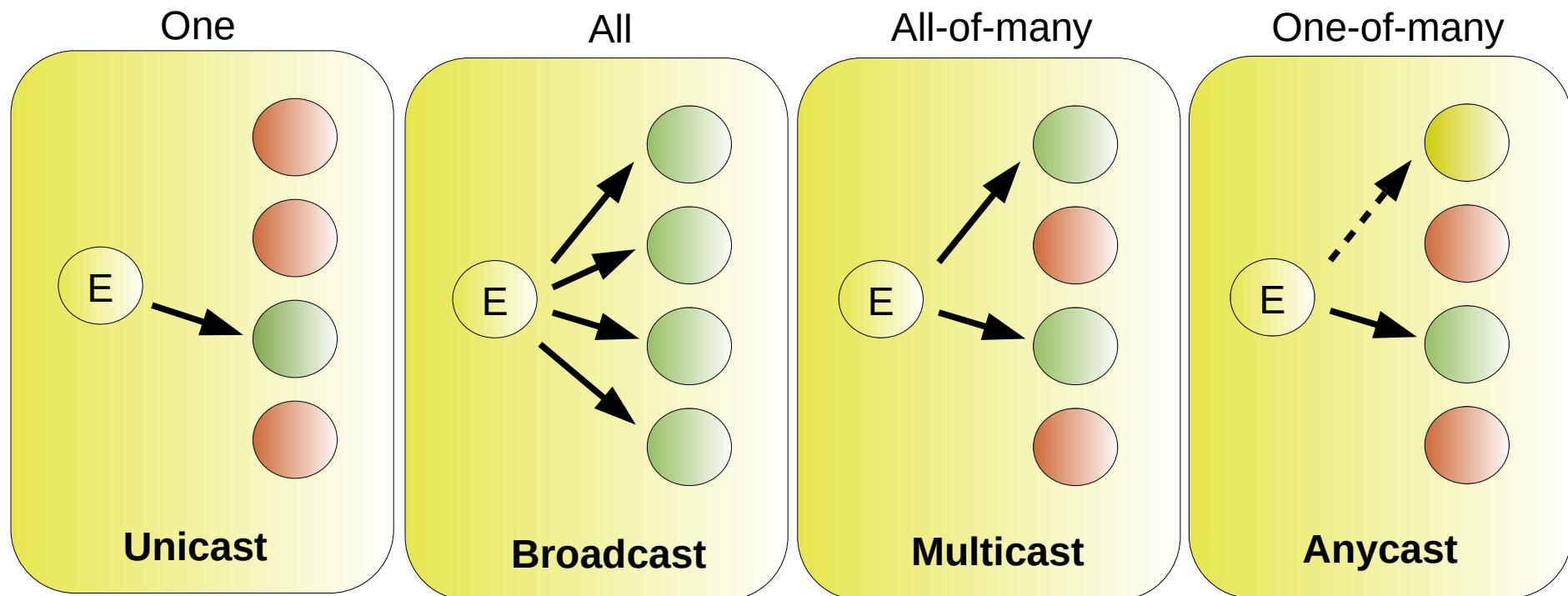
Frame P

Dest. = MAC2 | Source = MAC1



Types of Addresses

- Unicast – Identify a single sender/receiver.
- Broadcast – All are receivers.
- Multicast – Identify all elements of a group as receivers (all-of-many)
- Anycast – Identifies any element of group as receiver (one-of-many)



IPv4 Addressing

- An IPv4 address is a unique address for a network interface
- Exceptions:
 - ◆ Dynamically assigned IPv4 addresses (DHCP)
 - ◆ IP addresses in private networks (NAT)
- An IPv4 address:
 - ◆ is a **32 bit long** identifier
 - ◆ encodes a network number (**network prefix**)
and a **host identifier**



Network Prefix and Host Identifier

- The network prefix identifies a network and the host identifier identifies a specific host (actually, interface on the network).

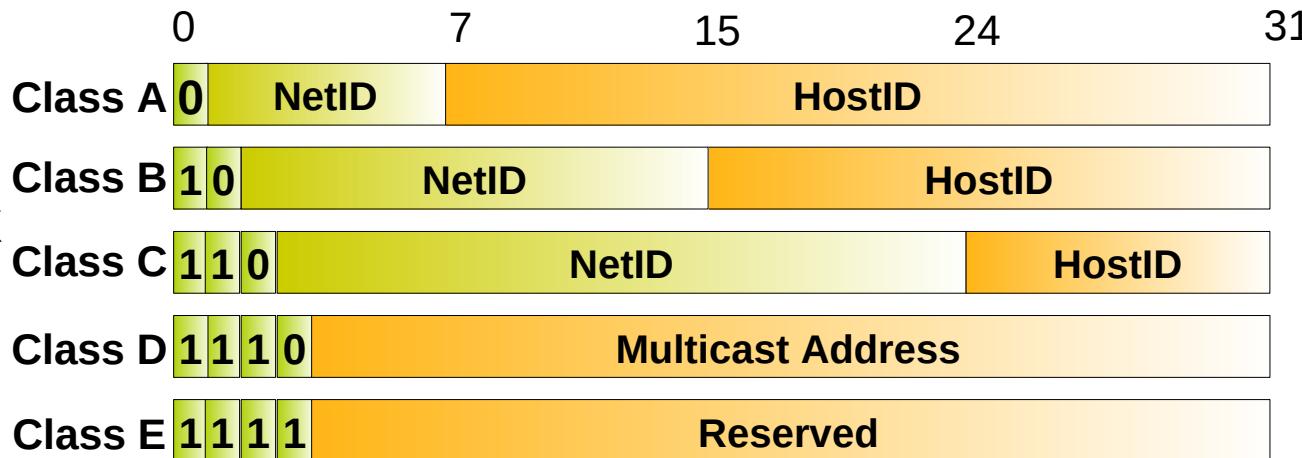


- How do we know how long the network prefix is?
 - ◆ **Before 1993:** The boundary between network prefix and host identifier is implicitly defined (**class-based/classful addressing**)
or
 - ◆ **After 1993:** The boundary between network prefix and host identifier is indicated by a **netmask**.



IPv4 Classful Addressing

- Initially (until 1993) the boundary between the network prefix and host identifier was predefined by the value of the first byte (class).
- Resulted in a huge waste of addresses:
 - Classes A and B were too big,
 - Not enough class C networks.
- Routing Tables were becoming very long
 - It was not possible to merge (aggregate) networks to simplify routing tables.



| Class | First Address | Last Address |
|-------|---------------|-----------------|
| A | 1.0.0.0 | 126.0.0.0 |
| B | 128.0.0.0 | 191.255.0.0 |
| C | 192.0.0.0 | 223.255.255.0 |
| D | 224.0.0.0 | 239.255.255.255 |
| E | 240.0.0.0 | 255.255.255.254 |



Classless Inter-Domain Routing (CIDR)

- New interpretation of the IP addressing to increase efficiency and flexibility.
 - ◆ Network Masks were created to define the boundary between the IP network prefix and host identifier.
 - ◆ A bit of the mask equal to one indicate that that bit (in that position) of the address belongs to the network prefix.
 - ◆ A bit of the mask equal to zero indicate that that bit (in that position) of the address belongs to the host identifier.
 - ◆ Called VLSM (Variable Length Subnet Mask).
 - ◆ Must be provided with the IP address.
- Allowed the partition of a network in smaller networks or sub-networks (subnets).
- Allowed to merge several network under a single prefix (aggregation or summary process).

| | decimal | binary |
|--------------|---------------|-------------------------------------|
| IPv4 Address | 193.136.92.1 | 11000001.10001000.01011100.00000001 |
| Mask | 255.255.255.0 | 11111111.11111111.11111111.00000000 |

← → ← →

network prefix host identifier network prefix host identifier



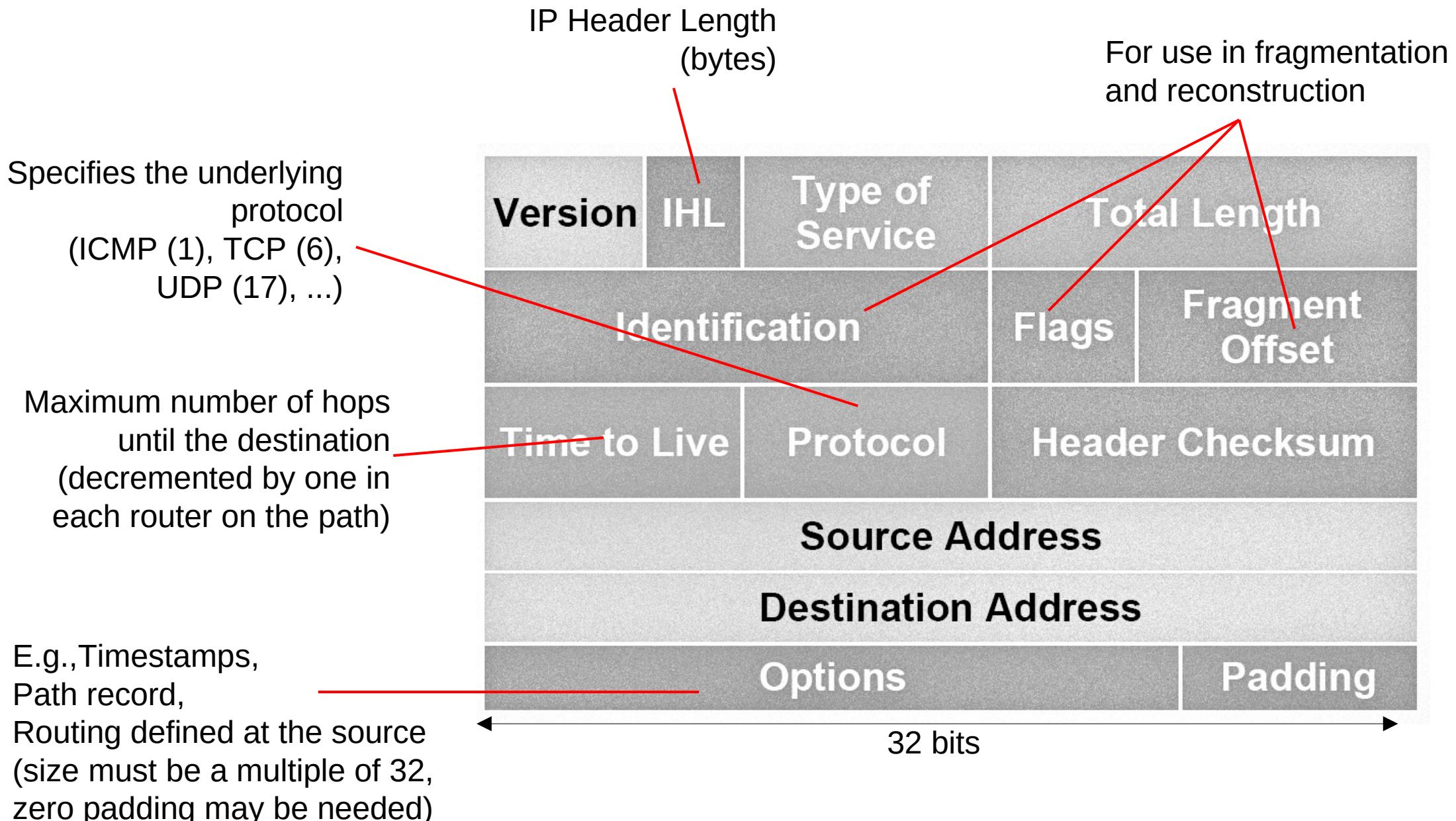
Mask Notations

- There are two notations for IPv4 masks:
 - ◆ Decimal: 4 bytes separated by dots.
 - ◆ CIDR: A slash (/) followed by a number with the number of bits of the network prefix.
- Both notations still exist today.
 - ◆ CIDR starts to become prevalent.
 - ◆ IPv6 only supports CIDR.

| CIDR | Decimal | CIDR | Decimal |
|-------------|----------------|-------------|-----------------|
| /21 | 255.255.248.0 | /30 | 255.255.255.252 |
| /20 | 255.255.240.0 | /29 | 255.255.255.248 |
| /19 | 255.255.224.0 | /28 | 255.255.255.240 |
| /18 | 255.255.192.0 | /27 | 255.255.255.224 |
| /17 | 255.255.128.0 | /26 | 255.255.255.192 |
| /16 | 255.255.0.0 | /25 | 255.255.255.128 |
| /15 | 255.248.0.0 | /24 | 255.255.255.0 |
| /14 | 255.240.0.0 | /23 | 255.255.254.0 |
| /13 | 255.224.0.0 | /22 | 255.255.252.0 |



IPv4 Packet Format (1)



IPv4 Packet Format (2)

- Version (4 bits) – Protocol version
- Header Length (4 bits) – Header size (number of blocks of 4 bytes)
 - ◆ Without options, the header uses 5 blocks of 4 bytes (20 bytes) and the first byte of the header is 0x45 (version 4, 5 blocks of 4 bytes).
- Type od Service (1 byte) – To implement QoS
 - ◆ By default is 0x00.
- Total Length (2 bytes) – packet size in bytes including the header.
 - ◆ Maximum IPv4 packet size is 65 535 bytes.
 - ◆ Usually this value is limited by the local network Maximum Transport Unit (MTU).



IPv4 Packet Format (3)

- Time to Live (1 byte) – maximum hops until destination
 - ◆ Each router on path reduces TTL by 1.
 - ◆ If TTL reaches 0 the packet is discarded and router may notify sender.
- Protocol (1 byte) – specifies the encapsulated protocol
- Header Checksum (2 bytes) – for header error detection
 - ◆ Each router on path must recalculate checksum.
 - ◆ Changes at least TTL.



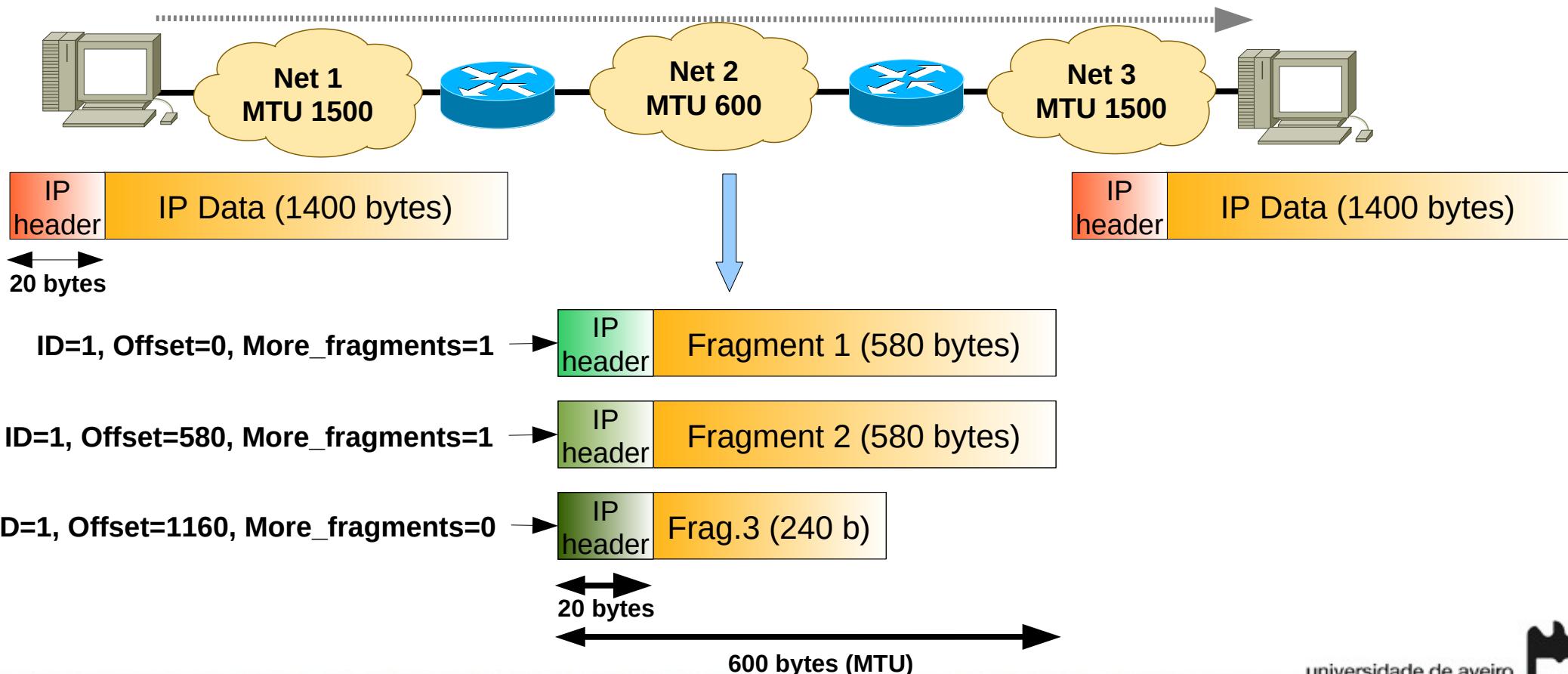
IPv4 Packet Format (4)

- Identification (2 bytes) – identifies fragments of the same original IPv4 packet.
- Flags (3 bits)
 - ◆ First bit for future use (always 0).
 - ◆ Second bit is 0 if packet can be fragment, and 1 otherwise (do not fragment).
 - ◆ The third bit is 0 for the last fragment, and 1 otherwise (more fragments flag).
- Fragment Offset (13 bits) – position (in multiples of 8 bytes) of a fragment in the original IPv4 packet (for first fragment is 0x00).



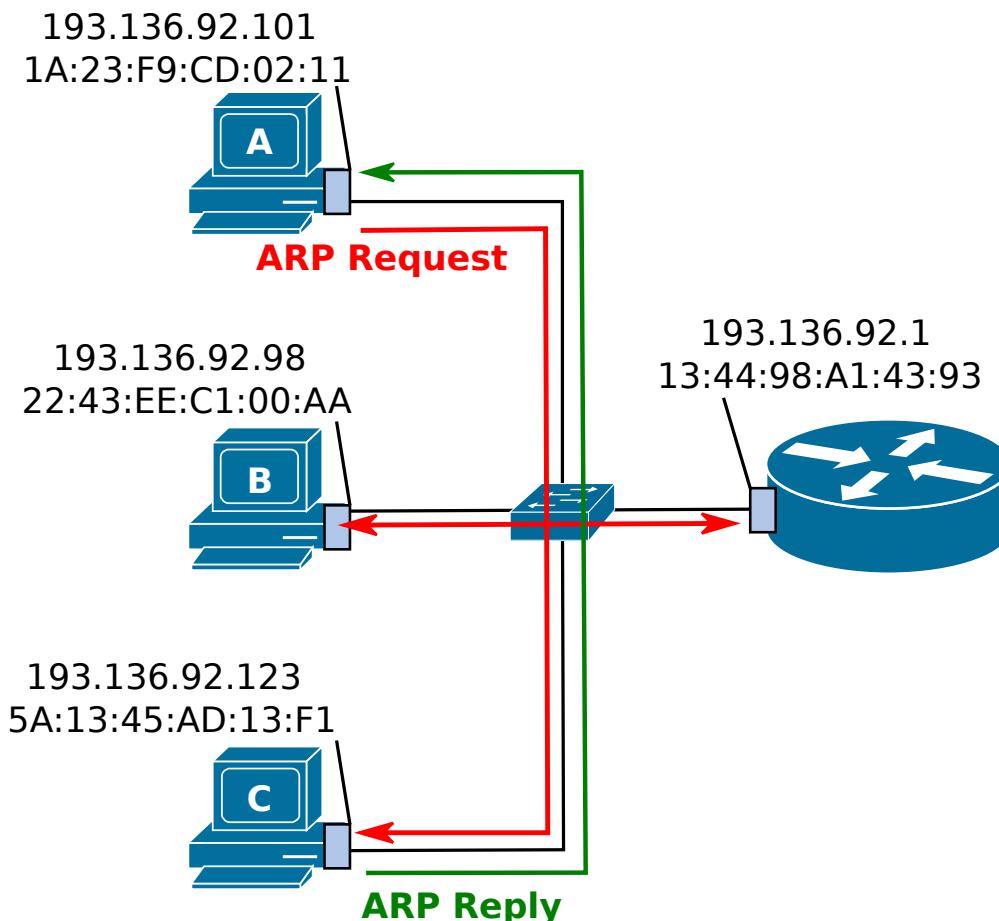
IPv4 Fragmentation and Reconstruction

- Each network defines the maximum packet that can be sent.
 - ◆ MTU - Maximum Transfer Unit
- For larger packets, the packet must be fragmented at entry and reconstructed after.
- Header fields used on the process:
 - ◆ Identification, fragment offset, flags: do not fragment e more fragments

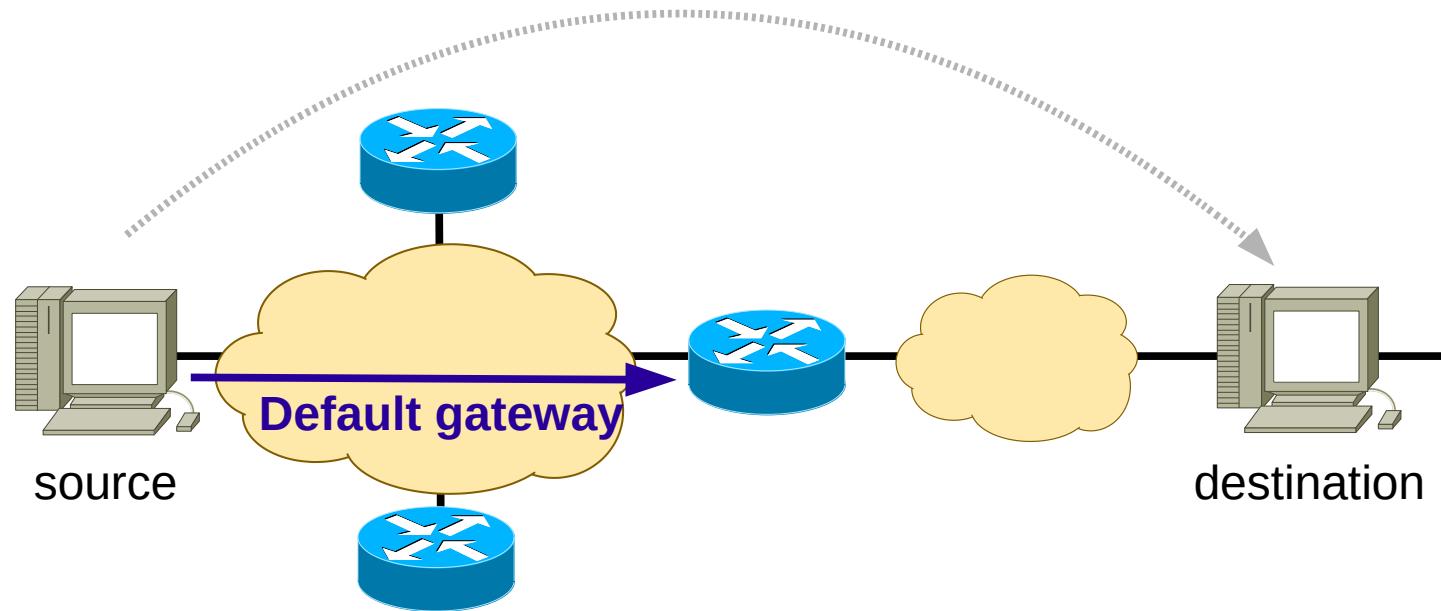


Address Resolution Protocol (ARP)

- IPv4: Address Resolution Protocol (ARP)
- Example:
 - ◆ When “A” wants to contact “C” by IPv4:
 - ◆ “A” requires “C” MAC address.
 - ◆ Only knows IPv4 address.
 - ◆ If “C” IPv4 address is not present in the ARP table, then:
 - “A” send an “ARP Request” in broadcast to the local network (destination MAC: FF:FF:FF:FF:FF:FF) with the IPv4 address of “C”,
 - All machines receive this packet,
 - “C” verifies that is IPv4 address is on the the “ARP request”, responds directly to “A” with a “ARP reply” (destination MAC==MAC of “A”) with its own MAC address.
 - ◆ MAC address resolution only happens in a the local network.
 - ◆ ARP packets do not pass through routers.



Routing to Another IP Network (1)



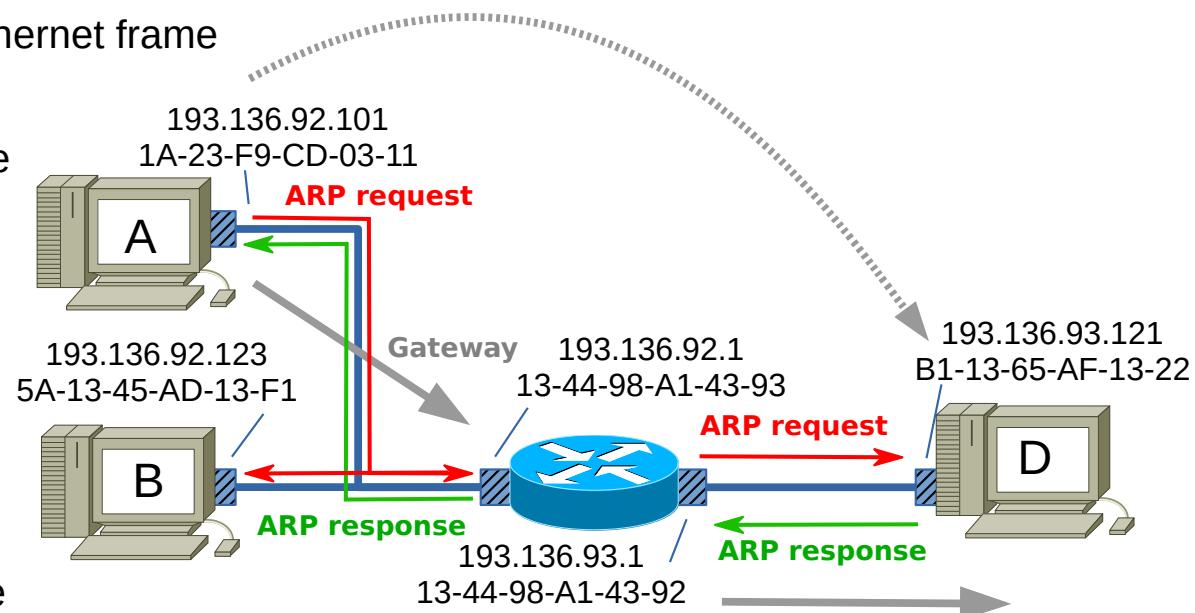
- When a host must send an IP packet to another IP, the packet must be sent to the **default gateway**.
- The **default gateway** must be provided at the same time than the IP address.
 - ◆ Manually or by self configuration.



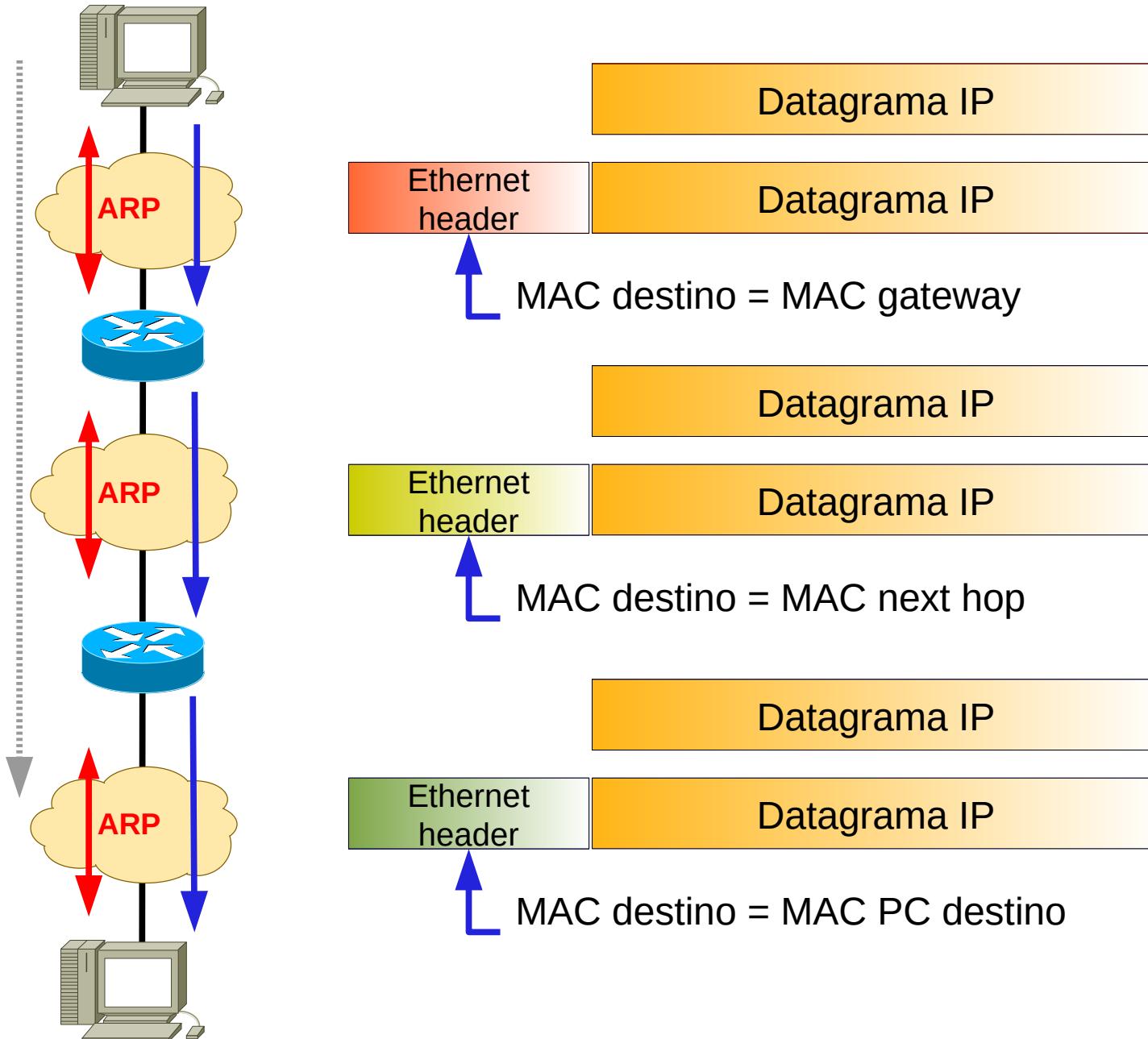
Routing to Another IP Network (2)

- Sending an IP packet from host “A” to host “D”

- “A” constructs the IP packet with the IPv4 address of “A” as source, and the IPv4 address of “D” as destination
- “A” verifies that the address of “D” belongs to a different IPv4 network, “A” will send the packet to the configured gateway (router)
- “A” determines the MAC address of the gateway (ARP)
- “A” constructs Ethernet frame with the MAC address of “A” as source and the MAC address of the gateway as destination
- “A” encapsulates the IP packet within the Ethernet frame
- “A” send the Ethernet Frame
- The router (GW) receives the Ethernet frame
- The router removes the IP packet from the Ethernet frame, and verifies that the destination is “D”
- The router determines the MAC address of “D” (ARP)
- The router constructs a new Ethernet frame with the MAC address of the output interface as source and the MAC address of “D” as destination
- The router encapsulates the received IP packet (changing just the TTL) within the Ethernet frame
- The router sends the Ethernet Frame



Routing over Multiple IP networks



IP Routing Overview (1)

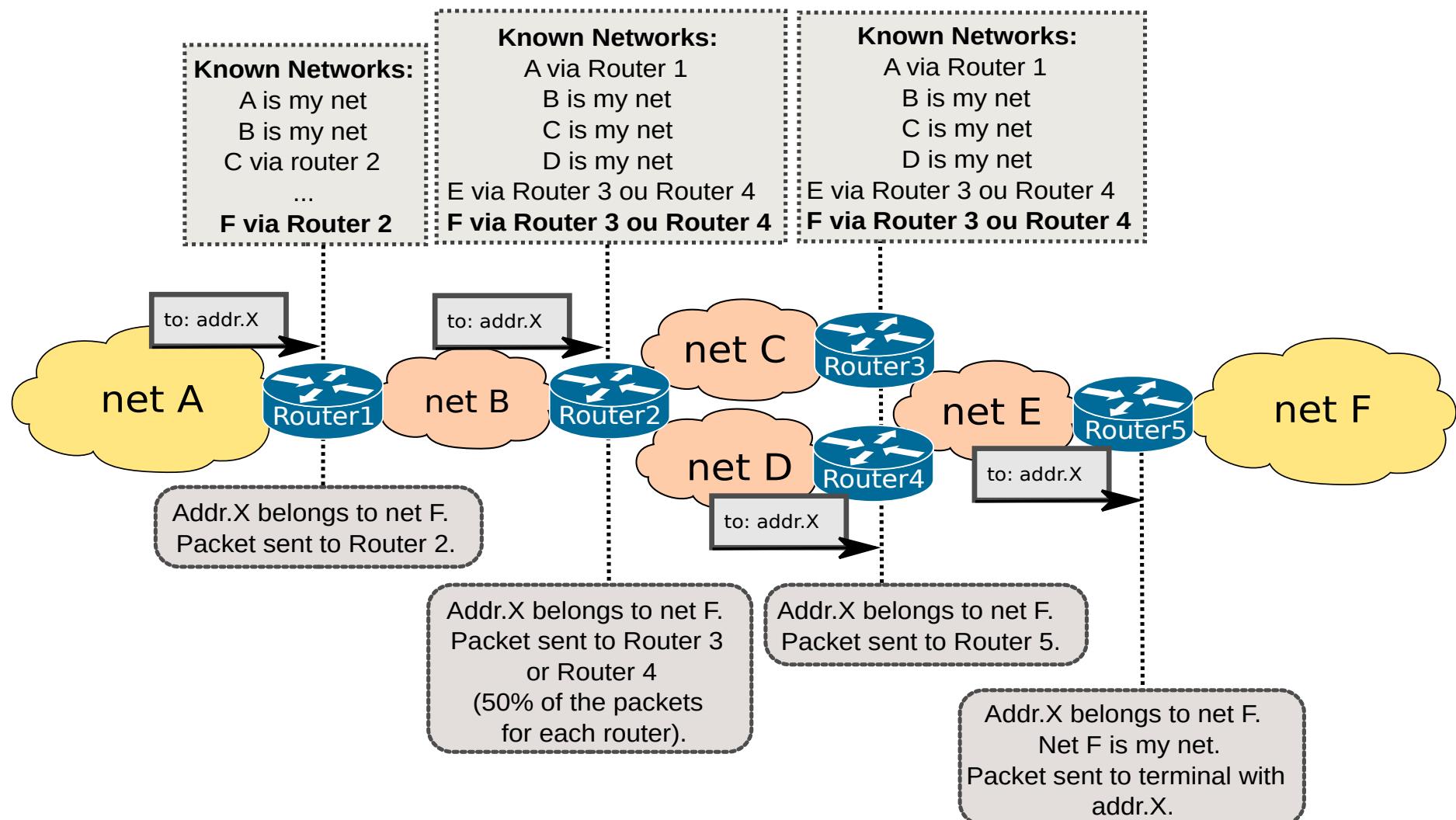
- Routers forward packets toward destination networks.
- Routers must be aware of destination networks to be able to forward packets to them.
- A router knows about the networks directly attached to its interfaces
- For networks not directly connected to one of its interfaces, however, the router must rely on outside information.
- A router can be made aware of remote networks by:
 - ◆ **Static routing:** An administrator manually configure the information.
 - ◆ **Dynamic routing:** Learns from other routers.
 - ◆ **Routing policies:** Manually routing rules that outweigh static/dynamic routing.



IP Routing Overview (2)

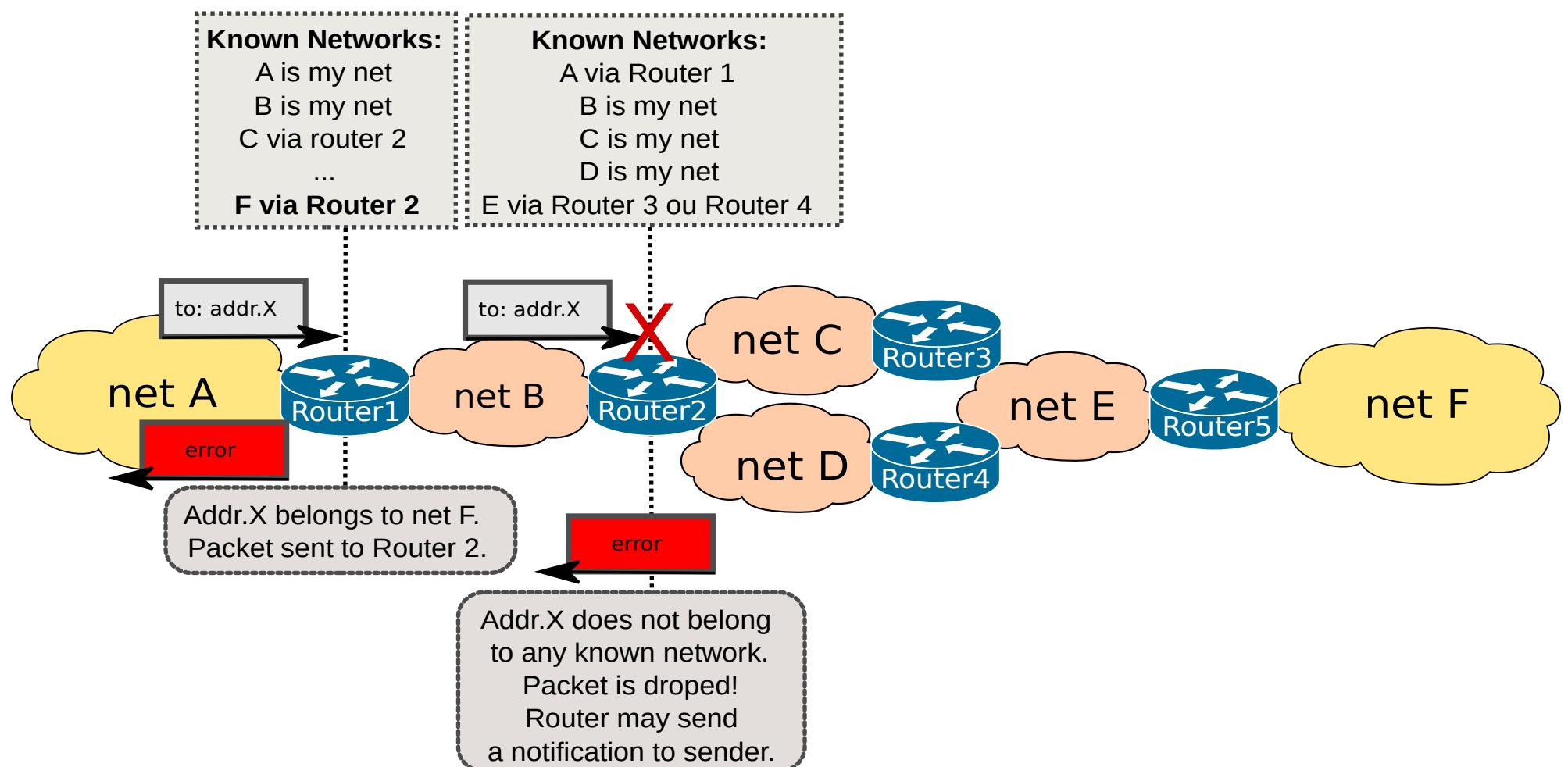
- Hop-by-hop decision:

- Based on the packets' **IP Destination Address**.
- Rules listed on the **IP Routing Table**.

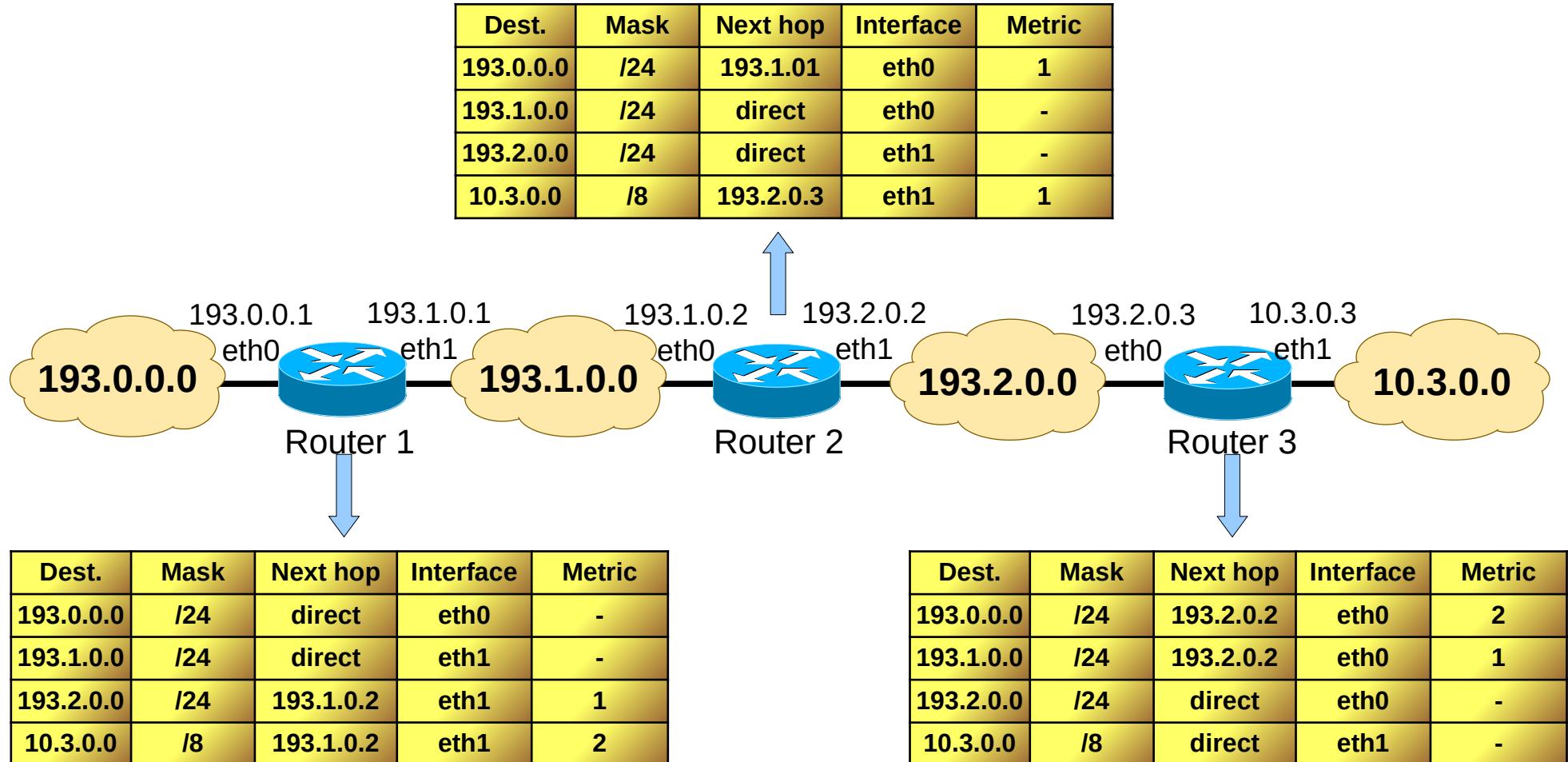


IP Routing Overview (3)

- Hop-by-hop decision:
 - ◆ If a packet for an unknown network reaches the router this will drop the packet, and MAY notify the sender about the routing error.



IP Routing Tables (1)



IP Routing Tables (2)

Cisco IOS

- Define how a remote network is reachable:

- Next-hop (identified by its address), and
- Local interface that provides connection.

- A network may be reachable using more than one path: (next-hop,local interface) pair.

- Mandatory elements

- Destination prefix
- Destination mask
- Metric
 - Could be defined by key tags.
 - e.g., Directly Connected

- One or both
 - Next-hop address
 - Output interface

- Optional elements

- Administrative distance
- Protocol
- Entry age (last time information received)
- Scope
- Flags
- Source-specific

- The next path hop (next hop address) may be found using more than one table entry (recursive resolution).

- e.g., Network A is reachable through address from network B, Network B is reachable through address from network C, ...

- The next-hop address may be obtained from external information (configurations or other mechanisms).

- e.g., Tunnels, Point-to-point connections, etc...

- When an entry uses a next-hop address from an unknown network, that entry is removed.

- All entries obtain by dynamic methods may have an entry age (time since last update/confirmation).

- After a timeout value without an update/confirmation the entry is removed.

```
R    200.1.1.0/24 [120/1] via 200.19.14.10, 00:00:16, FastEthernet0/1
    200.19.14.0/24 is variably subnetted, 2 subnets, 2 masks
C    200.19.14.0/24 is directly connected, FastEthernet0/1
L    200.19.14.4/32 is directly connected, FastEthernet0/1
R    200.38.0.0/24 [120/1] via 200.43.0.8, 00:00:03, FastEthernet1/1
    200.43.0.0/24 is variably subnetted, 2 subnets, 2 masks
C    200.43.0.0/24 is directly connected, FastEthernet1/1
L    200.43.0.1/32 is directly connected, FastEthernet1/1
```

Linux: route -n

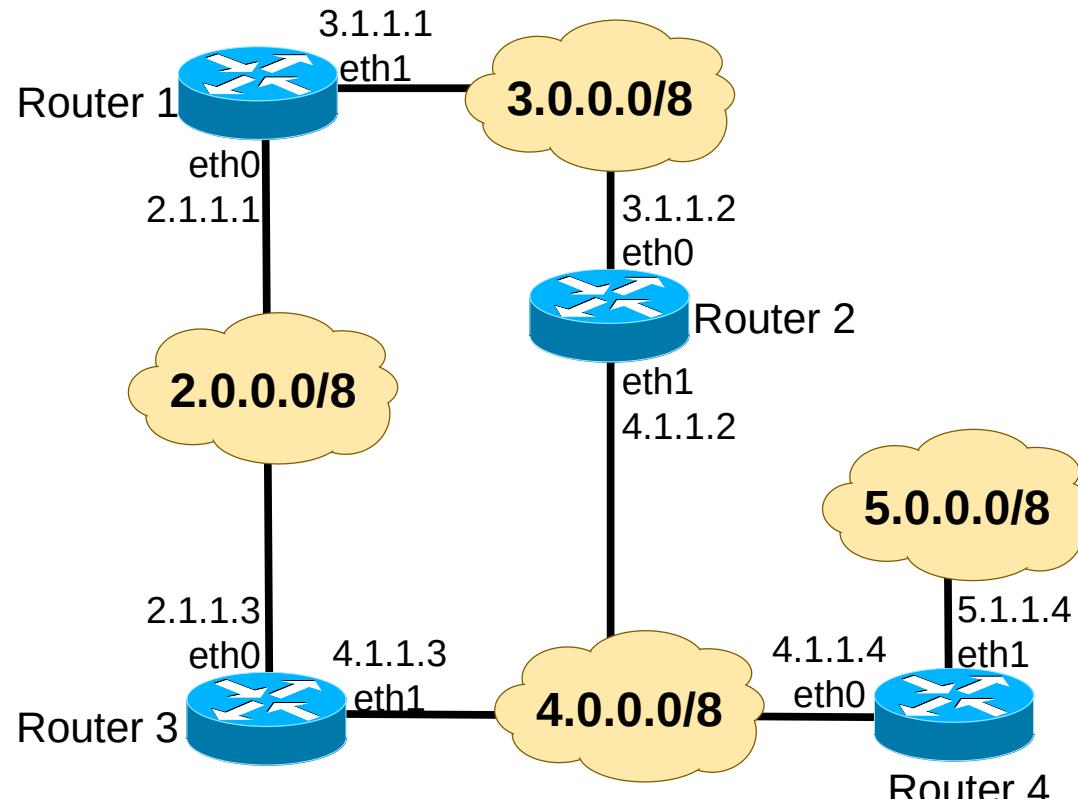
| Destination | Gateway | Genmask | Flags | Metric | Ref | Use | Iface |
|--------------|--------------|---------------|-------|--------|-----|-----|----------|
| 0.0.0.0 | 193.136.92.1 | 0.0.0.0 | UG | 100 | 0 | 0 | enp5s0f1 |
| 169.254.0.0 | 0.0.0.0 | 255.255.0.0 | U | 1000 | 0 | 0 | enp5s0f1 |
| 193.136.92.0 | 0.0.0.0 | 255.255.254.0 | U | 100 | 0 | 0 | enp5s0f1 |

Linux: ip route

```
default via 193.136.92.1 dev enp5s0f1 proto static metric 100
169.254.0.0/16 dev enp5s0f1 scope link metric 1000
193.136.92.0/23 dev enp5s0f1 proto kernel scope link src 193.136.93.104 metric 100
```



IP Routing Example



C 2.0.0.0/8 is directly connected, Ethernet0

R 3.0.0.0/8 [120/1] via 4.1.1.2, 00:00:06, Ethernet1

[120/1] via 2.1.1.1, 00:00:05, Ethernet0

C 4.0.0.0/8 is directly connected, Ethernet1

R 5.0.0.0/8 [120/1] via 4.1.1.4, 00:00:20, Ethernet1

Router 3

C 2.0.0.0/8 is directly connected, Ethernet0

C 3.0.0.0/8 is directly connected, Ethernet1

R 4.0.0.0/8 [120/1] via 3.1.1.2, 00:00:16, Ethernet1

[120/1] via 2.1.1.3, 00:00:12, Ethernet0

R 5.0.0.0/8 [120/2] via 3.1.1.2, 00:00:13, Ethernet1

[120/2] via 2.1.1.3, 00:00:02, Ethernet0

Router 1

R 2.0.0.0/8 [120/1] via 4.1.1.3, 00:00:26, Ethernet1

[120/1] via 3.1.1.1, 00:00:02, Ethernet0

C 3.0.0.0/8 is directly connected, Ethernet0

C 4.0.0.0/8 is directly connected, Ethernet1

R 5.0.0.0/8 [120/1] via 4.1.1.4, 00:00:23, Ethernet1

Router 2

R 2.0.0.0/8 [120/1] via 4.1.1.3, 00:00:13, Ethernet0

R 3.0.0.0/8 [120/1] via 4.1.1.2, 00:00:08, Ethernet0

C 4.0.0.0/8 is directly connected, Ethernet0

C 5.0.0.0/8 is directly connected, Ethernet1

Router 4



Layer 3 - Addressing

Fundamentos de Redes

**Mestrado Integrado em
Engenharia de Computadores e Telemática
DETI-UA**

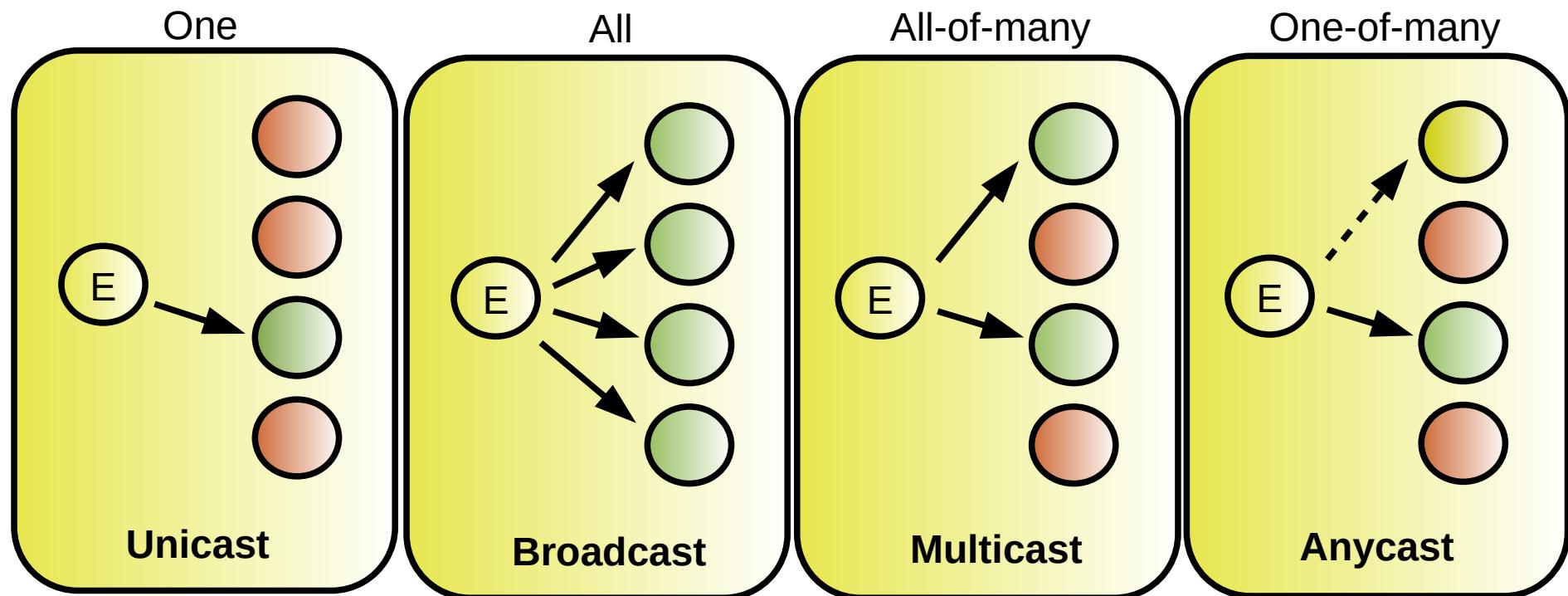


universidade de aveiro

deti.ua.pt

Types of Addresses

- Unicast – Identify a single sender/receiver.
- Broadcast – All are receivers.
- Multicast – Identify all elements of a group as receivers (all-of-many)
- Anycast – Identifies any element of group as receiver (one-of-many)



IPv4 Addressing

- An IPv4 address is a unique address for a network interface
- Exceptions:
 - ◆ Dynamically assigned IPv4 addresses (DHCP)
 - ◆ IP addresses in private networks (NAT)
- An IPv4 address:
 - ◆ is a **32 bit long** identifier
 - ◆ encodes a network number (**network prefix**)
and a **host identifier**



Network Prefix and Host Identifier

- The network prefix identifies a network and the host identifier identifies a specific host (actually, interface on the network).



- How do we know how long the network prefix is?
 - Before 1993:** The boundary between network prefix and host identifier is implicitly defined (**class-based/classful addressing**)
or
 - After 1993:** The boundary between network prefix and host identifier is indicated by a **netmask**.



Classless Inter-Domain Routing (CIDR)

- New interpretation of the IP addressing to increase efficiency and flexibility.
 - ◆ Network Masks were created to define the boundary between the IP network prefix and host identifier.
 - ◆ A bit of the mask equal to one indicate that that bit (in that position) of the address belongs to the network prefix.
 - ◆ A bit of the mask equal to zero indicate that that bit (in that position) of the address belongs to the host identifier.
 - ◆ Called VLSM (Variable Length Subnet Mask).
 - ◆ Must be provided with the IP address.
- Allowed the partition of a network in smaller networks or sub-networks (subnets).
- Allowed to merge several network under a single prefix (aggregation or summary process).

| | decimal | binary |
|--------------|---------------|-------------------------------------|
| IPv4 Address | 193.136.92.1 | 11000001.10001000.01011100.00000001 |
| Mask | 255.255.255.0 | 11111111.11111111.11111111.00000000 |

↔ → ← →

network prefix host identifier network prefix host identifier



Mask Notations

- There are two notations for IPv4 masks:
 - ◆ Decimal: 4 bytes separated by dots.
 - ◆ CIDR: A slash (/) a number with the number of bits of the network prefix.
- Both notations still exist today.
 - ◆ CIDR starts to become prevalent.
 - ◆ IPv6 only supports CIDR.

| CIDR | Decimal | CIDR | Decimal |
|-------------|----------------|-------------|-----------------|
| /21 | 255.255.248.0 | /30 | 255.255.255.252 |
| /20 | 255.255.240.0 | /29 | 255.255.255.248 |
| /19 | 255.255.224.0 | /28 | 255.255.255.240 |
| /18 | 255.255.192.0 | /27 | 255.255.255.224 |
| /17 | 255.255.128.0 | /26 | 255.255.255.192 |
| /16 | 255.255.0.0 | /25 | 255.255.255.128 |
| /15 | 255.248.0.0 | /24 | 255.255.255.0 |
| /14 | 255.240.0.0 | /23 | 255.255.254.0 |
| /13 | 255.224.0.0 | /22 | 255.255.252.0 |



CIDR Address Blocks

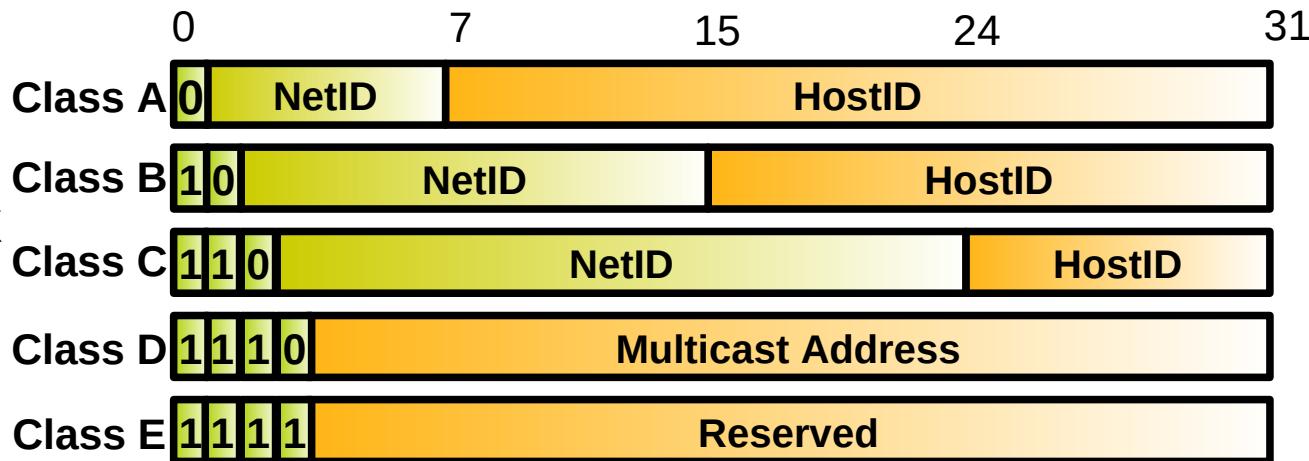
- CIDR defines a block of addresses.
- The addresses blocks are used to assign
- #Addresses= $2^{(32-\text{CIDR})}$
 - ◆ Example: $\backslash 24 \rightarrow 2^{(32-24)} = 2^8 = 256$, $\backslash 28 \rightarrow 2^{(32-28)} = 2^4 = 16$
- #Usable Addresses = #Addresses – 2 addresses
 - ◆ Network prefix and broadcast address

| CIDR | # of addresses | # usable addresses |
|------|----------------|--------------------|
| 21 | 2048 | 2046 |
| 20 | 4096 | 4094 |
| 19 | 8192 | 8190 |
| 18 | 16384 | 16382 |
| 17 | 32768 | 32766 |
| 16 | 65536 | 65534 |
| 15 | 131072 | 131070 |
| 14 | 262144 | 262142 |
| 13 | 524288 | 524286 |

| CIDR | # of addresses | # usable addresses |
|------|----------------|--------------------|
| 30 | 4 | 2 |
| 29 | 8 | 6 |
| 28 | 16 | 14 |
| 27 | 32 | 30 |
| 26 | 64 | 62 |
| 25 | 128 | 126 |
| 24 | 256 | 254 |
| 23 | 512 | 510 |
| 22 | 1024 | 1022 |

IPv4 Classful Addressing

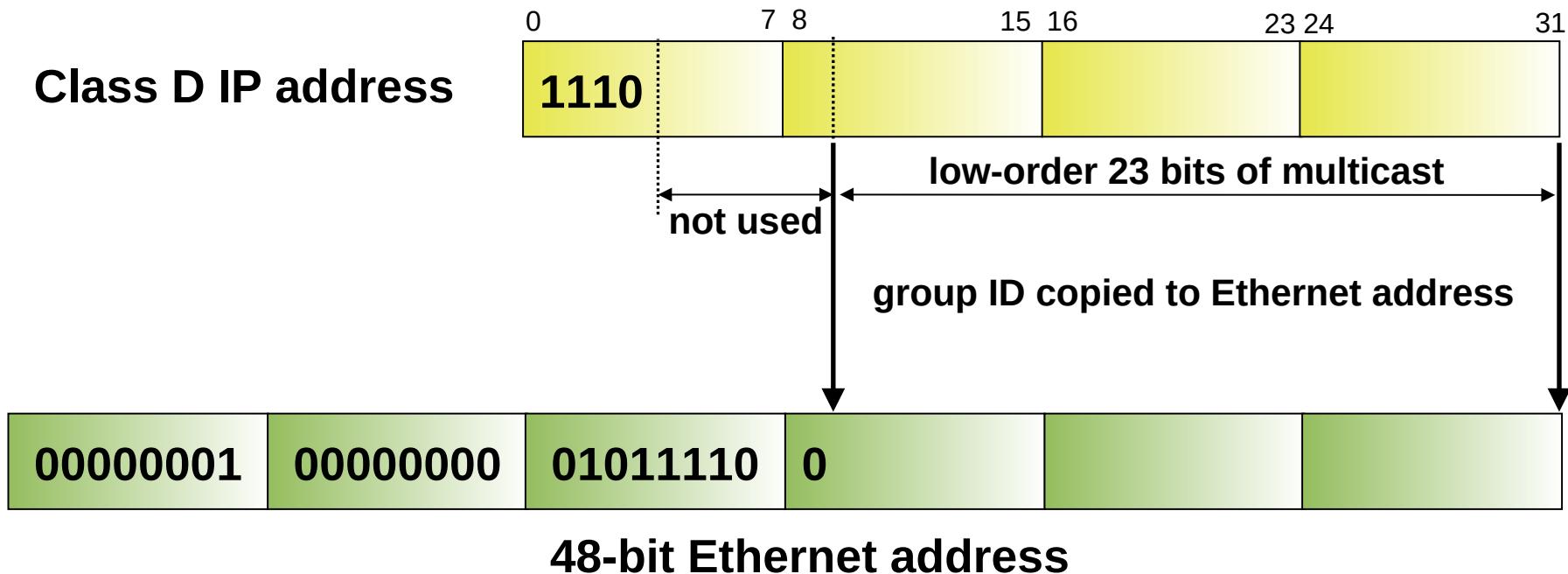
- Initially (until 1993) the boundary between the network prefix and host identifier was predefined by the value of the first byte (class).
- Resulted in a huge waste of addresses:
 - Classes A and B were too big,
 - Not enough class C networks.
- Routing Tables were becoming very long
 - It was not possible to merge (aggregate) networks to simplify routing tables.



| Class | First Address | Last Address |
|-------|---------------|-----------------|
| A | 1.0.0.0 | 126.0.0.0 |
| B | 128.0.0.0 | 191.255.0.0 |
| C | 192.0.0.0 | 223.255.255.0 |
| D | 224.0.0.0 | 239.255.255.255 |
| E | 240.0.0.0 | 255.255.255.254 |



Conversion of Multicast IPv4 Address to Ethernet Address



IPv4 Private Networks

| Prefix | First Address | Last Address |
|----------------|---------------|-----------------|
| 10.0.0.0/8 | 10.0.0.0 | 10.255.255.255 |
| 172.16.0.0/12 | 172.16.0.0 | 172.31.255.255 |
| 192.168.0.0/16 | 192.168.0.0 | 192.168.255.255 |
| 169.254.0.0/16 | 169.254.0.0 | 169.254.255.255 |

- To be used within a local network.
- Packets with these addresses as destination are not routed to the Internet.
- Packets with these addresses as source should not be routed to the Internet.
 - ◆ Not default behavior!



IPv4 Address Planning

IPv4 Network Sub-netting

- Made allowed by Variable Length Subnet Mask.
- Division of an IPv4 networks into smaller IPv4 networks.
- Allows to save IPv4 addresses.
 - ◆ Assign a large network to a small network will have many address not assigned.
 - ◆ A large network may divided into smaller networks and each one assign to different LAN.

$193.136.92.0/24 \rightarrow 193.136.92.0/25 + 193.136.92.128/25$

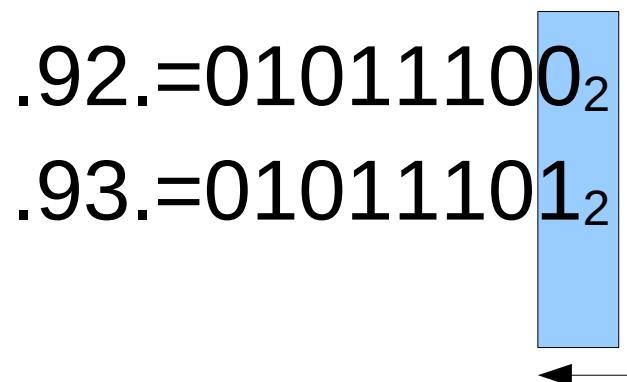
.92.000=01011100.00000000₂
.92.128=01011100.10000000₂



IPv4 Network Aggregation

- Inverse process to network sub-netting.
- Used to obtain a single network prefix to multiple networks.
 - Mainly used to simplify routing.
- Example:

$193.136.92.0/24 + 193.136.93.0/24 \rightarrow 193.136.92.0/23$



IPv4 Address Planning (1)

- Address planning is the assignment of an IP network to a (V)LAN.
 - ◆ To be assign address manually or dynamically (DHCP).
- Public addresses planning:
 - ◆ Limited number of available IPv4 addresses.
 - ◆ Planning ruled by the number of hosts in each LAN that require a public IPv4 address.
 - ◆ Not all LAN require IPv4 addresses.
 - ◆ Not all host in a LAN require IPv4 addresses.
 - ◆ Usually network managers receive /23, /24 or /25 networks.
- Private addresses planning:
 - ◆ Number of addresses is not an issue.
 - ◆ Number of hosts in a LAN is not so relevant.
 - ◆ Networks are usually divided in standard (/24), point-to-point (/30) and larger networks (may use /23, /22, /21, /20, etc...).



IPv4 Address Planning (2)

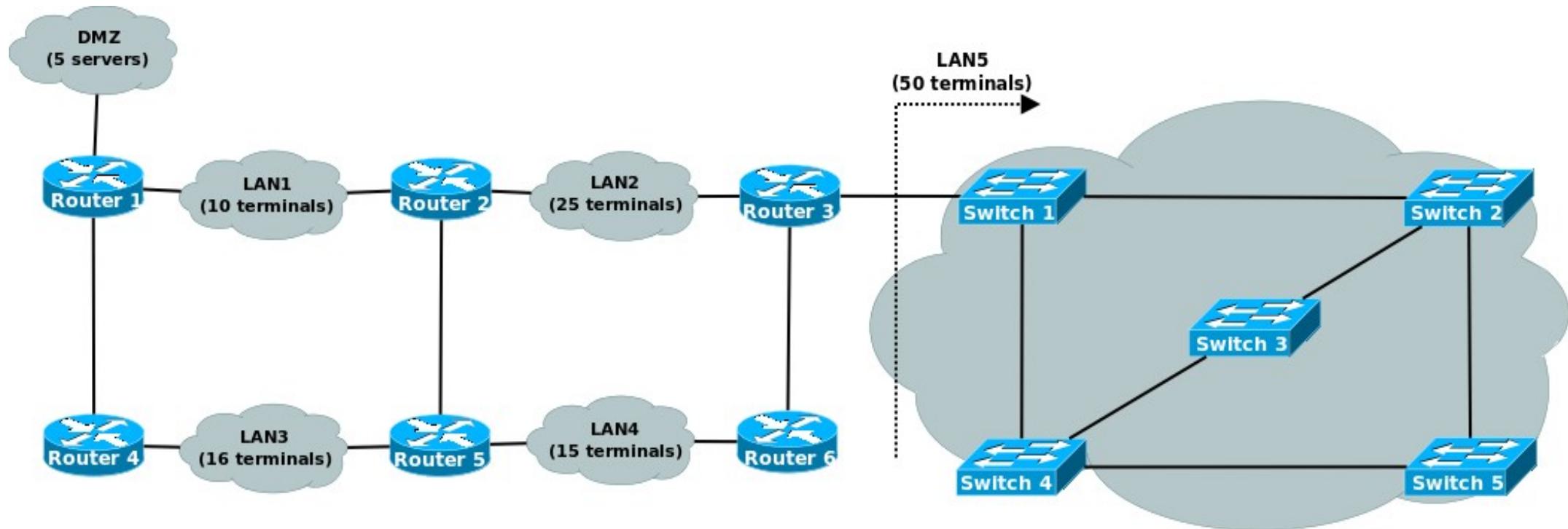
- Best practices:

- Best practices:
 - ◆ Identify the available IPv4 network(s).
 - ◆ Identify the number of host in each (V)LAN.
 - Including terminals and routers (gateways).
 - ◆ Define each sub-network size.
 - Network prefix and broadcast addresses are not usable.
 - Define network mask.
 - ◆ Sort sub-networks from larger to smaller.
 - Smaller CIDR to higher CIDR.
 - ◆ Start from the available network.
 - Sub-divide in half.
 - If sub-network size is required → **Assigned it** → ITS SUB-NETWORKS ARE NOT USABLE IN OTHER LAN.
 - If sub-network size is larger than required → **Sub-divide it in half.**
 - Repeat until all LAN have an assigned IPv4 network.
 - The overall available network may not be enough to assign sub-networks to all LAN. The solution is to reevaluate requirements and assign smaller sub-networks.



Example – IPv4 Public Planning (1)

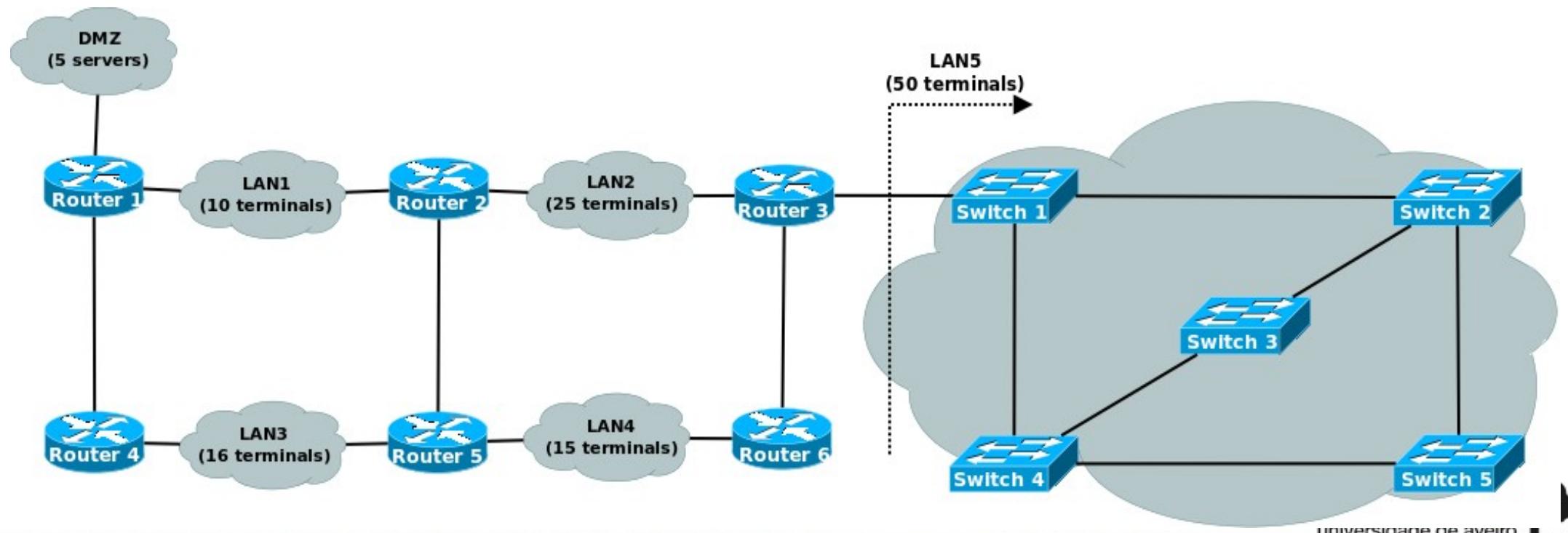
- Problem: Multiple (V)LAN require a small number of public IPv4 addresses. The public IPv4 network available is 193.1.1.0/24.
 - ◆ Note: All (V)LAN require IPv4 addresses, however may use private addresses (another IPv4 network).



192.1.1.0/24

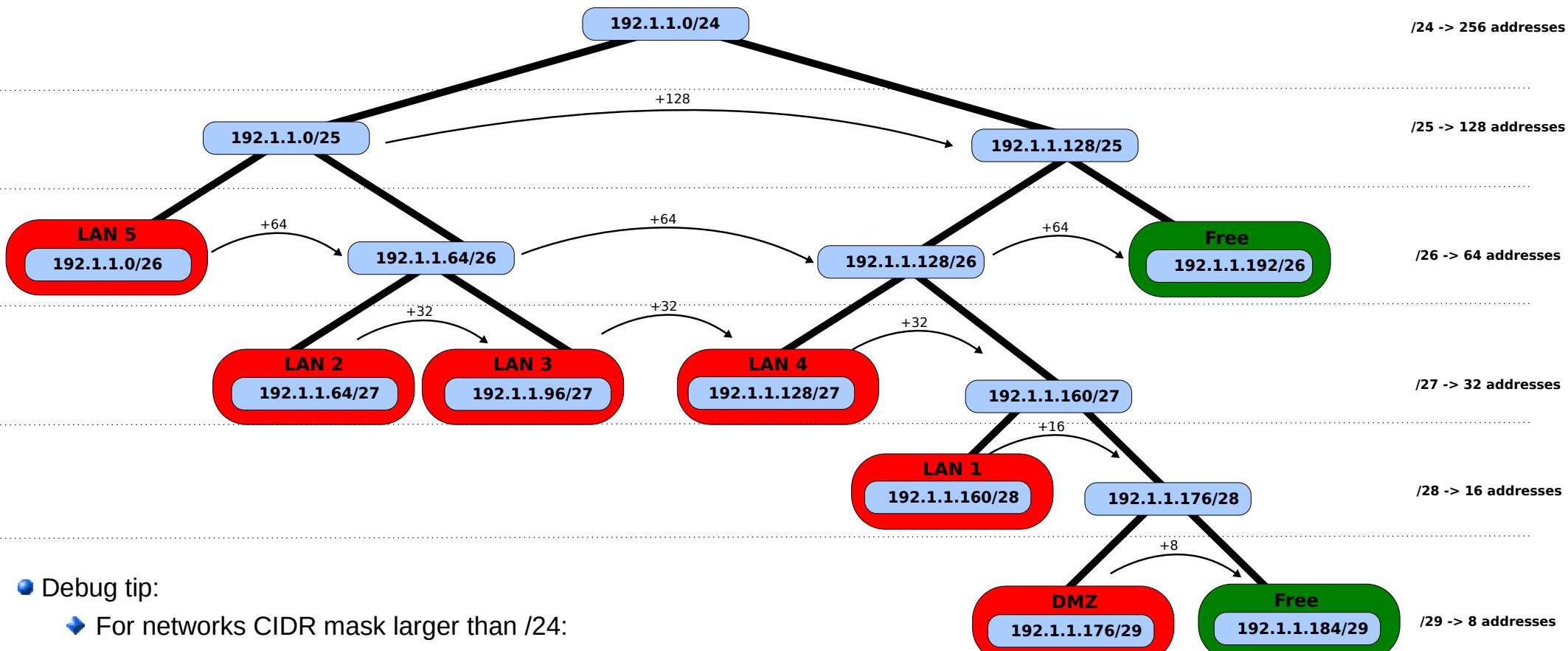
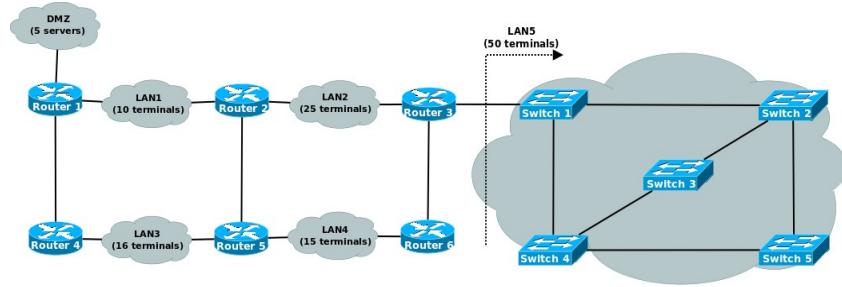
Example – IPv4 Public Planning (2)

- LAN 1 → 10+2 routers/gw+prefix+broadcast= 14 → 16 → /28 net
- LAN 2 → 25+2 routers/gw+prefix+broadcast= 29 → 32 → /27 net
- LAN 3 → 16+2 routers/gw+prefix+broadcast= 20 → 32 → /27 net
- LAN 4 → 15+2 routers/gw+prefix+broadcast= 19 → 32 → /27 net
- LAN 5 → 50+1 router/gw+prefix+broadcast= 53 → 64 → /26 net
- DMZ → 5+1 router/gw+prefix+broadcast = 8 → 8 → /29 net



- LAN 1 → $10+2+2=14$ → 16 → /28 net
- LAN 2 → $25+2+2=29$ → 32 → /27 net
- LAN 3 → $16+2+2=20$ → 32 → /27 net
- LAN 4 → $15+2+2=19$ → 32 → /27 net
- LAN 5 → $50+1+2=53$ → 64 → /26 net
- DMZ → $5+1+2=8$ → 8 → /29 net

Example (3)

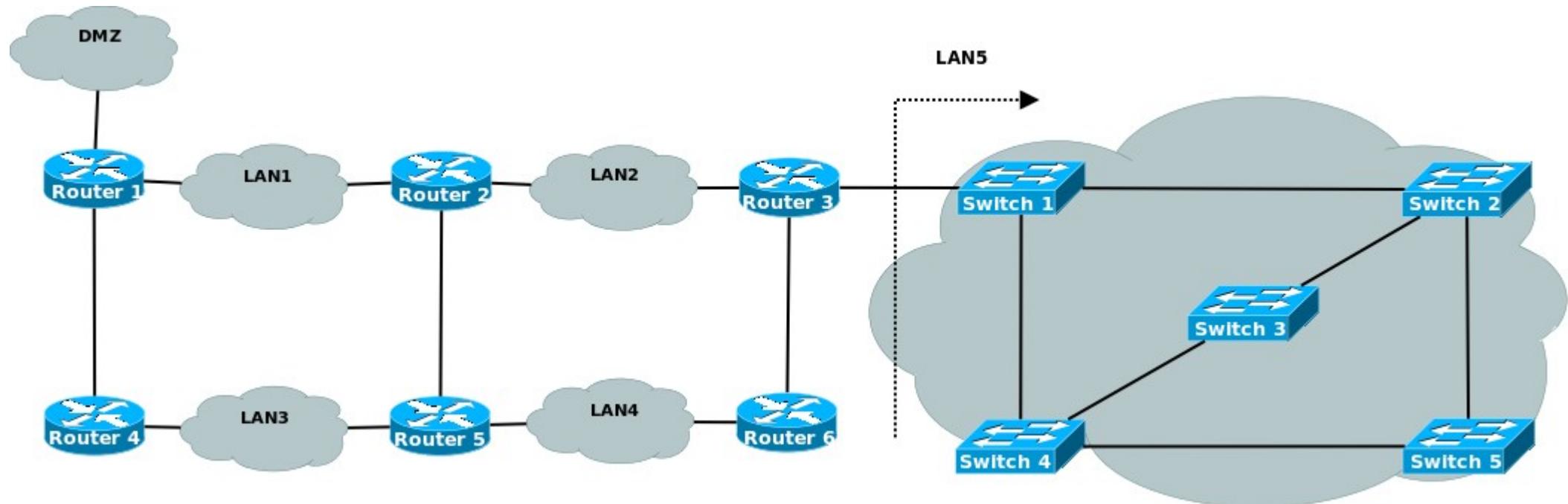


- Debug tip:
 - ◆ For networks CIDR mask larger than /24:
 - ◆ The last byte is always a multiple of the number of addresses for that network size.
 - ◆ Example: 192 is multiple of 64, 176 is multiple of 16, and 184 is multiple of 8.



Example – IPv4 Private Planning (1)

- Problem: All (V)LAN have a standard size, except LAN 5 that may have 1000 hosts.

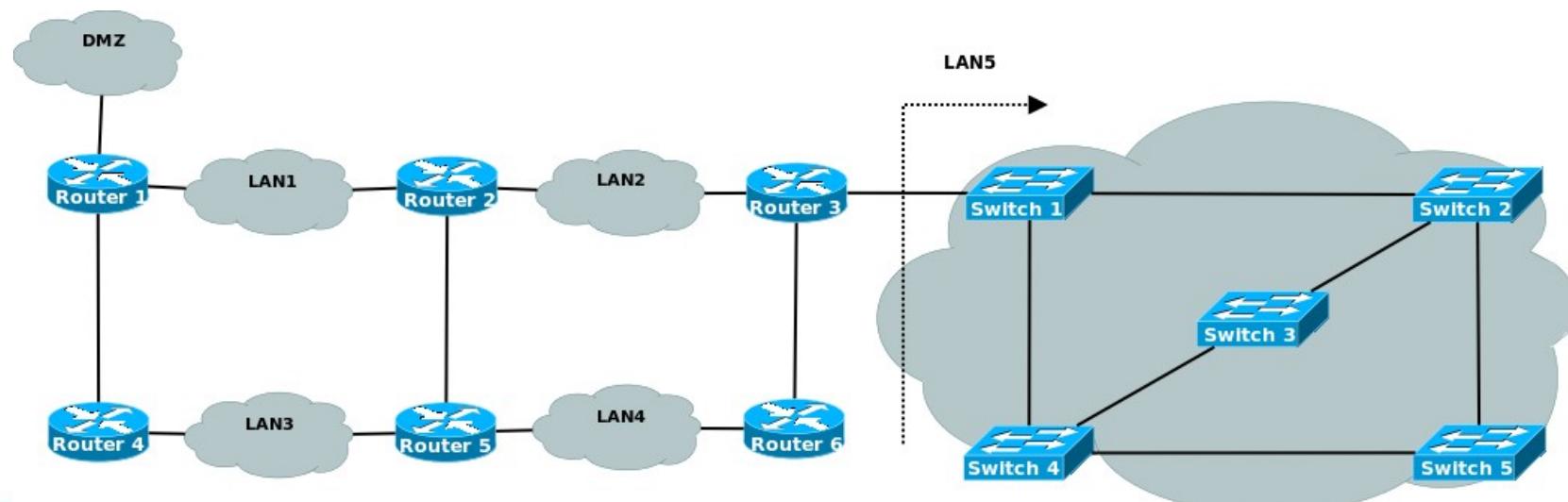


10.0.0.0/8



Example – IPv4 Private Planning (2)

- Easier approach is to start from /24 networks and perform sub-netting/aggregation as required.
- Start with larger networks.
- LAN5 with 1000 users (plus one router) will be a /22 network ($2^{(32-22)-2}=1022$ usable addresses).
 - ◆ Aggregation of networks 10.0.0.0/24, 10.0.1.0/24, 10.0.2.0/24 and 10.0.3.0/24.
 - ◆ Assigned: 10.0.0.0/22
- LAN1 to LAN4 and DMZ have a standard size and will be a /24 network.
 - ◆ Assigned: 10.0.4.0/24, 10.0.5.0/24, 10.0.6.0/24, 10.0.7.0/24, 10.0.8.0/24
- Point-to-point networks R1-R4, R2-R5 and R3-R6 will be /30 networks.
 - ◆ Network 10.0.9.0/24 will be used to perform the sub-netting.
 - ◆ Assigned: 10.0.9.0/30, 10.0.9.4/30, 10.0.9.8/30
 - ◆ Free: 10.0.9.12/30+10.0.9.16/28+10.0.9.32/27+10.0.9.64/26+10.0.9.128/25



DHCP

Dynamic Host Configuration Protocol (DHCP)

- Service for dynamic assignment of IP addresses.
 - ◆ Client-Server architecture.
- Extension of the Bootstrap Protocol, BOOTP, (RFC 1542)
 - ◆ Runs over UDP.
 - Server port 67 and client port 68.
- Address assignment follow a leasing paradigm.
- The assignment of address has four phases:
 - ◆ Discover
 - ◆ Offer
 - ◆ Request
 - ◆ Acknowledge
- DHCP servers provide:
 - ◆ Address, network mask and gateway.
 - ◆ May include additional information DNS server, Windows Domain Servers, etc...



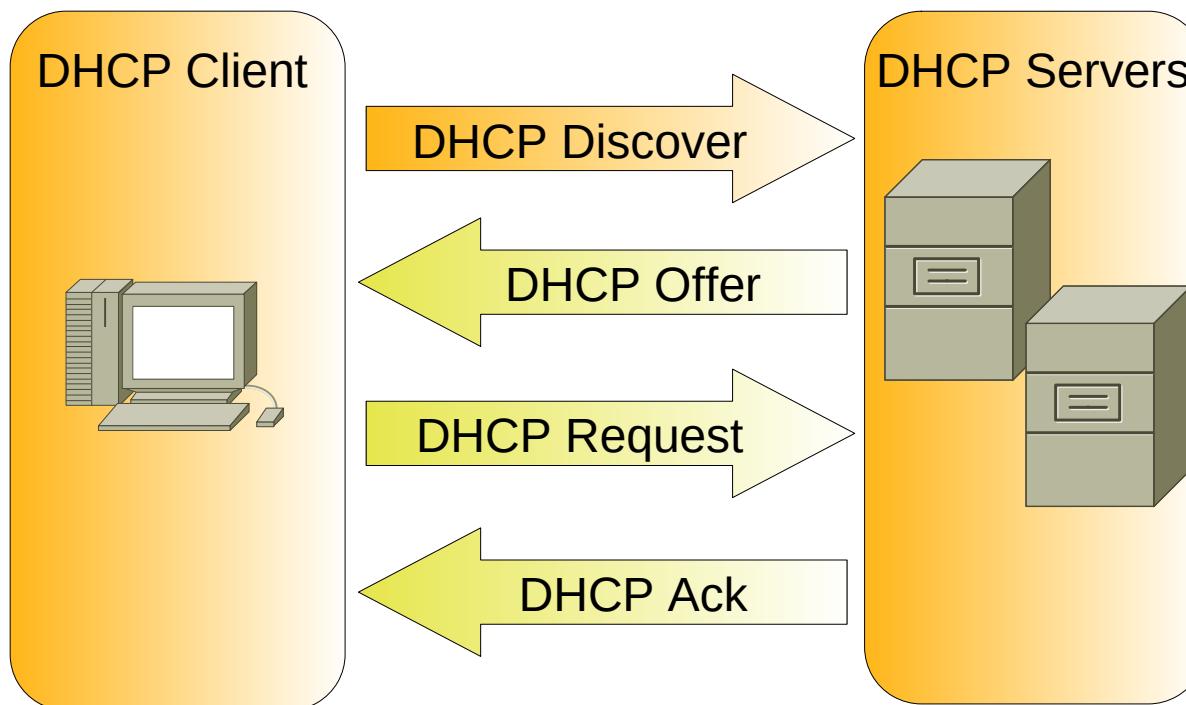
DHCP Server

- Pool of (public or private) addresses
 - ◆ List of IPv4 public or private addresses to be assigned, usually defined as network or range of IPv4 addresses.
- Exclusion ranges
 - ◆ Set of IPv4 addresses that belong to a pool but must be assigned.
 - ◆ Usually manually assigned address to routers (gateways) and servers.
- Reserved addresses and static assignment
 - ◆ Based on the MAC address is possible to define a permanently assigned IPv4 address.
 - ◆ Usually used on servers, printers and other network devices.
 - ◆ Should not be used by routers.
- Lease time
 - ◆ Define for how long can a host use an assigned IPv4 address without a new interaction.
- To serve multiple IPv4 networks (LAN):
 - ◆ The server must have multiple pools of addresses,
 - ◆ The routers must have the BootP/DHCP Relay feature configured.
- A LAN may have multiple DHCP servers
 - ◆ For redundancy. Pools must be disjoint.



Phase One: Discover

- The *DHCP Discover* message is encapsulated into a *BootP Request* packet.
 - ◆ Source address is 0.0.0.0.
- It is used to discover the available DHCP server(s).
- The client may include the desired address.
 - ◆ Server is not obliged to obey.



DHCP Discover

| No. | Time | Source | Destination | Protocol | Info |
|------|-----------|---------------|-----------------|----------|---------------|
| 1326 | 20.269579 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Discover |
| 1337 | 20.561380 | 193.136.92.65 | 193.136.93.228 | DHCP | DHCP Offer |
| 1338 | 20.561592 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Request |
| 1340 | 20.569560 | 193.136.92.65 | 193.136.93.228 | DHCP | DHCP ACK |

► Frame 1326 (342 bytes on wire, 342 bytes captured)

► Ethernet II, Src: 00:1d:ba:c0:a2:8e (00:1d:ba:c0:a2:8e), Dst: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)

► Internet Protocol, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)

► User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)

▼ Bootstrap Protocol

- Message type: Boot Request (1)
- Hardware type: Ethernet
- Hardware address length: 6
- Hops: 0
- Transaction ID: 0x42f5a54a
- Seconds elapsed: 0

► Bootp flags: 0x0000 (Unicast)

- Client IP address: 0.0.0.0 (0.0.0.0)
- Your (client) IP address: 0.0.0.0 (0.0.0.0)
- Next server IP address: 0.0.0.0 (0.0.0.0)
- Relay agent IP address: 0.0.0.0 (0.0.0.0)
- Client MAC address: 00:1d:ba:c0:a2:8e (00:1d:ba:c0:a2:8e)
- Client hardware address padding: 000000000000000000000000
- Server host name not given
- Boot file name not given
- Magic cookie: (OK)

► Option: (t=53,l=1) DHCP Message Type = DHCP Discover

► Option: (t=50,l=4) Requested IP Address = 192.168.1.71

► Option: (t=12,l=15) Host Name = "salvador-laptop"

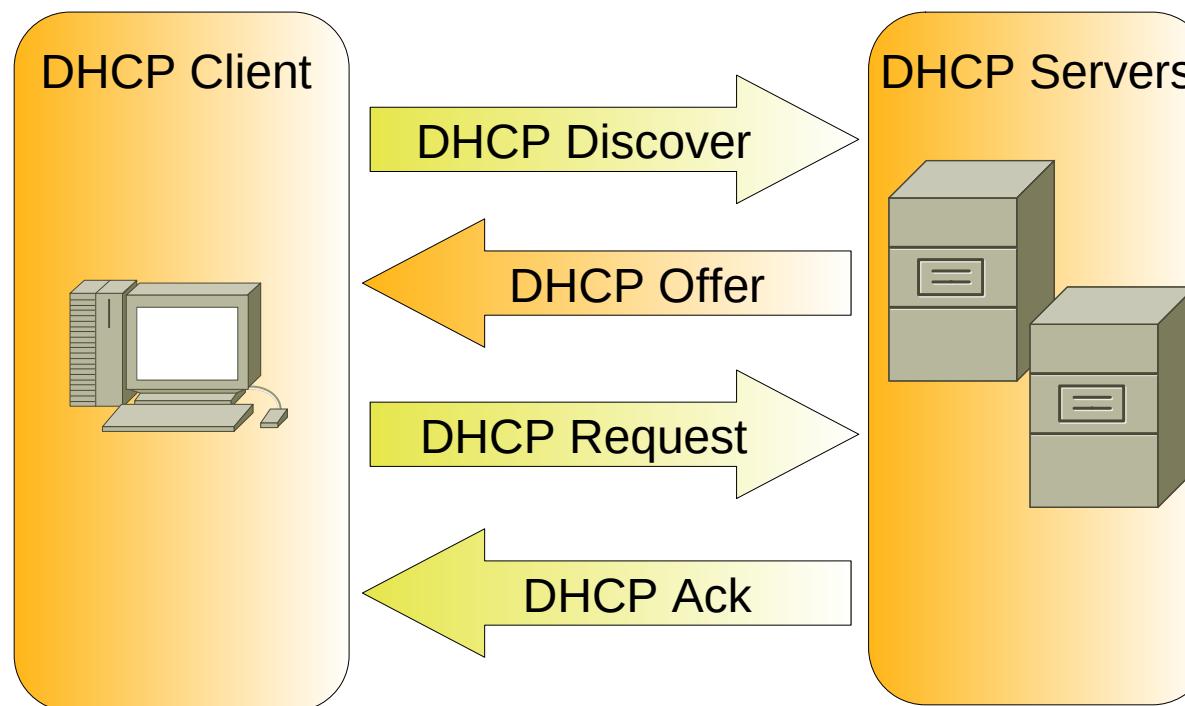
► Option: (t=55,l=13) Parameter Request List

- End Option
- Padding



Phase Two: Offer

- The *DHCP Offer* message is encapsulated into a *BootP Reply* packet.
- Each server proposes the lease of an IPv4 address to client.
 - ◆ If possible respect the client request (*Discovery*)



DHCP Offer

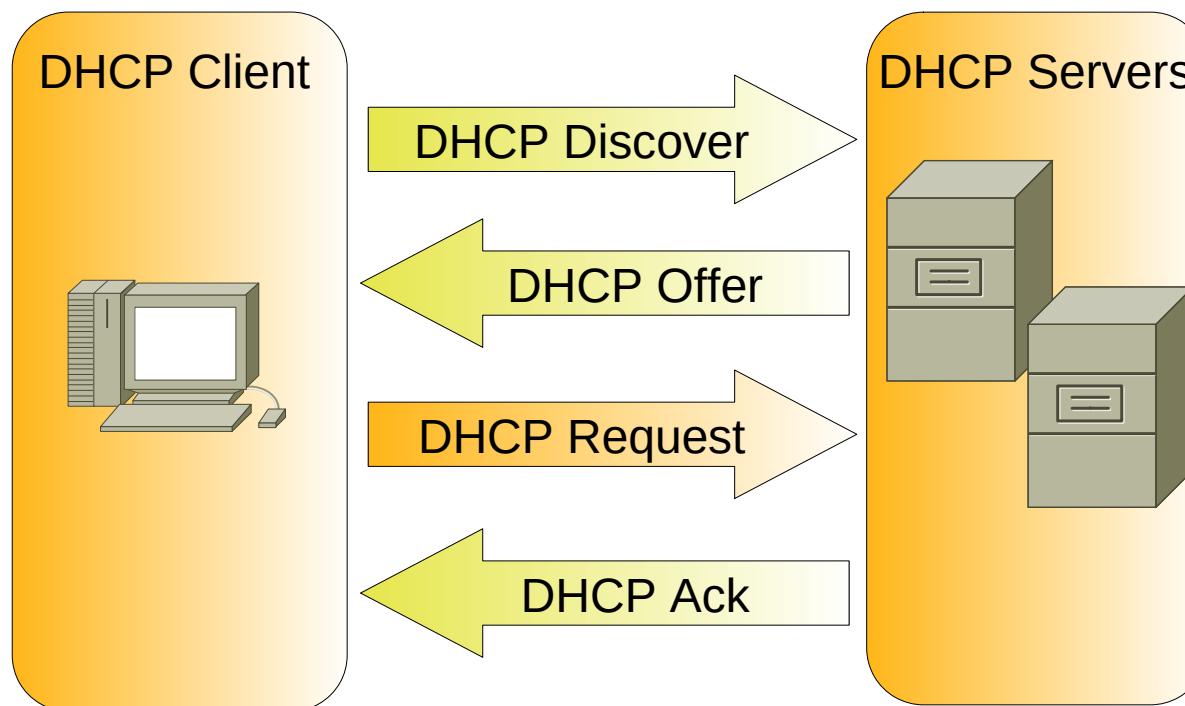
| No. | Time | Source | Destination | Protocol | Info |
|------|-----------|---------------|-----------------|----------|---------------|
| 1326 | 20.269579 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Discover |
| 1337 | 20.561380 | 193.136.92.65 | 193.136.93.228 | DHCP | DHCP Offer |
| 1338 | 20.561592 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Request |
| 1340 | 20.569560 | 193.136.92.65 | 193.136.93.228 | DHCP | DHCP ACK |

- ▷ Frame 1337 (342 bytes on wire, 342 bytes captured)
- ▷ Ethernet II, Src: 00:d0:b7:17:5b:6d (00:d0:b7:17:5b:6d), Dst: 00:1d:ba:c0:a2:8e (00:1d:ba:c0:a2:8e)
- ▷ Internet Protocol, Src: 193.136.92.65 (193.136.92.65), Dst: 193.136.93.228 (193.136.93.228)
- ▷ User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
- ▽ Bootstrap Protocol
 - Message type: Boot Reply (2)
 - Hardware type: Ethernet
 - Hardware address length: 6
 - Hops: 0
 - Transaction ID: 0x42f5a54a
 - Seconds elapsed: 0
 - ▷ Bootp flags: 0x0000 (Unicast)
 - Client IP address: 0.0.0.0 (0.0.0.0)
 - Your (client) IP address: 193.136.93.228 (193.136.93.228)
 - Next server IP address: 193.136.92.65 (193.136.92.65)
 - Relay agent IP address: 0.0.0.0 (0.0.0.0)
 - Client MAC address: 00:1d:ba:c0:a2:8e (00:1d:ba:c0:a2:8e)
 - Client hardware address padding: 00000000000000000000
 - Server host name not given
 - Boot file name not given
 - Magic cookie: (OK)
 - ▷ Option: (t=53,l=1) DHCP Message Type = DHCP Offer
 - ▷ Option: (t=54,l=4) DHCP Server Identifier = 193.136.92.65
 - ▷ Option: (t=51,l=4) IP Address Lease Time = 10 minutes
 - ▷ Option: (t=1,l=4) Subnet Mask = 255.255.254.0
 - ▷ Option: (t=3,l=4) Router = 193.136.92.1
 - ▷ Option: (t=15,l=8) Domain Name = "av.it.pt"
 - ▷ Option: (t=6,l=4) Domain Name Server = 193.136.92.65
 - End Option
 - Padding



Phase 3: Request

- The *DHCP Request* message is encapsulated into a *BootP Request* packet.
- The client may choose the offered IPv4 address (and DHCP server if more than one offer is received).



DHCP Request

| No. | Time | Source | Destination | Protocol | Info |
|------|-----------|---------------|-----------------|----------|------|
| 1326 | 20.269579 | 0.0.0.0 | 255.255.255.255 | DHCP | |
| 1337 | 20.561380 | 193.136.92.65 | 193.136.93.228 | DHCP | |
| 1338 | 20.561592 | 0.0.0.0 | 255.255.255.255 | DHCP | |
| 1340 | 20.569560 | 193.136.92.65 | 193.136.93.228 | DHCP | |

▶ Frame 1338 (342 bytes on wire, 342 bytes captured)

▶ Ethernet II, Src: 00:1d:ba:c0:a2:8e (00:1d:ba:c0:a2:8e), Dst: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)

▶ Internet Protocol, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)

▶ User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)

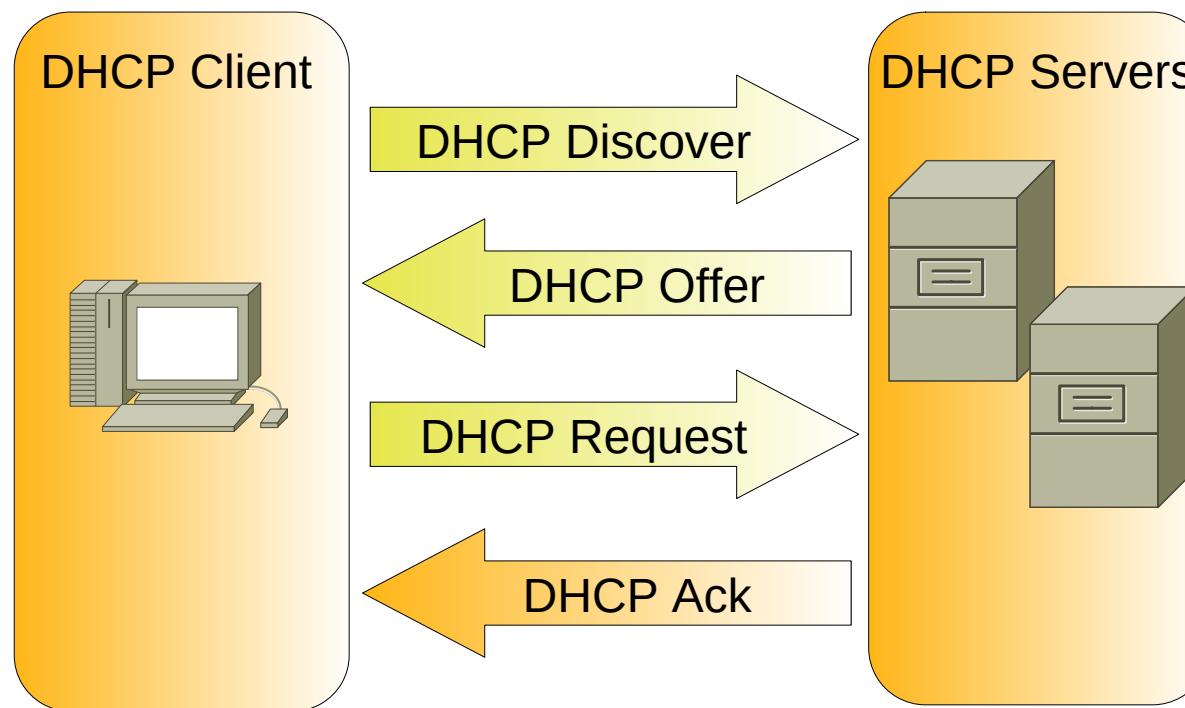
▼ Bootstrap Protocol

- Message type: Boot Request (1)
- Hardware type: Ethernet
- Hardware address length: 6
- Hops: 0
- Transaction ID: 0x42f5a54a
- Seconds elapsed: 0
- ▶ Bootp flags: 0x0000 (Unicast)
- Client IP address: 0.0.0.0 (0.0.0.0)
- Your (client) IP address: 0.0.0.0 (0.0.0.0)
- Next server IP address: 0.0.0.0 (0.0.0.0)
- Relay agent IP address: 0.0.0.0 (0.0.0.0)
- Client MAC address: 00:1d:ba:c0:a2:8e (00:1d:ba:c0:a2:8e)
- Client hardware address padding: 000000000000000000000000
- Server host name not given
- Boot file name not given
- Magic cookie: (OK)
- ▶ Option: (t=53,l=1) DHCP Message Type = DHCP Request
- ▶ Option: (t=54,l=4) DHCP Server Identifier = 193.136.92.65
- ▶ Option: (t=50,l=4) Requested IP Address = 193.136.93.228
- ▶ Option: (t=12,l=15) Host Name = "salvador-laptop"
- ▶ Option: (t=55,l=13) Parameter Request List
- End Option
- Padding



Phase 4: Acknowledge

- The *DHCP Ack* message is encapsulated into a *BootP Reply* packet.
- The server confirms the IPv4 address lease and provides additional information:
 - ◆ Lease time, Gateway(s), DNS server, etc...



DHCP Ack

| No. | Time | Source | Destination | Protocol | Info |
|------|-----------|---------------|-----------------|----------|---------------|
| 1326 | 20.269579 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Discover |
| 1337 | 20.561380 | 193.136.92.65 | 193.136.93.228 | DHCP | DHCP Offer |
| 1338 | 20.561592 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Request |
| 1340 | 20.569560 | 193.136.92.65 | 193.136.93.228 | DHCP | DHCP ACK |

```
▷ Frame 1340 (342 bytes on wire, 342 bytes captured)
▷ Ethernet II, Src: 00:d0:b7:17:5b:6d (00:d0:b7:17:5b:6d), Dst: 00:1d:ba:c0:a2:8e (00:1d:ba:c0:a2:8e)
▷ Internet Protocol, Src: 193.136.92.65 (193.136.92.65), Dst: 193.136.93.228 (193.136.93.228)
▷ User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
▽ Bootstrap Protocol
    Message type: Boot Reply (2)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x42f5a54a
    Seconds elapsed: 0
    ▷ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 193.136.93.228 (193.136.93.228)
    Next server IP address: 193.136.92.65 (193.136.92.65)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: 00:1d:ba:c0:a2:8e (00:1d:ba:c0:a2:8e)
    Client hardware address padding: 000000000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: (OK)
    ▷ Option: (t=53,l=1) DHCP Message Type = DHCP ACK
    ▷ Option: (t=54,l=4) DHCP Server Identifier = 193.136.92.65
    ▷ Option: (t=51,l=4) IP Address Lease Time = 10 minutes
    ▷ Option: (t=1,l=4) Subnet Mask = 255.255.254.0
    ▷ Option: (t=3,l=4) Router = 193.136.92.1
    ▷ Option: (t=15,l=8) Domain Name = "av.it.pt"
    ▷ Option: (t=6,l=4) Domain Name Server = 193.136.92.65
    End Option
    Padding
```



DHCP Operational Details

- Address Leasing Times
 - ◆ T1 Time (50% of Lease Time) – time after which the client must renew the address lease.
 - ◆ T2 Time (85% of Lease Time) – time after which the client must renew the address lease if the first attempt failed.
 - Lease Time – time after which the client can not use the leased address.
- DHCP allows multiple servers
 - ◆ Recommended for redundancy.
 - ◆ Requires
 - ◆ Advantage: resilience to operational failures.
 - ◆ Requirement: Disjointed pool of addresses in different servers.



DHCP Other Messages

- **DHCP Decline:**
 - ◆ Used by a client to reject the offer made by a server and must restart the leasing process.
- **DHCP Nack:**
 - ◆ Used by a server informing that cannot satisfy the received request (*DHCP Request*).
- **DHCP Release:**
 - ◆ Used by a client informing the server that no longer requires an address. The lease is terminated.
- **DHCP Inform:**
 - ◆ Used by a client to request additional information after receiving an address.



DHCP Release

| No. | Time | Source | Destination | Protocol | Info |
|------|-----------|----------------|---------------|----------|--------------|
| 1330 | 24.011686 | 193.136.93.228 | 193.136.92.65 | DHCP | DHCP Release |

► Frame 1330 (342 bytes on wire, 342 bytes captured)
► Ethernet II, Src: 00:1d:ba:c0:a2:8e (00:1d:ba:c0:a2:8e), Dst: 00:d0:b7:17:5b:6d (00:d0:b7:17:5b:6d)
► Internet Protocol, Src: 193.136.93.228 (193.136.93.228), Dst: 193.136.92.65 (193.136.92.65)
► User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
▽ Bootstrap Protocol
 Message type: Boot Request (1)
 Hardware type: Ethernet
 Hardware address length: 6
 Hops: 0
 Transaction ID: 0xc099a870
 Seconds elapsed: 0
 Bootp flags: 0x0000 (Unicast)
 Client IP address: 193.136.93.228 (193.136.93.228)
 Your (client) IP address: 0.0.0.0 (0.0.0.0)
 Next server IP address: 0.0.0.0 (0.0.0.0)
 Relay agent IP address: 0.0.0.0 (0.0.0.0)
 Client MAC address: 00:1d:ba:c0:a2:8e (00:1d:ba:c0:a2:8e)
 Client hardware address padding: 000000000000000000000000
 Server host name not given
 Boot file name not given
 Magic cookie: (OK)
 Option: (t=53,l=1) DHCP Message Type = DHCP Release
 Option: (t=54,l=4) DHCP Server Identifier = 193.136.92.65
 Option: (t=12,l=15) Host Name = "salvador-laptop"
 End Option
 Padding



DHCP Inform

| No. | Time | Source | Destination | Protocol | Info |
|------|------------|----------------|-----------------|----------|-------------|
| 3117 | 18.9.105.1 | 193.136.93.122 | 255.255.255.255 | DHCP | DHCP Inform |
| 4107 | 65.374546 | 193.136.93.173 | 255.255.255.255 | DHCP | DHCP Inform |
| 5446 | 86.143470 | 193.136.93.102 | 255.255.255.255 | DHCP | DHCP Inform |

► Frame 4107 (342 bytes on wire, 342 bytes captured)
► Ethernet II, Src: d0:df:9a:cb:d1:3c (d0:df:9a:cb:d1:3c), Dst: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
► Internet Protocol, Src: 193.136.93.173 (193.136.93.173), Dst: 255.255.255.255 (255.255.255.255)
► User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)

▼ Bootstrap Protocol

Message type: Boot Request (1)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0xfb8eebf9
Seconds elapsed: 0

► Bootp flags: 0x8000 (Broadcast)
Client IP address: 193.136.93.173 (193.136.93.173)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: d0:df:9a:cb:d1:3c (d0:df:9a:cb:d1:3c)
Client hardware address padding: 000000000000000000000000
Server host name not given
Boot file name not given
Magic cookie: (OK)

► Option: (t=53,l=1) DHCP Message Type = DHCP Inform
► Option: (t=61,l=7) Client identifier
► Option: (t=12,l=7) Host Name = "IT-TOSH"
► Option: (t=60,l=8) Vendor class identifier = "MSFT 5.0"
► Option: (t=55,l=13) Parameter Request List

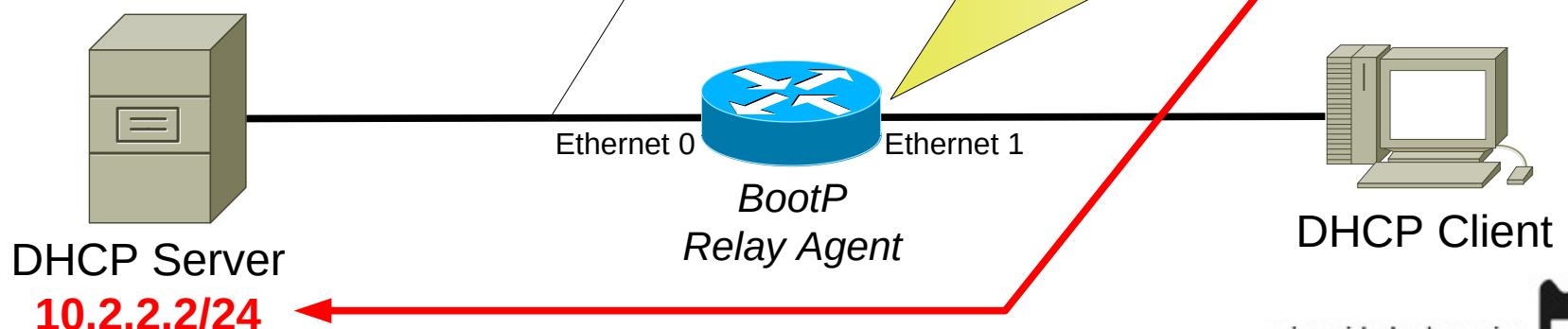
End Option
Padding

▼ Option: (t=55,l=13) Parameter Request List
Option: (55) Parameter Request List
Length: 13
Value: 010F03062C2E2F1F2179F92BFC
1 = Subnet Mask
15 = Domain Name
3 = Router
6 = Domain Name Server
44 = NetBIOS over TCP/IP Name Server
46 = NetBIOS over TCP/IP Node Type
47 = NetBIOS over TCP/IP Scope
31 = Perform Router Discover
33 = Static Route
121 = Classless Static Route
249 = Private/Classless Static Route (Microsoft)
43 = Vendor-Specific Information
252 = Private/Proxy autodiscovery



DHCP in Complex Environments

- In complex network environments where one (or more) DHCP server provide addresses to multiple (V)LAN.
 - Router must have a “BootP Relay Agent” configured and active.
 - Router redirects the client DHCP (broadcast) packets to DHCP server(s) using unicast,
 - Append information of the network/interface where it received the DHCP packet from client.
 - Router redirects server responses to the client.
 - From the client point of view, the Router behaves like a DHCP server.
- Multiple VLAN require multiple pools of addresses at server(s).
 - When using multiple DHCP servers, pools must be disjoint.



NAT and PAT

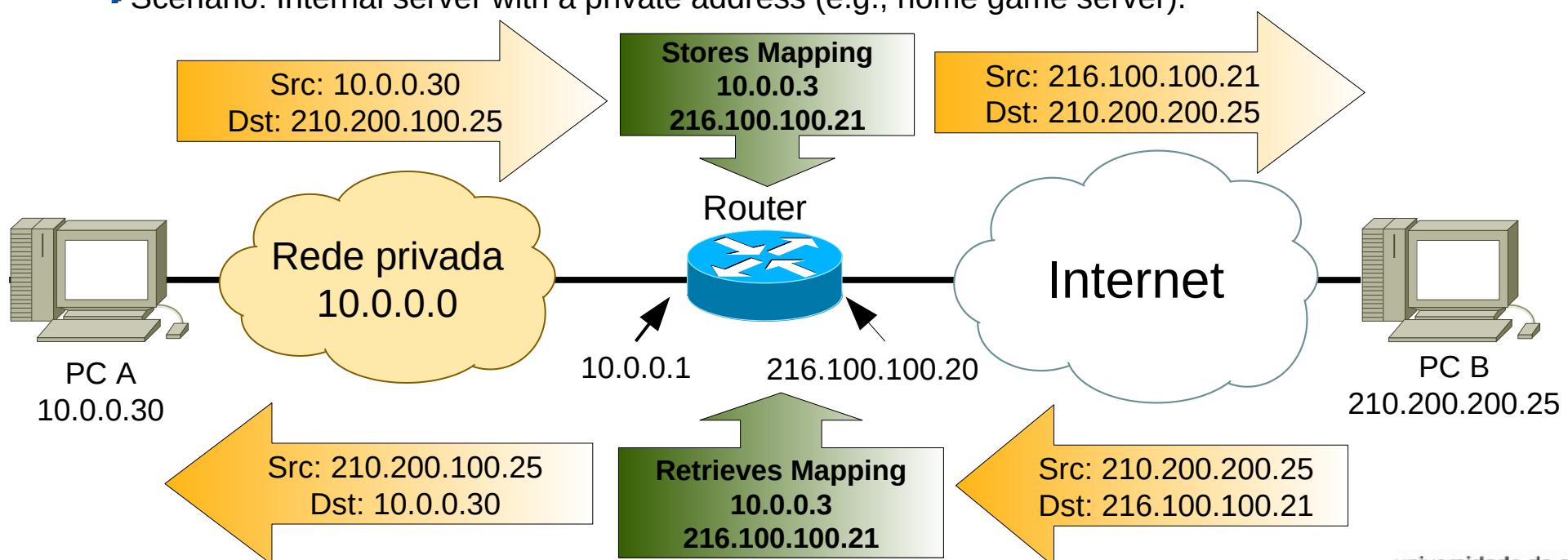
NAT (Network Address Translation) e PAT (Port Address Translation)

- NAT – Translates private address into public addresses.
- PAT – Translates address and also UDP/TCP ports.
 - ◆ ICMP does not have ports. ICMP identifier field is used instead.
 - ◆ Also called NAPT (Network Address and Port Translation)
- Mapping between a private and public address may be dynamic or static.
- Allows a LAN that has a limited number of IPv4 public address allow the connectivity of many internal host to the Internet.
 - ◆ The available IPv4 addresses are called the address pool.
 - ◆ A packet passing from a private network to a public network will have its IPV4 source address (and UDP/TCP port) changed to one of the available IPv4 public addresses (and ports).
 - ◆ That change will be stored on the device on the boundary between the private and public network (Router, Firewall or Security Appliance).
 - ◆ It's called mapping or translation table.
 - ◆ The answer to that packet will have a reverse change.

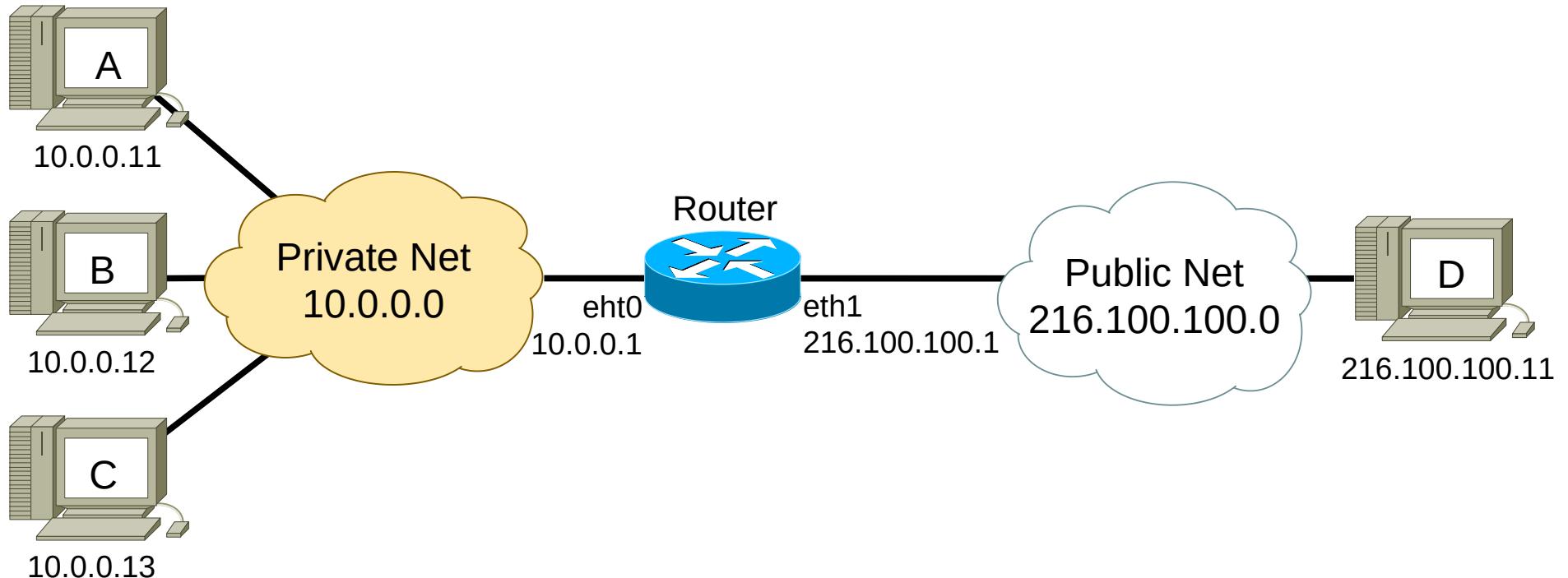


NAT/PAT Mapping

- Dynamic Mapping:
 - ◆ The choice of public address (and port) and mapping to the private address (and port) is done automatically by the Router when it receives a packet from an inside host.
 - ◆ An external host cannot initiate a conversation with a inside host.
 - ◆ May respond to conversation initiated from an inside host.
- Static Mapping:
 - ◆ The choice of public address (and port) and mapping to the private address (and port) is done by configuration.
 - ◆ Allows an external host to initiate a conversation with an internal host with a private address.
 - ◆ External host contacts the public address/port statically mapped to the private address/port.
 - ◆ Scenario: Internal server with a private address (e.g., home game server).



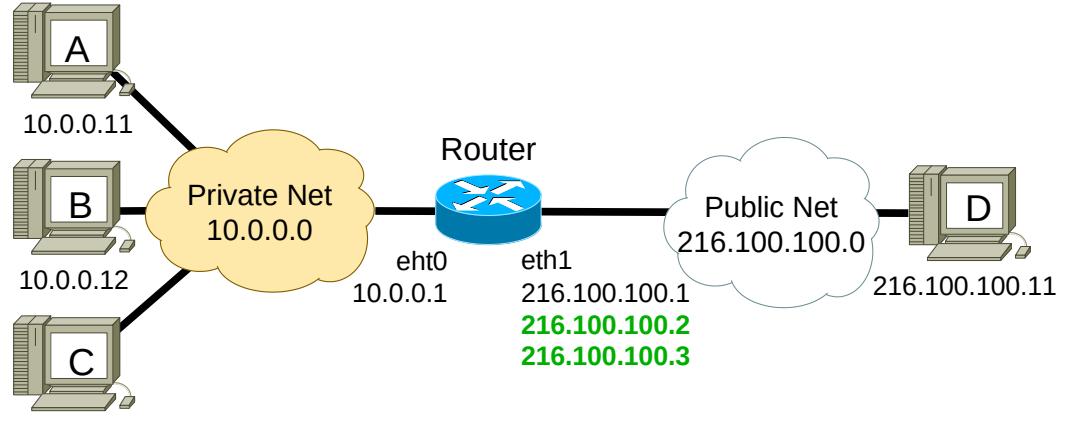
Example – NAT (1)



- Router configures with Dynamic NAT.
- Public IPv4 addresses:
 - ◆ 216.100.100.2 and 216.100.100.3 to NAT mappings,
 - ◆ 216.100.100.1 to be used by the interface.
 - ◆ The IPv4 on the interface may also be used for mapping.



Example – NAT (2)



Ping from 10.0.0.11 to 216.100.100.11:

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|----------------|----------------|----------|--------|---|
| 6 | 15.892528 | 10.0.0.11 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x6c3a, seq=1/256, ttl=64 |
| 7 | 15.911436 | 216.100.100.11 | 10.0.0.11 | ICMP | 98 | Echo (ping) reply id=0x6c3a, seq=1/256, ttl=63 |
| 8 | 16.912087 | 10.0.0.11 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x6d3a, seq=2/512, ttl=64 |
| 9 | 16.932449 | 216.100.100.11 | 10.0.0.11 | ICMP | 98 | Echo (ping) reply id=0x6d3a, seq=2/512, ttl=63 |
| 10 | 17.933103 | 10.0.0.11 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x6f3a, seq=3/768, ttl=64 |
| 11 | 17.952490 | 216.100.100.11 | 10.0.0.11 | ICMP | 98 | Echo (ping) reply id=0x6f3a, seq=3/768, ttl=63 |
| 12 | 18.954005 | 10.0.0.11 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x703a, seq=4/1024, ttl=64 |
| 13 | 18.974316 | 216.100.100.11 | 10.0.0.11 | ICMP | 98 | Echo (ping) reply id=0x703a, seq=4/1024, ttl=63 |
| 14 | 19.975028 | 10.0.0.11 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x713a, seq=5/1280, ttl=64 |
| 15 | 19.986293 | 216.100.100.11 | 10.0.0.11 | ICMP | 98 | Echo (ping) reply id=0x713a, seq=5/1280, ttl=63 |

Private Network

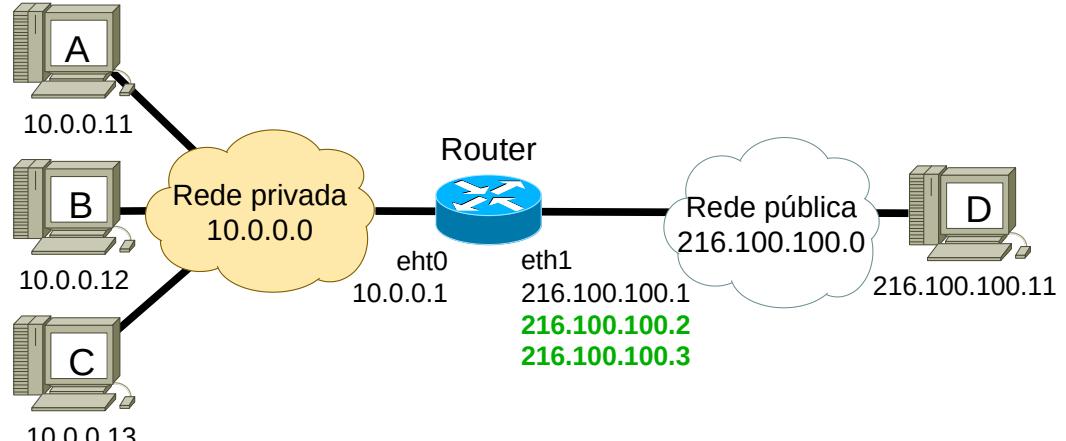
| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|----------------|----------------|----------|--------|---|
| 2 | 3.913049 | 216.100.100.2 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x6c3a, seq=1/256, ttl=63 |
| 3 | 3.913320 | 216.100.100.11 | 216.100.100.2 | ICMP | 98 | Echo (ping) reply id=0x6c3a, seq=1/256, ttl=64 |
| 4 | 4.934041 | 216.100.100.2 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x6d3a, seq=2/512, ttl=63 |
| 5 | 4.934405 | 216.100.100.11 | 216.100.100.2 | ICMP | 98 | Echo (ping) reply id=0x6d3a, seq=2/512, ttl=64 |
| 6 | 5.954132 | 216.100.100.2 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x6f3a, seq=3/768, ttl=63 |
| 7 | 5.954324 | 216.100.100.11 | 216.100.100.2 | ICMP | 98 | Echo (ping) reply id=0x6f3a, seq=3/768, ttl=64 |
| 8 | 6.975911 | 216.100.100.2 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x703a, seq=4/1024, ttl=63 |
| 9 | 6.976473 | 216.100.100.11 | 216.100.100.2 | ICMP | 98 | Echo (ping) reply id=0x703a, seq=4/1024, ttl=64 |
| 10 | 7.987741 | 216.100.100.2 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x713a, seq=5/1280, ttl=63 |
| 11 | 7.988265 | 216.100.100.11 | 216.100.100.2 | ICMP | 98 | Echo (ping) reply id=0x713a, seq=5/1280, ttl=64 |

Public Network

| Router#show ip nat translation | | | |
|--------------------------------|---------------|--------------|---------------|
| Pro | Inside global | Inside local | Outside local |
| | --- | 10.0.0.11 | --- |
| | 216.100.100.2 | | --- |



Example – NAT (3)



Ping from 10.0.0.12 to 216.100.100.11:

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|---------------|----------------|----------------|----------|--------|---|
| → | 53 311.240021 | 10.0.0.12 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x943b, seq=1/256, ttl=64 |
| ← | 54 311.258670 | 216.100.100.11 | 10.0.0.12 | ICMP | 98 | Echo (ping) reply id=0x943b, seq=1/256, ttl=63 |
| → | 56 312.259967 | 10.0.0.12 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x953b, seq=2/512, ttl=64 |
| ← | 57 312.280140 | 216.100.100.11 | 10.0.0.12 | ICMP | 98 | Echo (ping) reply id=0x953b, seq=2/512, ttl=63 |
| → | 58 313.281645 | 10.0.0.12 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x963b, seq=3/768, ttl=64 |
| ← | 59 313.302003 | 216.100.100.11 | 10.0.0.12 | ICMP | 98 | Echo (ping) reply id=0x963b, seq=3/768, ttl=63 |
| → | 60 314.303181 | 10.0.0.12 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x973b, seq=4/1024, ttl=64 |
| ← | 61 314.323635 | 216.100.100.11 | 10.0.0.12 | ICMP | 98 | Echo (ping) reply id=0x973b, seq=4/1024, ttl=63 |
| → | 62 315.325157 | 10.0.0.12 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x983b, seq=5/1280, ttl=64 |
| ← | 63 315.345519 | 216.100.100.11 | 10.0.0.12 | ICMP | 98 | Echo (ping) reply id=0x983b, seq=5/1280, ttl=63 |

Private Network

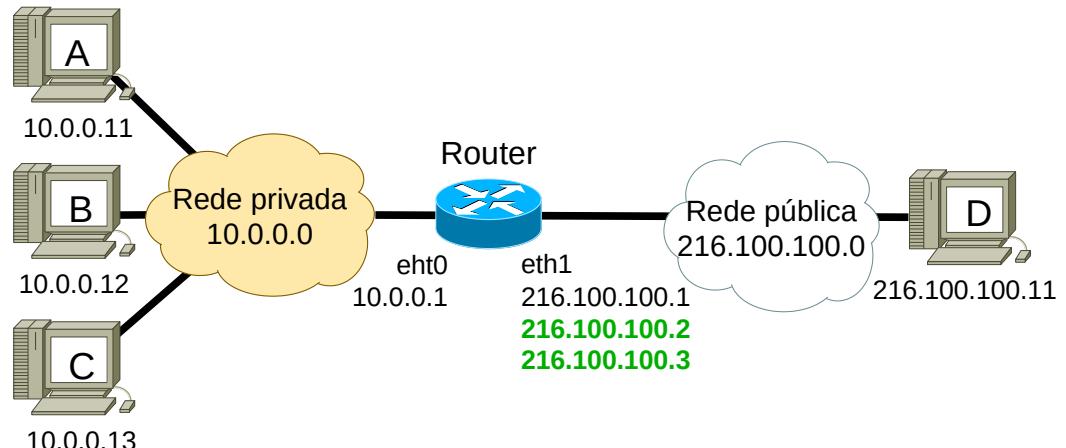
| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|---------------|----------------|----------------|----------|--------|---|
| → | 47 299.260334 | 216.100.100.3 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x943b, seq=1/256, ttl=63 |
| ← | 48 299.260929 | 216.100.100.11 | 216.100.100.3 | ICMP | 98 | Echo (ping) reply id=0x943b, seq=1/256, ttl=64 |
| → | 50 300.281677 | 216.100.100.3 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x953b, seq=2/512, ttl=63 |
| ← | 51 300.282286 | 216.100.100.11 | 216.100.100.3 | ICMP | 98 | Echo (ping) reply id=0x953b, seq=2/512, ttl=64 |
| → | 52 301.303570 | 216.100.100.3 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x963b, seq=3/768, ttl=63 |
| ← | 53 301.304103 | 216.100.100.11 | 216.100.100.3 | ICMP | 98 | Echo (ping) reply id=0x963b, seq=3/768, ttl=64 |
| → | 54 302.325227 | 216.100.100.3 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x973b, seq=4/1024, ttl=63 |
| ← | 55 302.325755 | 216.100.100.11 | 216.100.100.3 | ICMP | 98 | Echo (ping) reply id=0x973b, seq=4/1024, ttl=64 |
| → | 56 303.347148 | 216.100.100.3 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x983b, seq=5/1280, ttl=63 |
| ← | 57 303.347704 | 216.100.100.11 | 216.100.100.3 | ICMP | 98 | Echo (ping) reply id=0x983b, seq=5/1280, ttl=64 |

Public Network

| Router#show ip nat translation | | | |
|--------------------------------|---------------|--------------|---------------|
| Pro | Inside global | Inside local | Outside local |
| --- | 216.100.100.2 | 10.0.0.11 | --- |
| --- | 216.100.100.3 | 10.0.0.12 | --- |



Example – NAT (4)



Ping from 10.0.0.13 to 216.100.100.11:

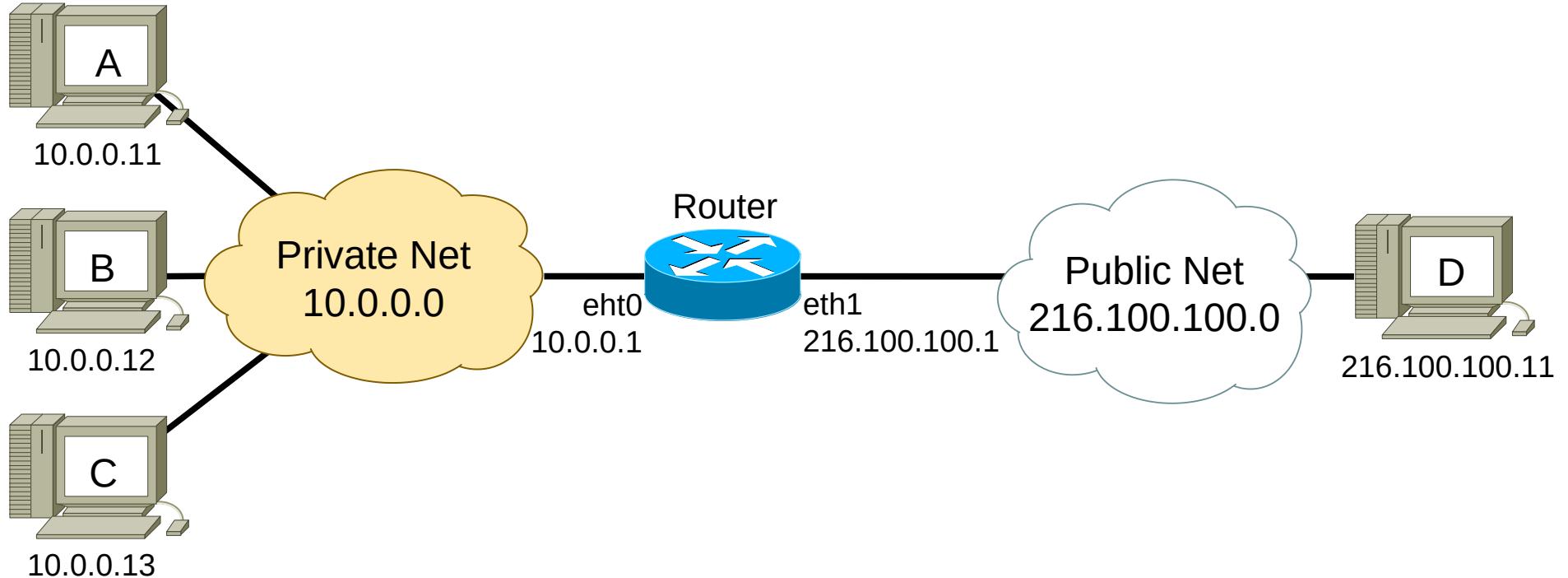
| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|------------|-----------|----------------|----------|--------|---|
| 113 | 506.016226 | 10.0.0.13 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x573c, seq=1/256, ttl=64 |
| 114 | 506.035020 | 10.0.0.1 | 10.0.0.13 | ICMP | 70 | Destination unreachable (Host unreachable) |
| 115 | 507.036014 | 10.0.0.13 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x583c, seq=2/512, ttl=64 |
| 116 | 507.046188 | 10.0.0.1 | 10.0.0.13 | ICMP | 70 | Destination unreachable (Host unreachable) |
| 117 | 508.047177 | 10.0.0.13 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x593c, seq=3/768, ttl=64 |
| 118 | 508.057193 | 10.0.0.1 | 10.0.0.13 | ICMP | 70 | Destination unreachable (Host unreachable) |
| 119 | 509.058553 | 10.0.0.13 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x5a3c, seq=4/1024, ttl=64 |
| 120 | 509.068436 | 10.0.0.1 | 10.0.0.13 | ICMP | 70 | Destination unreachable (Host unreachable) |
| 121 | 510.069971 | 10.0.0.13 | 216.100.100.11 | ICMP | 98 | Echo (ping) request id=0x5b3c, seq=5/1280, ttl=64 |
| 122 | 510.079907 | 10.0.0.1 | 10.0.0.13 | ICMP | 70 | Destination unreachable (Host unreachable) |

Private Network

- Host C (10.0.0.13) cannot access the public network.
 - All IPv4 public address available on the Router have been mapped to Host A and Host B.
- All NAT mappings have a limited lifetime (*timeout*).
 - After some time without traffic to the public network the mappings will be deleted.



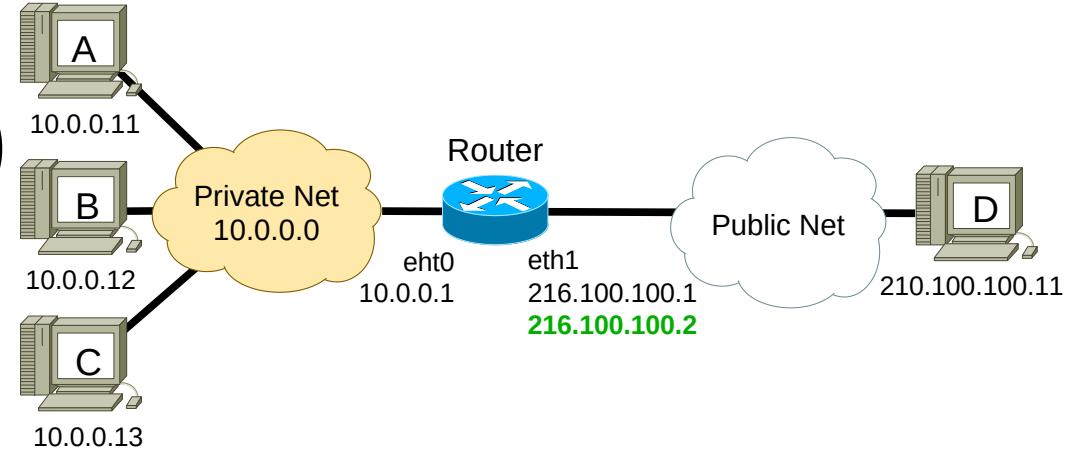
Example – NAT/PAT (1)



- Host D has a UDP server (ECHO) on port 5005.
- Public IPv4 addresses:
 - ◆ 216.100.100.2 and 216.100.100.3 to NAT mappings,
 - ◆ 216.100.100.1 to be used by the interface.



Example – NAT/PAT (2)



Hosts A, B and C access Host D (UDP Port 5005):

| Source | Destination | Protocol | Length | Info |
|----------------|----------------|----------|--------|--------------|
| 10.0.0.11 | 216.100.100.11 | UDP | 98 | 22147 → 5005 |
| 216.100.100.11 | 10.0.0.11 | UDP | 98 | 5005 → 22147 |
| 10.0.0.11 | 216.100.100.11 | UDP | 98 | 22147 → 5005 |
| 216.100.100.11 | 10.0.0.11 | UDP | 98 | 5005 → 22147 |

| Source | Destination | Protocol | Length | Info |
|----------------|----------------|----------|--------|-------------|
| 216.100.100.2 | 216.100.100.11 | UDP | 98 | 1024 → 5005 |
| 216.100.100.11 | 216.100.100.2 | UDP | 98 | 5005 → 1024 |
| 216.100.100.2 | 216.100.100.11 | UDP | 98 | 1024 → 5005 |
| 216.100.100.11 | 216.100.100.2 | UDP | 98 | 5005 → 1024 |

| Source | Destination | Protocol | Length | Info |
|----------------|----------------|----------|--------|--------------|
| 10.0.0.12 | 216.100.100.11 | UDP | 98 | 40521 → 5005 |
| 216.100.100.11 | 10.0.0.12 | UDP | 98 | 5005 → 40521 |
| 10.0.0.12 | 216.100.100.11 | UDP | 98 | 40521 → 5005 |
| 216.100.100.11 | 10.0.0.12 | UDP | 98 | 5005 → 40521 |

| Source | Destination | Protocol | Length | Info |
|----------------|----------------|----------|--------|-------------|
| 216.100.100.2 | 216.100.100.11 | UDP | 98 | 1025 → 5005 |
| 216.100.100.11 | 216.100.100.2 | UDP | 98 | 5005 → 1025 |
| 216.100.100.2 | 216.100.100.11 | UDP | 98 | 1025 → 5005 |
| 216.100.100.11 | 216.100.100.2 | UDP | 98 | 5005 → 1025 |

| Source | Destination | Protocol | Length | Info |
|----------------|----------------|----------|--------|--------------|
| 10.0.0.13 | 216.100.100.11 | UDP | 98 | 61252 → 5005 |
| 216.100.100.11 | 10.0.0.13 | UDP | 98 | 5005 → 61252 |
| 10.0.0.13 | 216.100.100.11 | UDP | 98 | 61252 → 5005 |
| 216.100.100.11 | 10.0.0.13 | UDP | 98 | 5005 → 61252 |

| Source | Destination | Protocol | Length | Info |
|----------------|----------------|----------|--------|-------------|
| 216.100.100.2 | 216.100.100.11 | UDP | 98 | 1026 → 5005 |
| 216.100.100.11 | 216.100.100.2 | UDP | 98 | 5005 → 1026 |
| 216.100.100.2 | 216.100.100.11 | UDP | 98 | 1026 → 5005 |
| 216.100.100.11 | 216.100.100.2 | UDP | 98 | 5005 → 1026 |

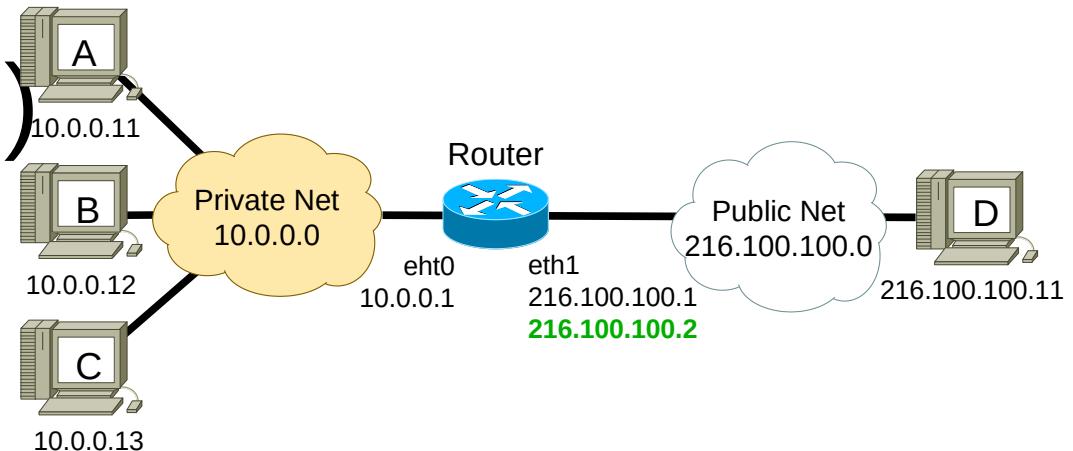
Private Network

Public Network

- Mapping choices by the Router depends on local algorithm, is not defined by standards.
- All hosts were mapped to IPv4 216.100.100.2.
 - ◆ Host A used the UDP client port 22147, and was mapped to port 1024.
 - ◆ Host B used the UDP client port 40521, and was mapped to port 1025.
 - ◆ Host C used the UDP client port 61252, and was mapped to port 1026.



Example – NAT/PAT (3)



Hosts A, B and C access Host D (UDP Port 5005):

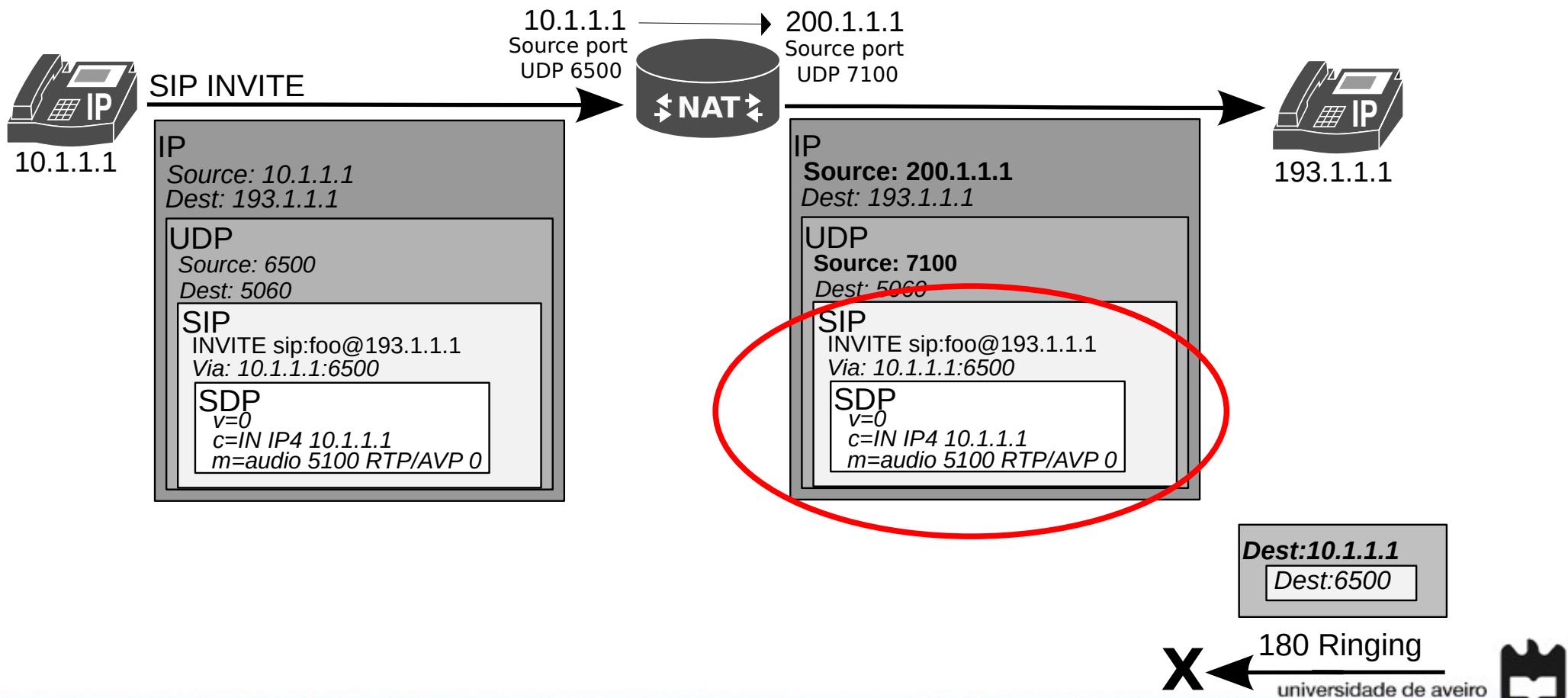
| Router#show ip nat translation | | | | |
|--|---------------|--------------|---------------------|---------------------|
| Pro | Inside global | Inside local | Outside local | Outside global |
| udp 216.100.100.2:1024 10.0.0.11:22147 | | | 216.100.100.11:5005 | 216.100.100.11:5005 |
| udp 216.100.100.2:1025 10.0.0.12:40521 | | | 216.100.100.11:5005 | 216.100.100.11:5005 |
| udp 216.100.100.2:1026 10.0.0.13:61252 | | | 216.100.100.11:5005 | 216.100.100.11:5005 |

- All hosts were mapped to IPv4 address 216.100.100.2.
- Host A used the UDP client port 22147, and was mapped to port 1024.
- Host B used the UDP client port 40521, and was mapped to port 1025.
- Host C used the UDP client port 61252, and was mapped to port 1026.



Some Protocols Require Translation at the Application Level

- Some protocols (e.g., SIP) require the translation of addresses and ports also at the application protocol level.
 - ◆ Very computational demanding and not all devices allow it.



IPv6 Addressing

IPv6 Background

- IETF IPv6 WG began to work on a solution to solve addressing growth issues in early 1990s
- Reasons to late deployment
 - ◆ Classless Inter-Domain Routing (CIDR) and Network address translation (NAT) were developed
 - ◆ Investments on field equipments (not IPv6 aware) had to reach the predicted “return of investment”
 - ◆ Massive re-equipment price



IPv6 Features

- Larger address space enabling:
 - ◆ Global reachability, flexibility, aggregation, multihoming, autoconfiguration, “plug and play” and renumbering
- Simpler header enabling:
- Routing efficiency, performance and forwarding rate scalability
- Improved option support

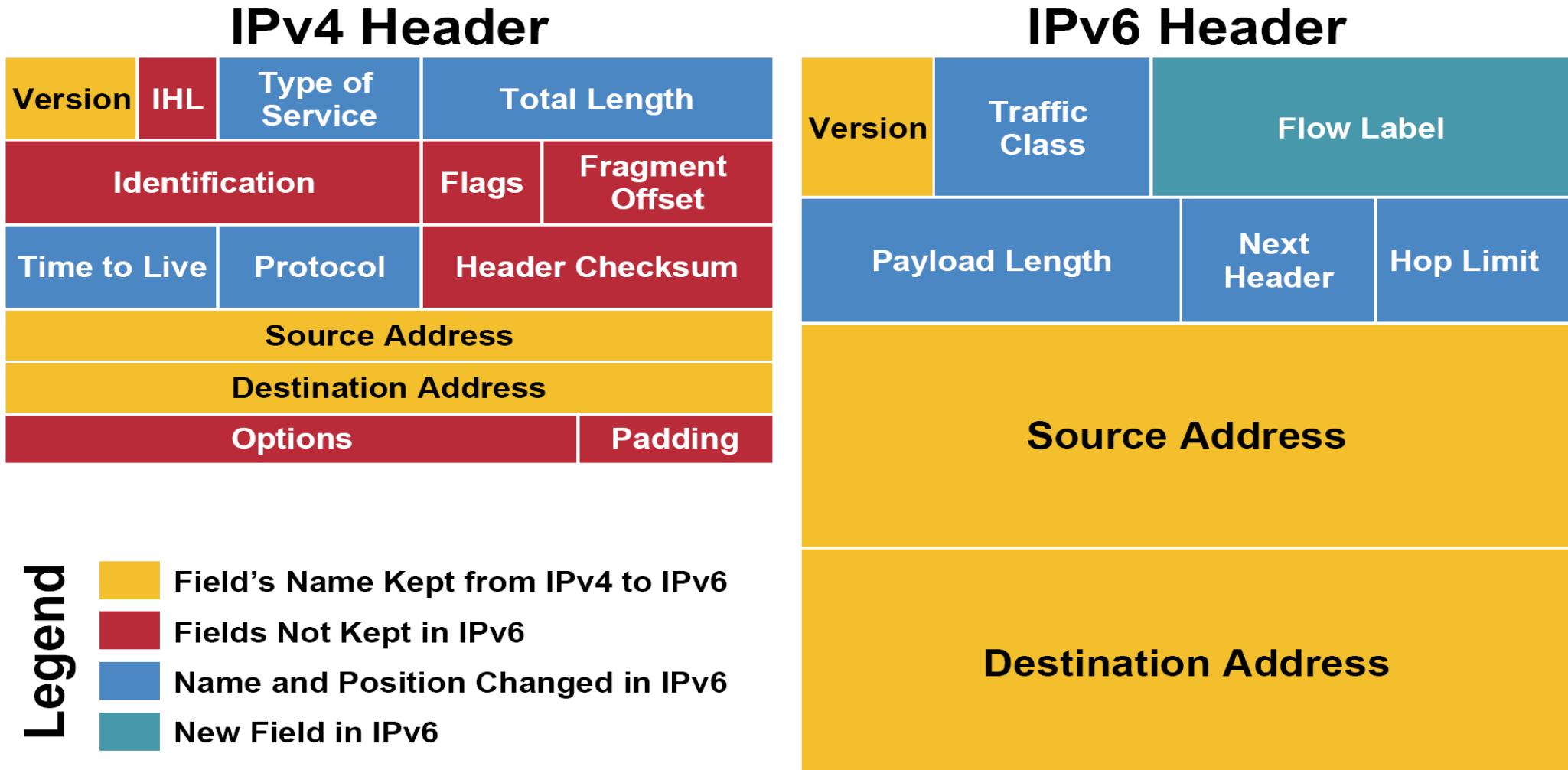


IPv6 Addressing

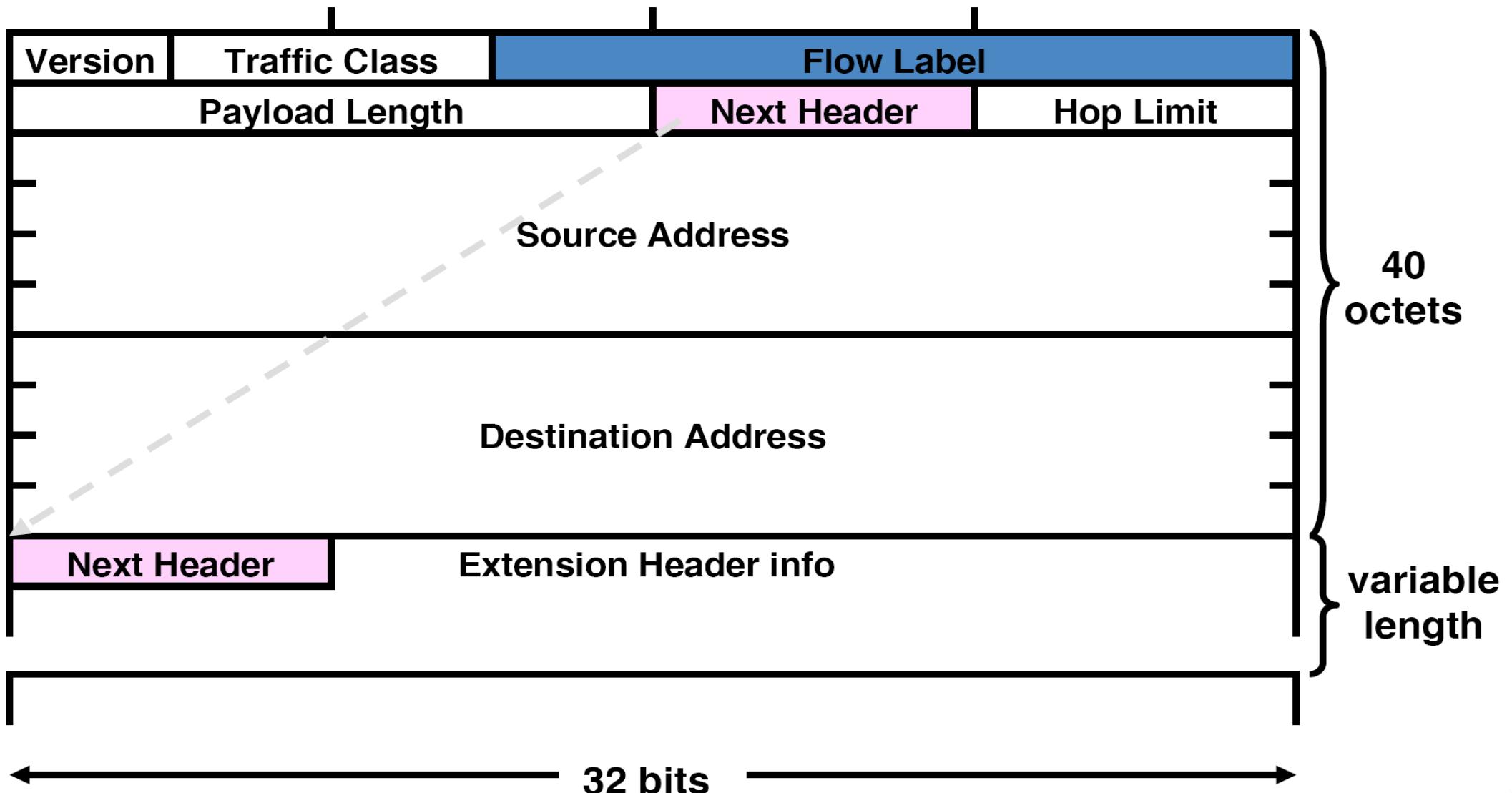
- IPv4: 4bytes/32 bits
 - ◆ ~ 4,294,967,296 possible addresses
- IPv6: 16bytes/128 bits
 - ◆ 340,282,366,920,938,463,463,374,607,431,768,211,456 possible addresses
- Representation
 - ◆ 16-bit hexadecimal numbers
 - ◆ Hex numbers are not case sensitive
 - ◆ Numbers are separated by (:)
 - ◆ Abbreviations are possible
 - Leading zeros in contiguous block could be represented by (::)
 - Example:
 - 2001:0db8:0000:130F:0000:0000:087C:140B = 2001:0db8:0:130F::87C:140B
 - Double colon only appears once in the address
 - ◆ Address's prefix is represented as: prefix/mask_number_of_bits



IPv4 vs. IPv6 Headers



IPv6 Header Format



IPv6 Addressing Model

- Interface have multiple addresses
- Addresses have scope:
 - ◆ Link Local
 - ◆ Valid within the same LAN or link
 - ◆ Unique Local
 - ◆ Valid within the same private domain
 - ◆ Can not be used in Internet
 - ◆ Global
- Addresses have lifetime
 - ◆ Valid and preferred lifetime



Types of IPv6 Addresses

- Unicast
 - ◆ Address of a single interface.
 - ◆ One-to-one delivery to single interface
- Multicast
 - ◆ Address of a set of interfaces.
 - ◆ One-to-many delivery to all interfaces in the set
- Anycast
 - ◆ Address of a set of interfaces.
 - ◆ One-to-one-of-many delivery to a single interface in the set that is closest
- No more broadcast addresses

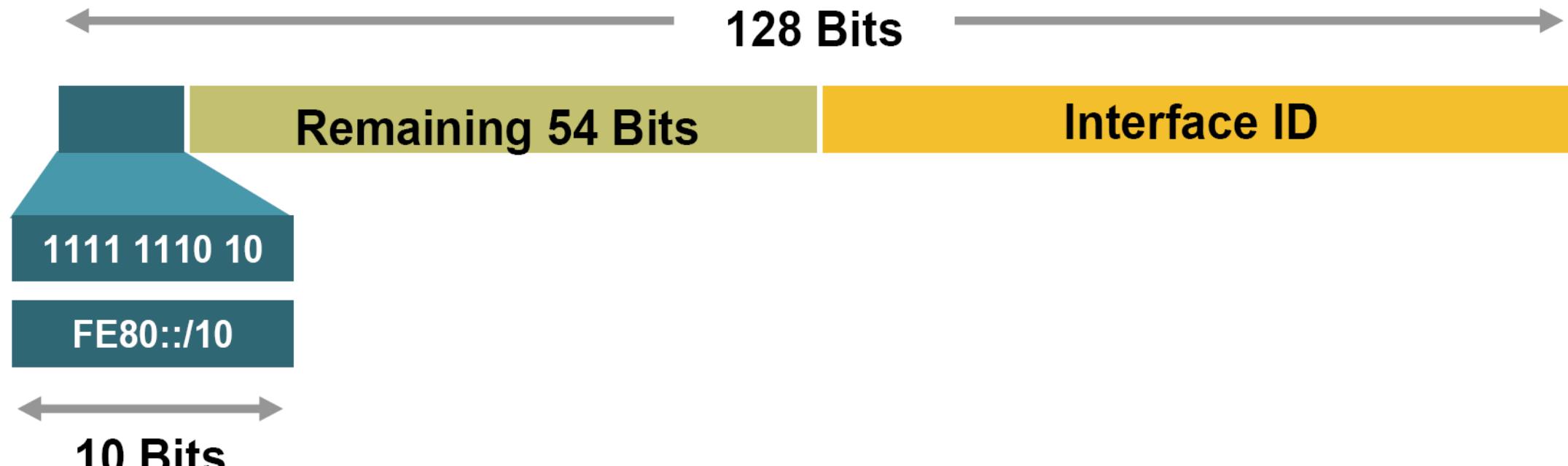


IPv6 Addressing

| Type | Binary | Hexadecimal |
|-------------------------------------|------------------------|----------------------|
| <i>Global Unicast Address</i> | 0010 | 2 |
| <i>Link-Local Unicast Address</i> | 1111 1110 10 | FE80::/10 |
| <i>Unique-Local Unicast Address</i> | 1111 1100 1111 1101 | FC00::/8 FD00::/8 |
| <i>Multicast Address</i> | 1111 1111 | FF00::/16 |



Link-Local Address

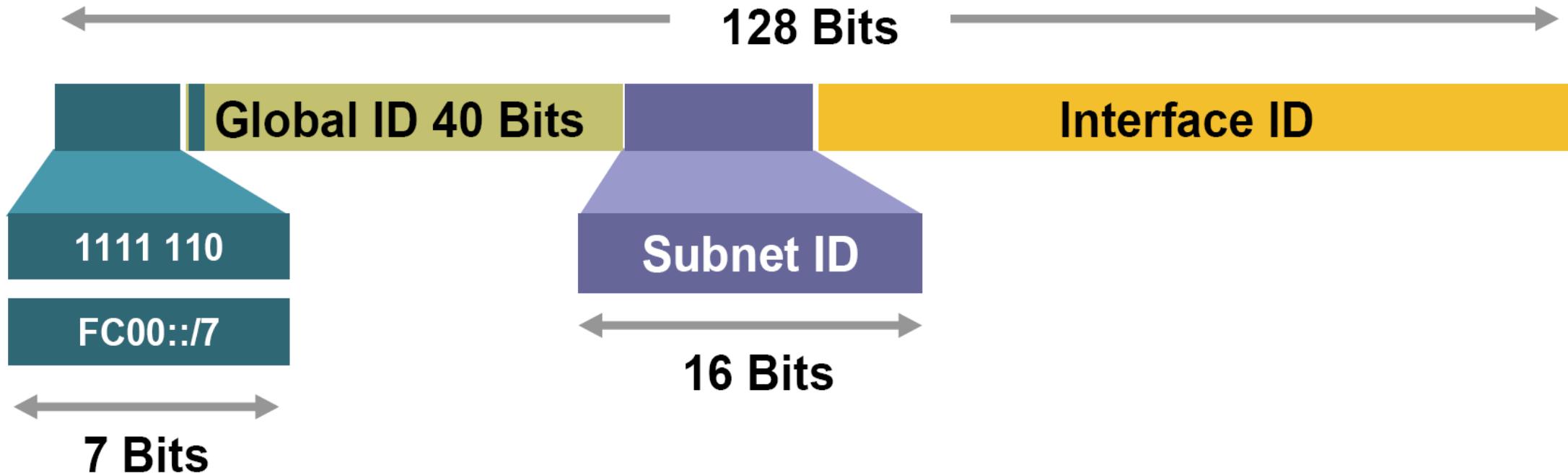


10 Bits

- Used For:
 - ◆ Mandatory address for local communication between two IPv6 devices
 - ◆ Next-Hop calculation in Routing Protocols
- Automatically assigned as soon as IPv6 is enabled
- Remaining 54 bits could be Zero or any manual configured value



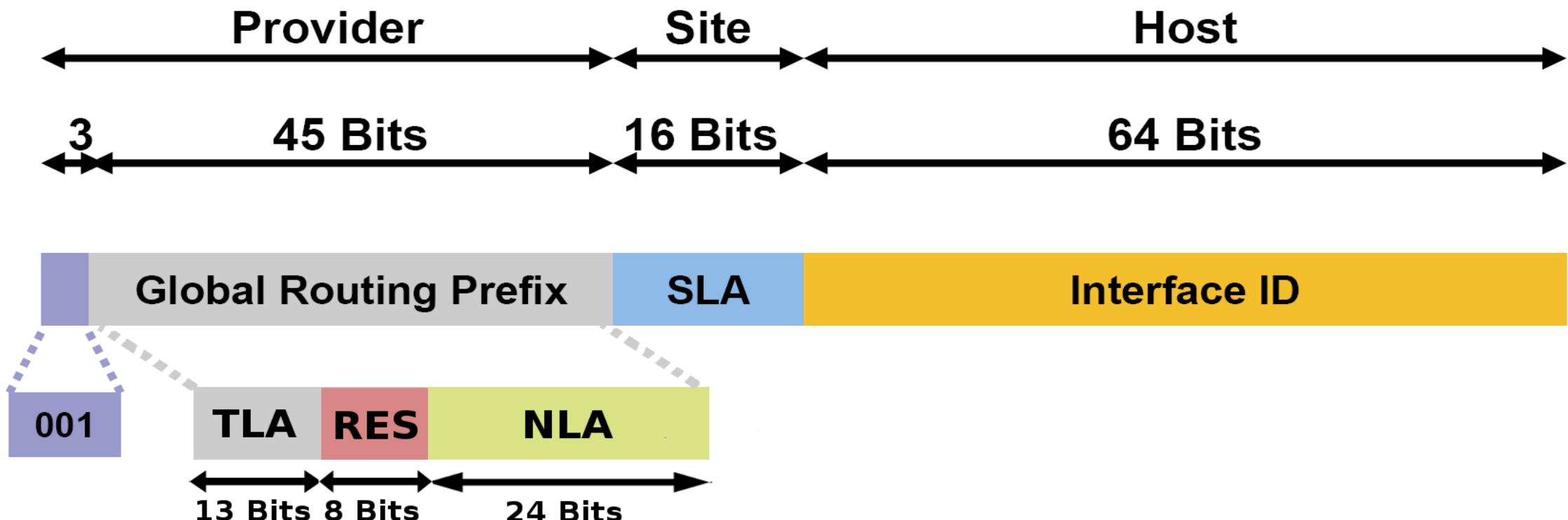
Unique-Local Address



- Used For:
 - ◆ Local communications
 - ◆ Inter-site VPNs
- Can be routed only within the same Autonomous System
 - ◆ Can not be used on the Internet



Global Unicast Addresses



- LA, NLA and SLA used for hierarchical addressing
 - ◆ TLA - Top-Level Aggregation
 - ◆ RES – Reserved (must be zero)
 - ◆ NLA - Next-Level Aggregation Identifier
 - ◆ SLA - Site-Level Aggregation Identifier



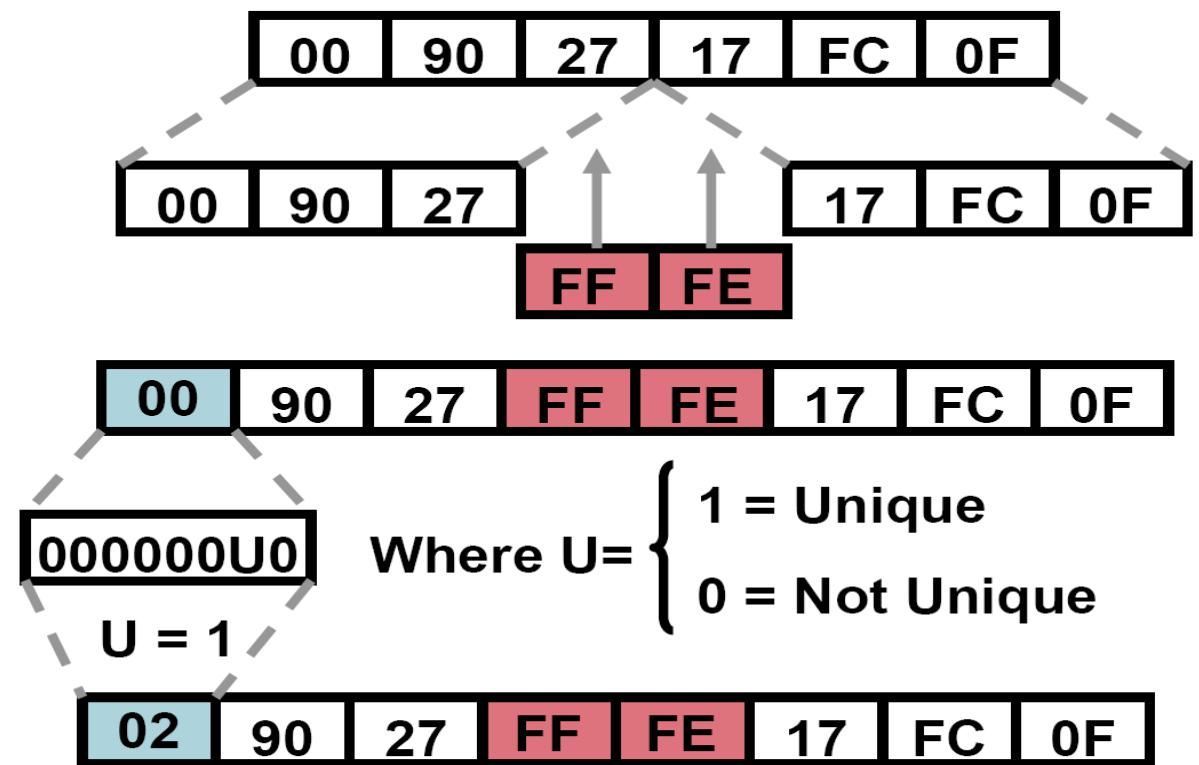
IPv6 Interface Identifier

- Lowest-Order 64-Bit field of any address:
 - ◆ Auto-configured from a 64-bit EUI-64, or expanded from a 48-bit MAC address (e.g. Ethernet address)
 - ◆ Auto-generated pseudo-random number
 - ◆ Assigned via DHCP
 - ◆ Manually configured



MAC to Interface ID (EUI-64 format)

- Stateless auto-configuration
- Expands the 48 bit MAC address to 64 bits by inserting FFFE into the middle 16 bits
- To make sure that the chosen address is from a unique Ethernet MAC address
 - “u”bit is set to 1 for global scope
 - “u”bit is set to 0 for local scope



Anycast Address

IPv6 Address



- Address that is assigned to a set of interfaces
 - ◆ Typically belong to different nodes
- A packet sent to an Anycast address is delivered to the closest interface (determined by routing and timings)
- Anycast addresses can be used only by routers, not hosts
- Must not be used as the source address of an IPv6 packet
- Nodes to which the anycast address is assigned must be explicitly configured to recognize that the address is an Anycast address



Multicast Addresses



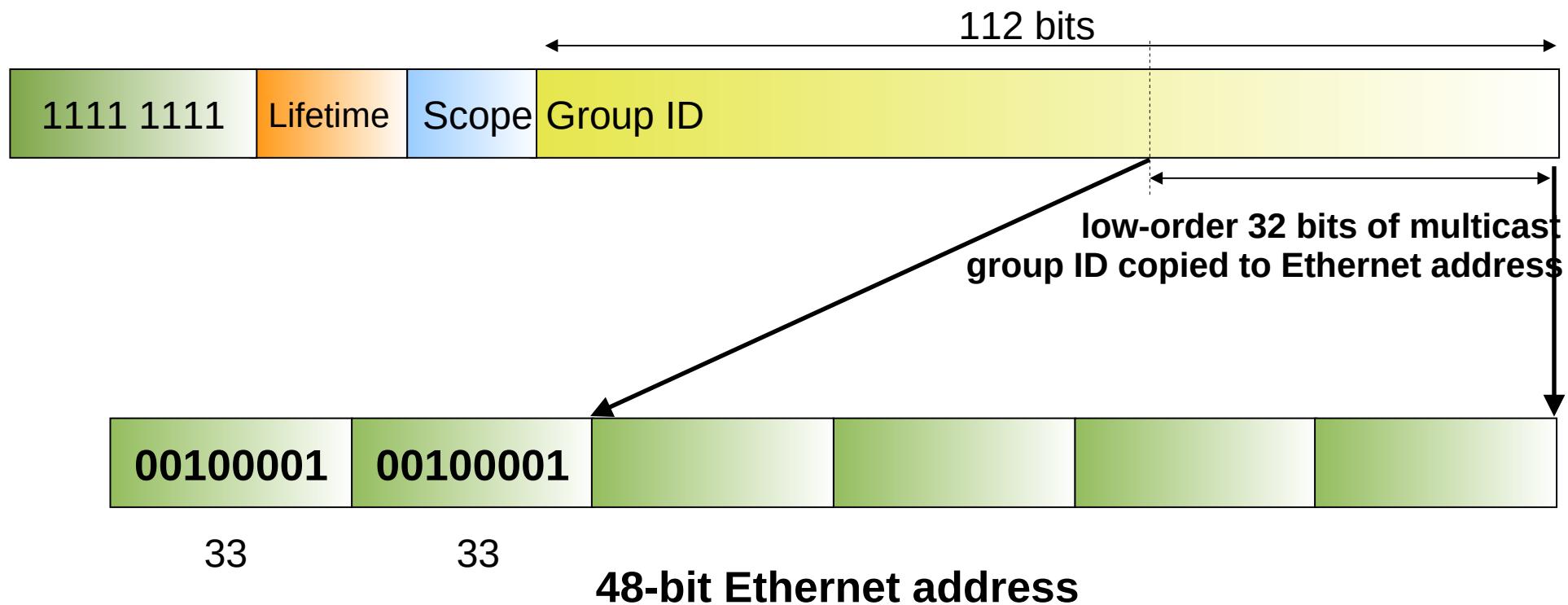
| Lifetime | |
|----------|--------------|
| 0 | If Permanent |
| 1 | If Temporary |

| Scope | |
|-------|--------------|
| 1 | Node |
| 2 | Link |
| 5 | Site |
| 8 | Organization |
| E | Global |

- Multicast addresses have a prefix FF00::/8
- The second byte defines the lifetime and scope of the multicast address.



Mapping a IPv6 Multicast Address to Ethernet Address



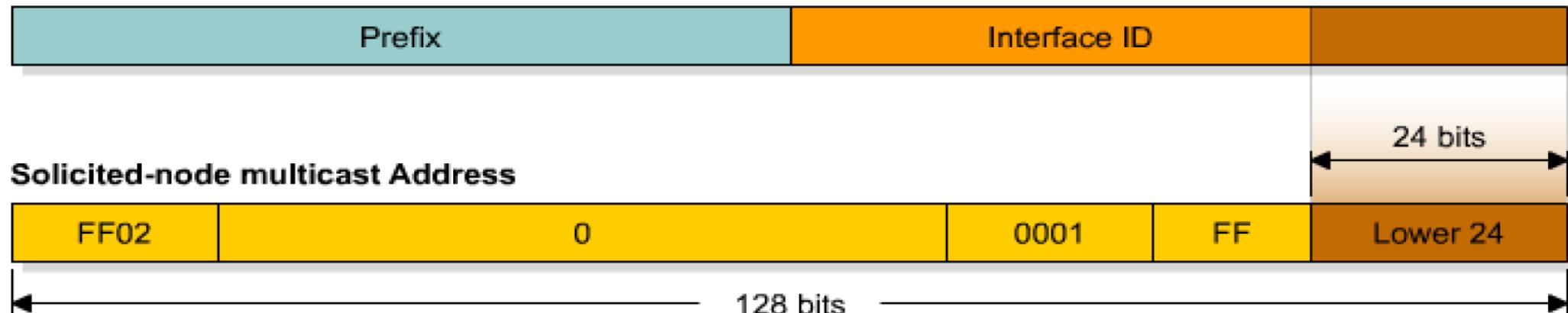
Common Multicast Addresses

- Node Scope
 - ✚ FF01:::1 All Nodes Address (Node scope)
 - ✚ FF01:::2 All Routers Address (Node scope)
- Link Scope
 - ✚ FF02::1 All Nodes Address (Node scope)
 - ✚ FF02::2 All Routers Address
 - ✚ FF02::4 DVMRP Routers
 - ✚ FF02::5 OSPF IGP
 - ✚ FF02::6 OSPF IGP Designated Routers
 - ✚ FF02::9 RIP Routers
 - ✚ FF02::B Mobile-Agents
 - ✚ FF02::D All PIM Routers
 - ✚ FF02::E RSVP-ENCAPSULATION
 - ✚ FF02::16 All MLDv2-capable routers
 - ✚ FF02:::1:2 All DHCP agents



Solicited-Node Multicast Address

IPv6 Address



- For each unicast and anycast address configured there is a corresponding solicited-node multicast
- FF02::1:FF:<interface ID's lower 24 bits>
- This address has link local significance only
- Used in “Neighbour Solicitation Messages”
 - ◆ MAC/Physical addresses resolution
 - ◆ Duplicate Address Detection (DAD)
 - ◆ Random or assigned interface IDs may result in equal global/link addresses



Physical Addresses Resolution

- In IPv6 ARP does not exist anymore.
- ARP table is now called **NDP table**
 - ◆ NDP: Neighbor Discovery Protocol
 - ◆ Maintains a list of known neighbors (IPv6 addresses and MAC addresses).
- Uses ICMPv6 “Neighbor Solicitation” and “Neighbor Advertisement” messages.
 - ◆ To resolve an address a Neighbor Solicitation message is sent to the Solicited-Node multicast address of the target machine (IPv6 address).
 - ◆ Response is sent in unicast using a Neighbor Advertisement message.



ICMPv6

- Internet Control Message Protocol version 6 (ICMPv6) is the implementation ICMP for IPv6
 - ◆ RFC 4443
 - ◆ ICMPv6 is an integral part of IPv6.
- Have the same functionalities of ICMP, plus:
 - ◆ Replaces and enhances ARP,
 - ◆ ICMPv6 implements a Neighbor Discovery Protocol (NDP),
 - ◆ Hosts use it to discover routers and perform auto configuration of addresses,
 - ◆ Used to perform Duplicate Address Detection (DAD),
 - ◆ Used to test reachability of neighbors.



Neighbor Discovery

- Neighbor discovery uses ICMPv6 messages, originated from node on link local with hop limit of 255
- Consists of IPv6 header, ICMPv6 header, neighbor discovery header, and neighbor discovery options
- Five neighbor discovery messages
 - ◆ Router solicitation (ICMPv6 type 133)
 - ◆ Router advertisement (ICMPv6 type 134)
 - ◆ Neighbor solicitation (ICMPv6 type 135)
 - ◆ Neighbor advertisement (ICMPv6 type 136)
 - ◆ Redirect (ICMPV6 type 137)



Router Solicitation

- Host send to inquire about presence of a router on the link
- Send to all routers multicast address of FF02::2 (all routers multicast address)
- Source IP address is either link local address or unspecified IPv6 address

Router advertisement

- Sent out by routers periodically, or in response to a router solicitation
- Includes auto-configuration information
- Includes a "preference level" for each advertised router address
- Also includes a "lifetime" field



Neighbor Solicitation

- Send to discover link layer address of IPv6 node
- IPv6 header, source address is set to unicast address of sending node, or :: for DAD
- Destination address is set to
 - ◆ Unicast address for reachability
 - ◆ Solicited node multicast for address resolution and DAD



Neighbor Advertisement

- Response to neighbor solicitation message
- Also send to inform change of link layer address

Redirect

- Redirect is used by a router to signal the reroute of a packet to a better router



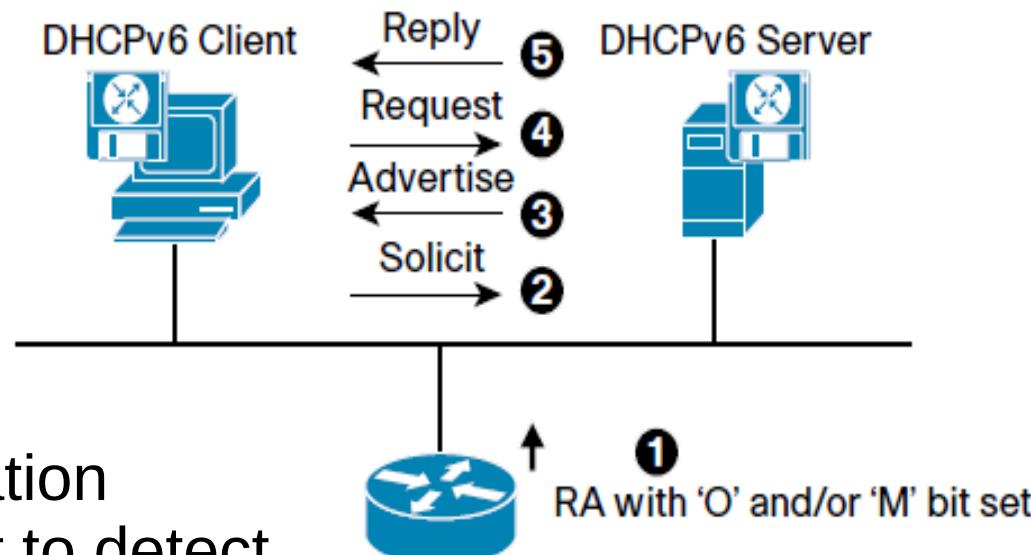
Auto-configuration

- Stateless
 - ◆ A node on the link can automatically configure global IPv6 addresses by appending its interface identifier (64 bits) to the prefixes (64 bits) included in the Router Advertisement messages
 - ◆ Additional/Other network information may be obtained
 - ◆ Additional fields in Router Advertisement messages,
 - ◆ Using a stateless DHCPv6 server.
- Stateful
 - ◆ Addresses are obtained using DHCPv6.
- The default gateway may send two configurable flags in Router Advertisements (RA)
 - ◆ Other flag bit: client can use DHCPv6 to retrieve other configuration parameters (e.g.: DNS server addresses)
 - ◆ Managed flag bit: client may use DHCPv6 to retrieve a Managed IPv6 address from a server



DHCPv6

- Basic DHCPv6 concept is similar to DHCP for IPv4.
- If a client wishes to receive configuration parameters, it will send out a request to detect available DHCPv6 servers.
 - ◆ This done through the “Solicit” and “Advertise” messages.
 - ◆ Well known DHCPv6 Multicast addresses are used for this process.
- Next, the DHCPv6 client will “Request” parameters from an available server which will respond with the requested information with a “Reply” message.
- DHCPv6 relaying works differently from DHCP for IPv4 relaying
 - ◆ Relay agent will encapsulate the received messages from the directly connected DHCPv6 client (RELAY-FORW message)
 - ◆ Forward these encapsulated DHCPv6 packets towards the DHCPv6 server.
 - ◆ In the opposite direction, the Relay Agent will decapsulate the packets received from the central DHCPv6 Server (RELAY-REPL message).

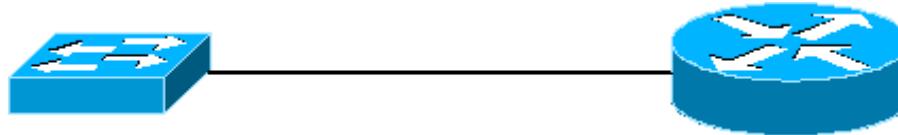


Multicast Listener Discovery (MLD)

- MLD permits the creation/management of multicast groups
- MLD is used by an IPv6 router to:
 - ◆ Discover the presence of multicast listeners on directly attached links
 - ◆ And to discover which multicast addresses are of interest to those neighboring nodes
 - ◆ Report interest in router specific multicast addresses
- Routers and hosts use MLD to report interest in respective Solicited-Node Multicast Addresses
- MLD will be studied later in detail.



IPv6 Start-up - Router



| | | |
|---------------------------------------|------------------------------------|-------------------------------|
| Multicast (all MLDv2-capable routers) | MLDv2 Report Message | Null address |
| ff02::16 | (Multicast all routers) | :: |
| Multicast (all MLDv2-capable routers) | MLDv2 Report Message | Null address |
| ff02::16 | (Multicast solicited-node address) | :: |
| Multicast solicited-node address | Neighbor Solicitation | Null address |
| ff02::1:ff+(address's last 24 bits) | (DAD link-local address) | :: |
| Multicast (all hosts) | Neighbor Advertisement | Link-local address |
| ff02::1 | | fe80::+(interface ID 64-bits) |
| Multicast (all MLDv2-capable routers) | MLDv2 Report Message | Link-local address |
| ff02::16 | (Multicast all routers) | fe80::+(interface ID 64-bits) |
| Multicast (all MLDv2-capable routers) | MLDv2 Report Message | Link-local address |
| ff02::16 | (Multicast solicited-node address) | fe80::+(interface ID 64-bits) |
| Multicast solicited-node address | Neighbor Solicitation | Null address |
| ff02::1:ff+(address's last 24 bits) | (DAD global address) | :: |
| Multicast (all hosts) | Router Advertisement | Link-local address |
| ff02::1 | | fe80::+(interface ID 64-bits) |

Only if global address is configured



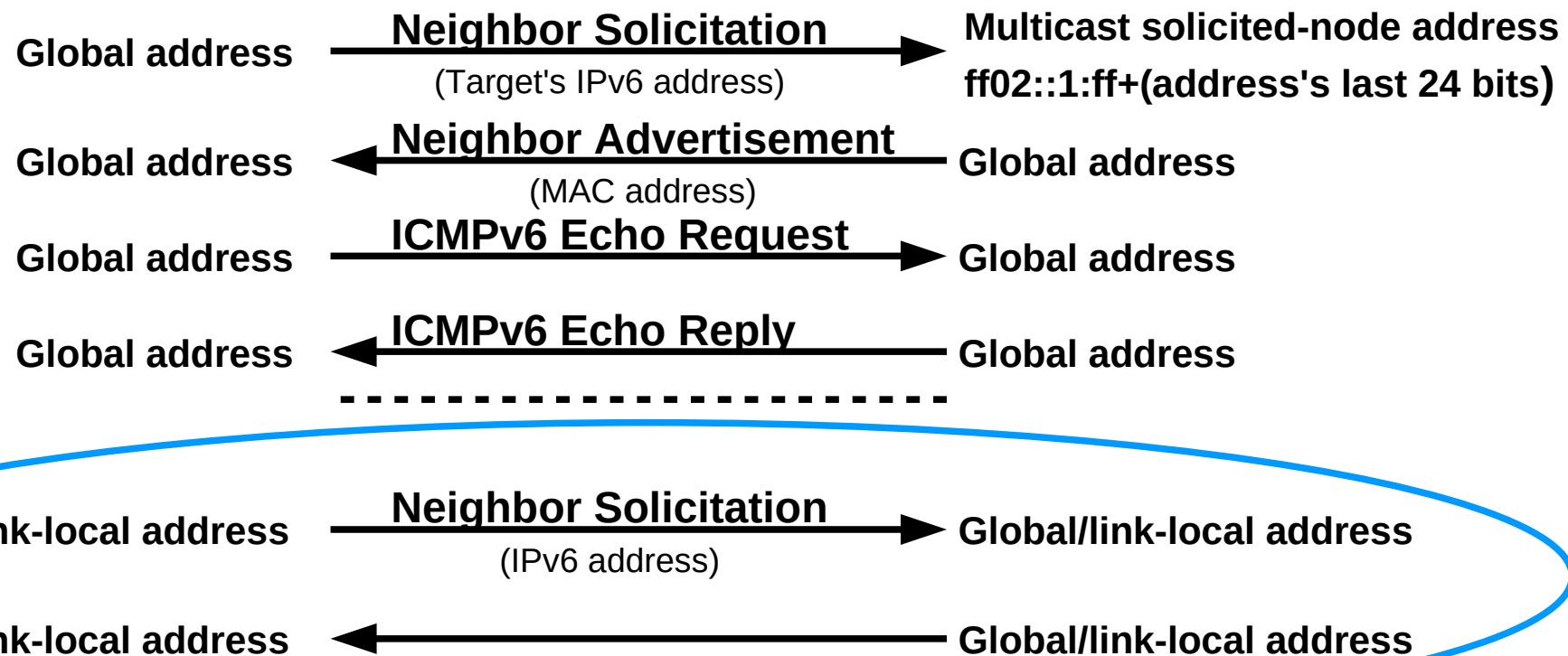
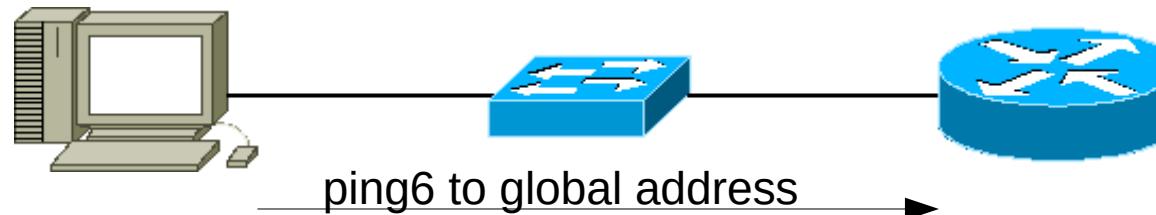
IPv6 Start-up – Terminal/Router Interaction



| | | |
|---|---|--|
| Null address :: | Neighbor Solicitation (DAD link-local address) | → Multicast solicited-node address $ff02::1:ff+(address's\ last\ 24\ bits)$ |
| Link-local address $fe80::+(interface\ ID\ 64-bits)$ | Router Solicitation | → Multicast (all routers) $ff02::2$ |
| Null address :: | MLDv2 Report Message (Multicast solicited-node address) | → Multicast (all MLDv2-capable routers) $ff02::16$ |
| Multicast (all hosts) $ff02::1$ | Router Advertisement | ← Link-local address $fe80::+(interface\ ID\ 64-bits)$ |
| Null address :: | Neighbor Solicitation (DAD global address) | → Multicast solicited-node address $ff02::1:ff+(address's\ last\ 24\ bits)$ |



Address Resolution and Ping6



To verify the reachability of a neighbor after physical address of a neighbor is identified



IPv6 Subnetting/Aggregation

- In IPv6 the same principles of IPv4 subnetting and aggregation are still valid.
 - ◆ Using the TLA, NLA and SLA bits of the IPv6 addresses.
 - ◆ Example: network 2001:A:A:/48 can be divided in 2^{16} sub-networks with identifiers 2001:A:A:****:/64
- By standard, the maximum mask size is /64, however it is possible to subnet also the host part of the IPv6 address.
 - ◆ Usage of mask /120 to protect the network from NDP Table Exhaustion attacks.
 - ◆ With mask /120 the maximum size of the NDP table is limited to 2^8 .
 - ◆ “Larger” masks also work.
 - ◆ Some tools/services may break.
 - ◆ Point-to-point links may use /126.
 - ◆ Some devices accept use /127, however in others may not work.
 - ◆ Requires manual, DHCPv6 address configuration or modified auto-configuration mechanisms.



IPv6 Addresses Planning

- Due to IPv6 nature, there are many networks and networks are large.
 - ◆ Number of hosts in LAN is not an issue!
 - ◆ Usually network managers receive /48 networks:
 - ◆ Allows for 2^{16} /64 networks.
 - ◆ Standard LAN use /64
 - or /120 to protect against attacks, however breaks stateless assignment.
 - ◆ Point-to-point links use /126.
 - Usually a /64 network is sub-netted into multiple /126.



Layer 2

Ethernet and Wi-Fi (802.11)

Fundamentos de Redes

**Mestrado Integrado em
Engenharia de Computadores e Telemática
DETI-UA**

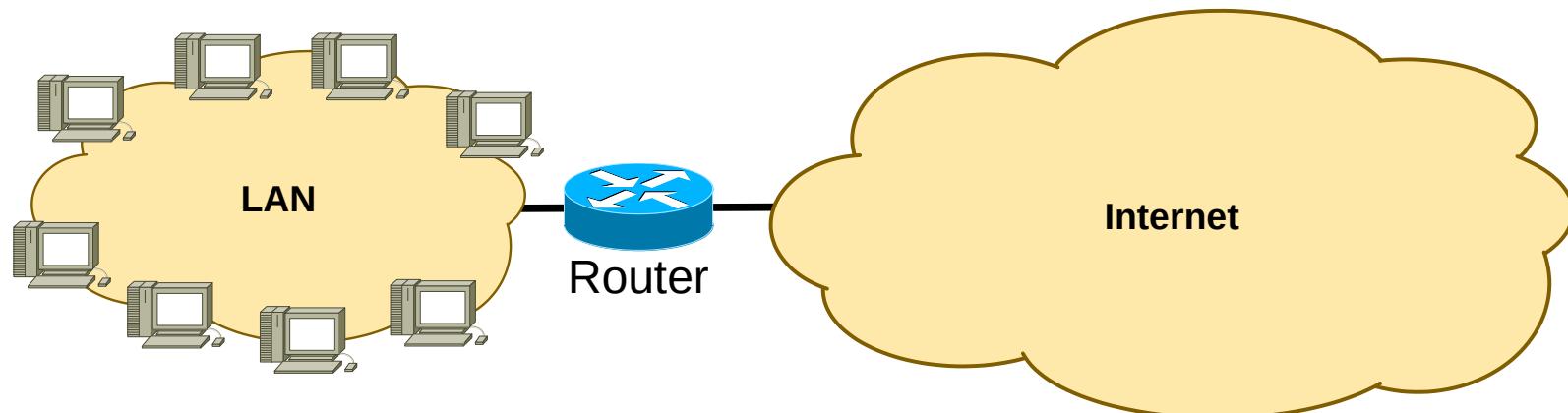


universidade de aveiro

deti.ua.pt

Local Area Network (LAN)

- Is a computer network within a small geographical area.
 - ◆ Home, school, room, office building or group of buildings.
- Is composed of inter-connected hosts capable of accessing and sharing data, network resources and Internet access.
 - ◆ Host refers generically to a PC, server, or any other terminal.
- Technologies
 - ◆ Current: Ethernet, 802.11 (Wi-Fi)
 - Legacy: Token Ring, FDDI, ...



Ethernet

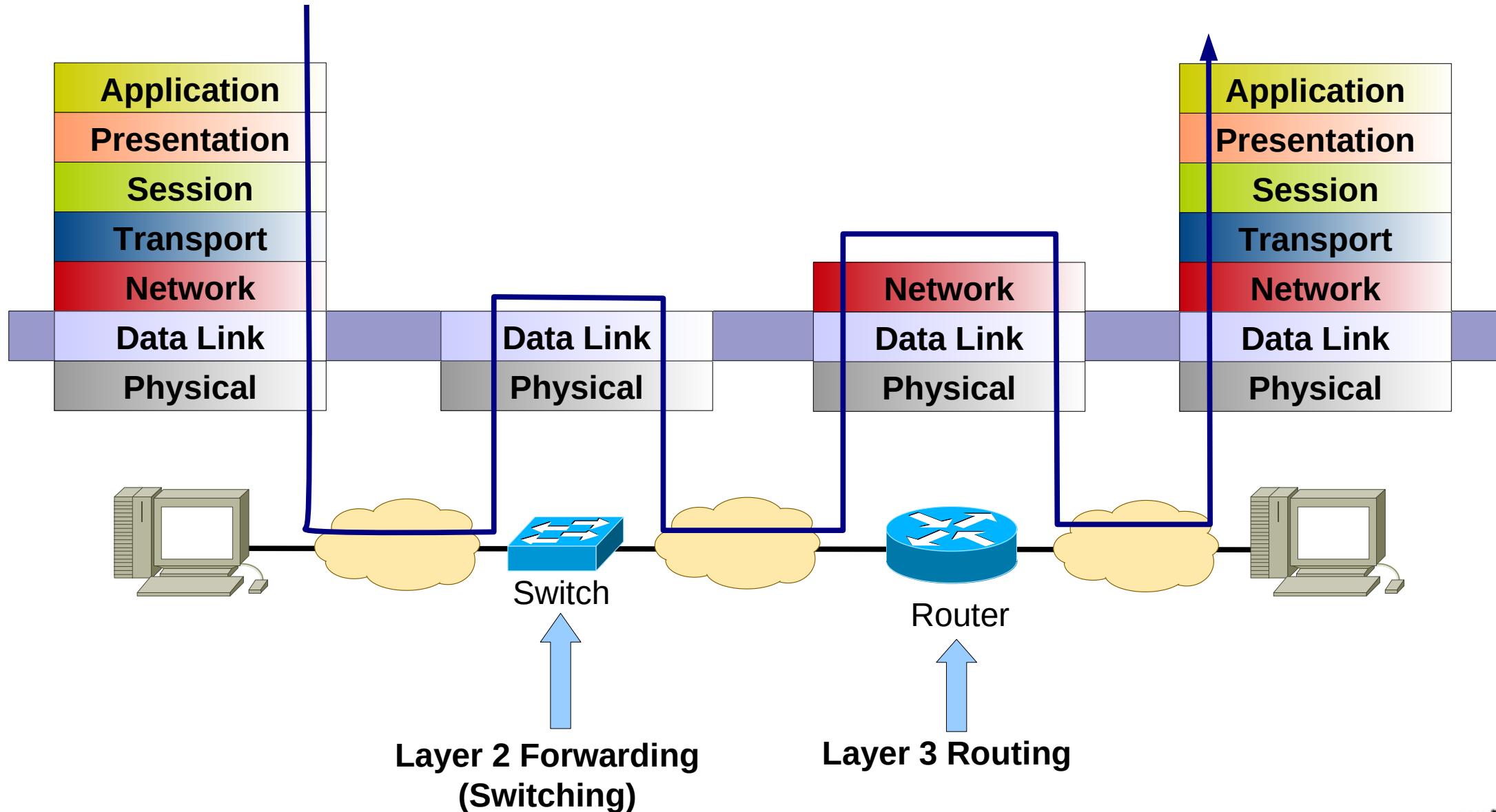


Ethernet (802.3)

- Most successful LAN technology.
- Invented at Xerox Palo Alto Research Center (PARC).
- Xerox, DEC and Intel defined in 1978 the standard for Ethernet 10Mbps.
- Uses “Carrier Sense/Multiple Access” with “Collision Detect” (CSMA/CD)
 - ◆ Carrier Sense: hosts can perceive if the communication channel is being used.
 - ◆ Multiple Access: multiple host can access simultaneously
 - ◆ Collision Detect: host “listen” the communication channel while transmitting to detect transmission collisions.
 - ◆ Collision: multiple physical signals overlapping and interfering with each other.



Ethernet based LAN

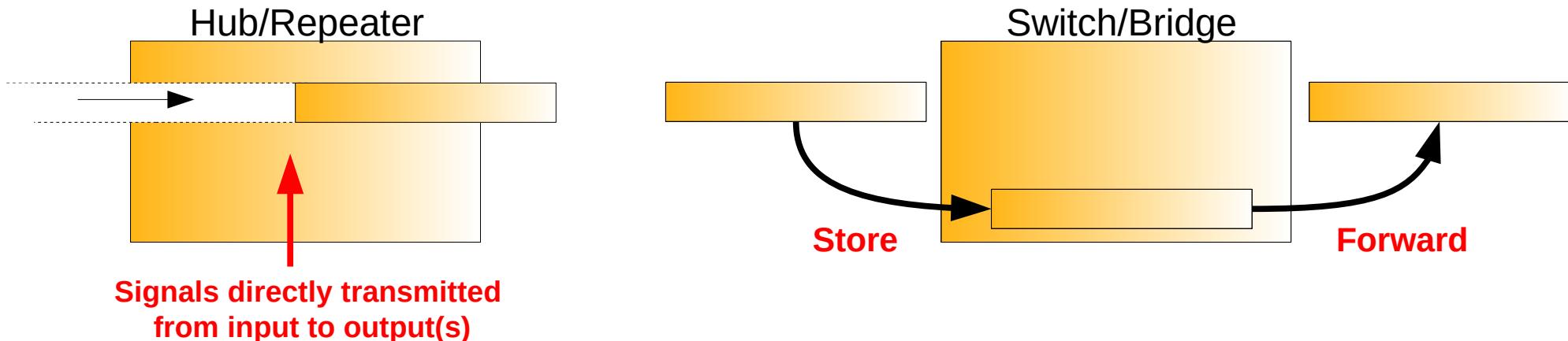


Ethernet Equipment

- Hub/Repeater:
 - ◆ Operates only at the physical level (OSI Layer 1).
 - ◆ Replicates and regenerates electrical signals.
 - ◆ Hub = repetidor com múltiplas portas.
 - ◆ **Não é usado nas redes locais actuais!**
- Switch/Bridge:
 - ◆ Store-and-forward operation.
 - ◆ Operates only at the data link level (OSI Layer 2).
 - ◆ Physically separates (and logically interconnects) different collision domains
 - ◆ Nowadays all Ethernet hosts are connected to a switch → There no Ethernet collision domains!
 - ◆ Forwards frames based on MAC addresses.
 - ◆ Switch = bridge with multiple ports.
- Router:
 - ◆ Store-and-forward operation.
 - ◆ Operates only at the network level (OSI Layer 3).
 - ◆ Routes packets based on network addresses (e.g., IPv4 and IPv6).



Switches/Bridges vs. Hubs/Repeaters

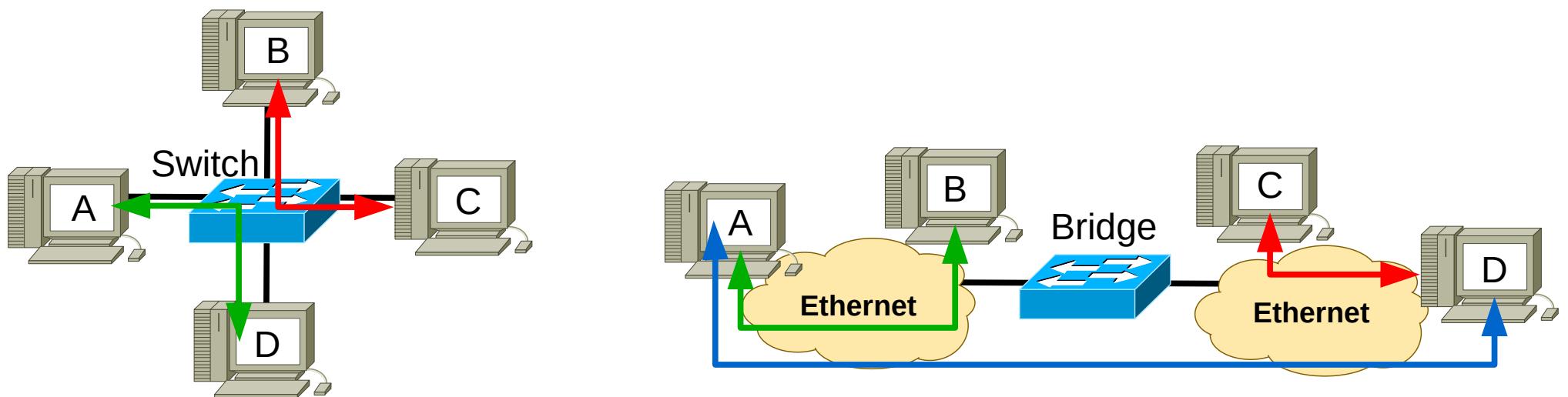


- Bridges/switches interconnect different local networks.
- Bridges/switches additional functions:
 - ◆ Store & Forward + Filtering
 - ✚ The Forwarding process decides to send a frame to a specific port based on the destination MAC address of the frame.
 - ✚ Ports may operate at different speeds.

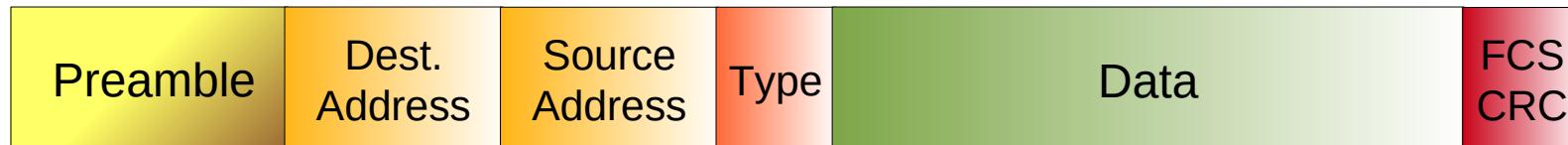


Switching

- With Switches/Bridges
 - Interconnection done at OSI Layer 2.
 - Hosts can transmit simultaneously.
 - A network of Switches is a **Broadcast Domain**
 - An Ethernet frame with destination FF:FF:FF:FF:FF:FF (Broadcast) will reach all connected switches and hosts.



Ethernet Frame

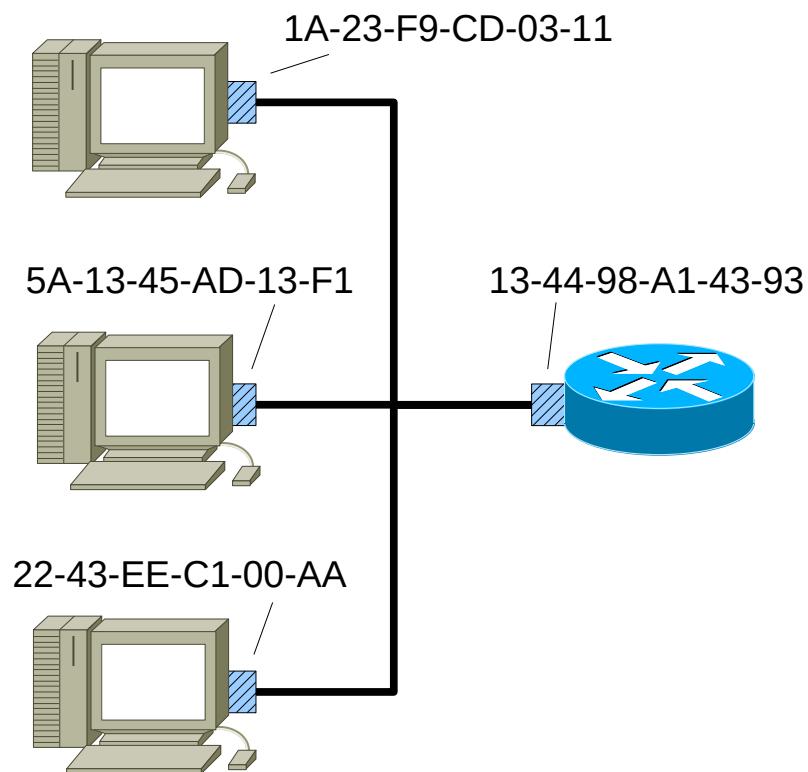


- The sender's network card encapsulates an IP datagrama (or any other network protocol) in an Ethernet frame.
- Preamble:
 - ◆ 7 bytes with pattern 10101010 followed by one byte with pattern10101011.
 - ◆ Used to sincronize the sending and receiving clocks.
- Destination and Source addresses: 6 bytes Physical (MAC) address
 - ◆ If the network card receives a frame with destination equal to its own address or its the broadcast address, it will pass data to the network level process.
 - ◆ If not, drops the frame.
- Type defines which protocol is encapsulated in the frame (usually IPv4 or IPV6).
- The frame check sequence (FCS) is a four-octet cyclic redundancy check (CRC) that allows detection of corrupted data within the entire frame as received on the receiver side.



MAC Addresses

- MAC (Physical, Ethernet or LAN) Address:
 - ◆ Function: Allow the exchange of data between network interfaces connected using a Layer 2 network.
 - ◆ Have 6 bytes/48 bits.
 - ◆ Are unique.
 - ◆ Each network card has its own address.
 - ◆ Defined by manufacturer
 - ◆ Some hardware allows change.
 - ◆ First 24-, 28-, or 36-bits assign to manufacturer.
 - ◆ Hexadecimal notation
 - ◆ Broadcast: FF-FF-FF-FF-FF-FF



Ethernet Frame Minimum Size

- Historically there were Ethernet technologies that allowed collisions and a collision detection mechanism had to be present (CSMA/CD).
- Depending on the technology and maximum cable size, the Ethernet frame had to be big enough to allow the collision detection mechanism to detect a frame being transmitted before the last frame byte leaving the source host.
- By legacy (it is possible to merge different Ethernet technologies) the **minimum frame size is 64 bytes**.
- If the frame's header plus data do not reach 64 bytes, a set of zeros must be added to the end of the frame to reach 64 bytes.
 - ◆ This is called **padding**.



Switches Basic Operations

- Switches have a **Forwarding Table**.
- When a switch receives an Ethernet frame:
 - ◆ Registers an entry at the Forwarding Table the frame's source MAC address and the port where the frame was received.
 - If no frames are received from that MAC address after some time (**aging time**) the entry is removed.
 - ◆ Searches the Forwarding Table for the frame's destination MAC address and forwards the packet according:
 - **Forwarding** mechanism:
 - If the frame's destination MAC address exists in the table, the switches forwards the frame through the port associated with that MAC address.
 - **Flooding** mechanism:
 - If the frame's destination MAC address DOES NOT exist in the table, the switches forwards the frame through all active ports (except the one where it was received).
 - » Note: Just within the same VLAN (more details later).

| MAC | Porta |
|-------------------|-------|
| 00:11:11:11:11:11 | 1 |
| 00:22:22:22:22:22 | 1 |
| A1:33:33:33:33:33 | 2 |
| 44:44:44:44:44:44 | 3 |
| 55:55:55:00:00:55 | 3 |



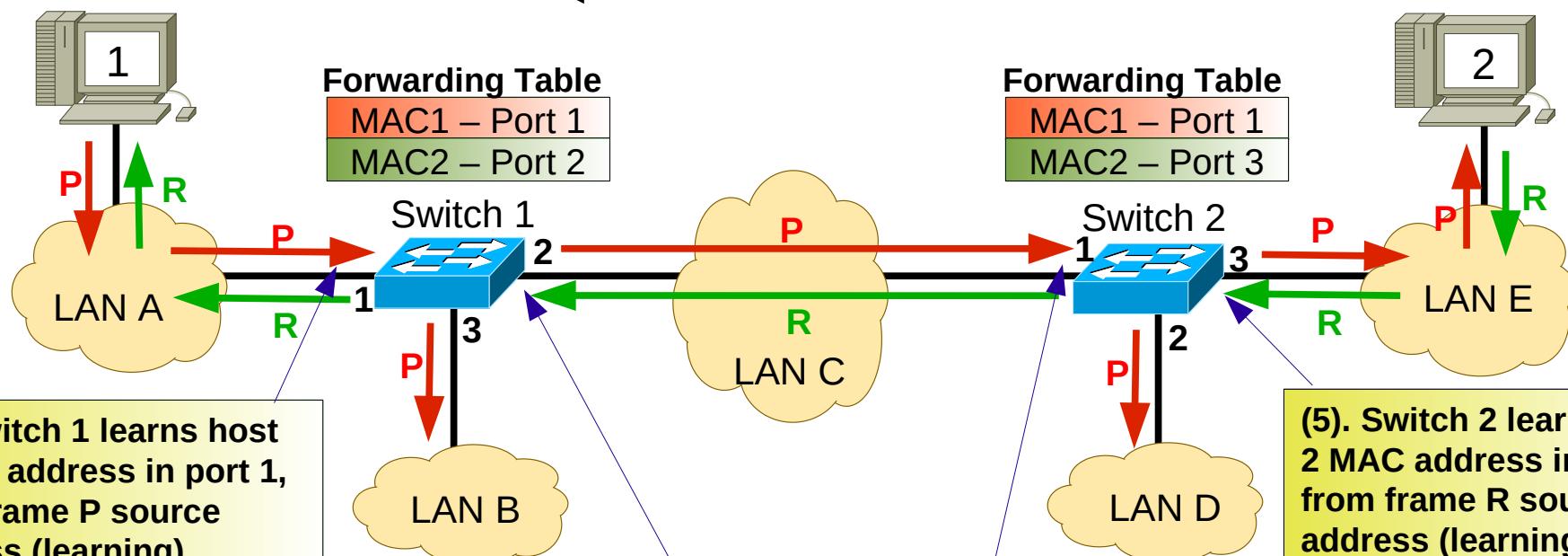
Learning, Flooding and Forwarding

Frame P

Dest. = MAC2 Source = MAC1

Frame R (Answer to P)

Dest. = MAC1 Source = MAC2



(1). Switch 1 learns host 1 MAC address in port 1, from frame P source address (learning).
 (2). Switch 1 does not have frame's P destination (MAC 2) in the table, sends frame P to all ports except port 1 (flooding).

(7). Switch 1 learns host 2 MAC address in port 2, from frame R source address (learning).
 (8). Switch 2 have frame's R destination (MAC 1) in the table, sends frame R to port 1 (forwarding).

(3). Switch 2 learns host 1 MAC address in port 1, from frame P source address (learning).
 (4). Switch 2 does not have frame's P destination (MAC 2) in the table, sends frame P to all ports except port 1 (flooding).

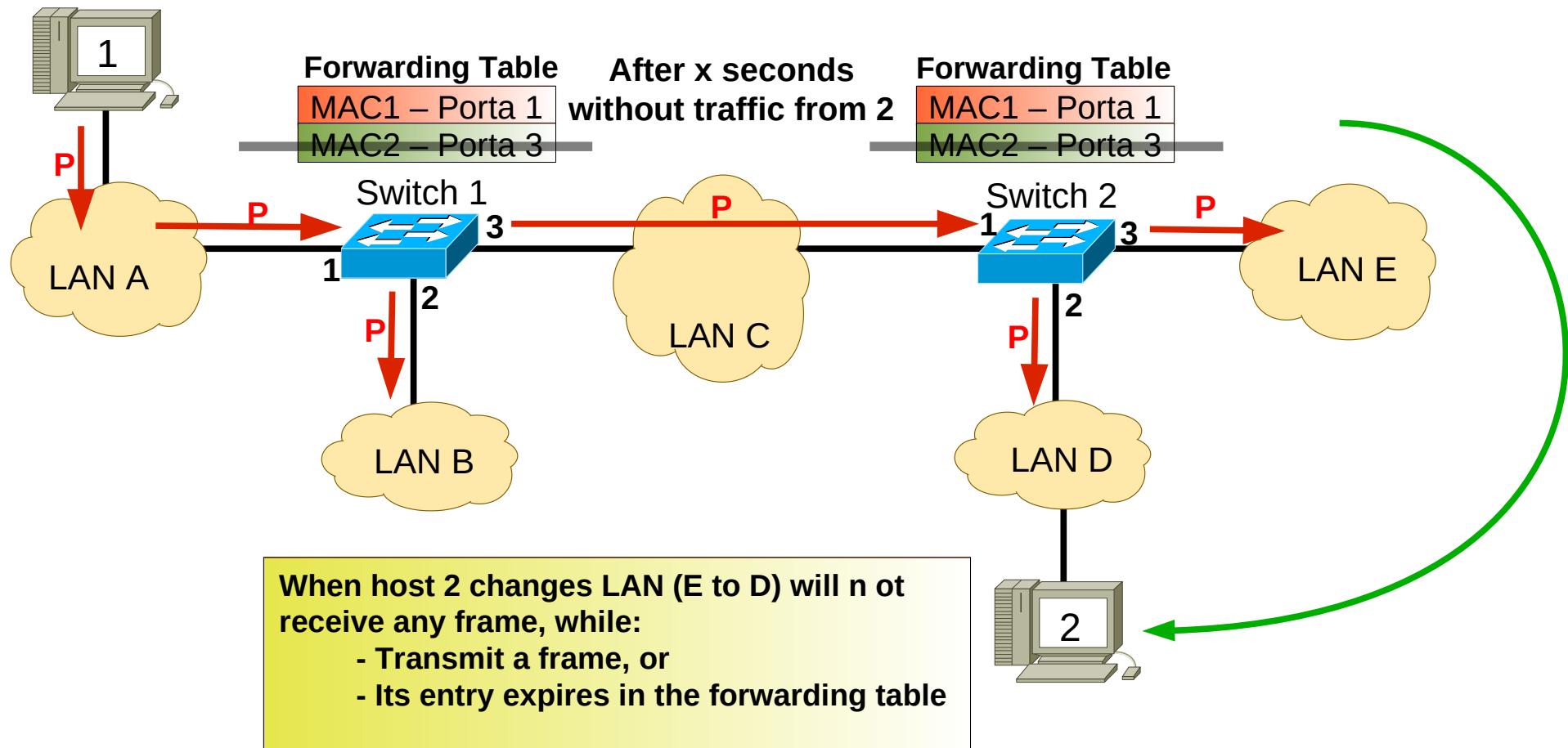
(5). Switch 2 learns host 2 MAC address in port 3, from frame R source address (learning).
 (6). Switch 2 have frame's R destination (MAC 1) in the table, sends frame R to port 1 (forwarding).



Forwarding Table Aging Time

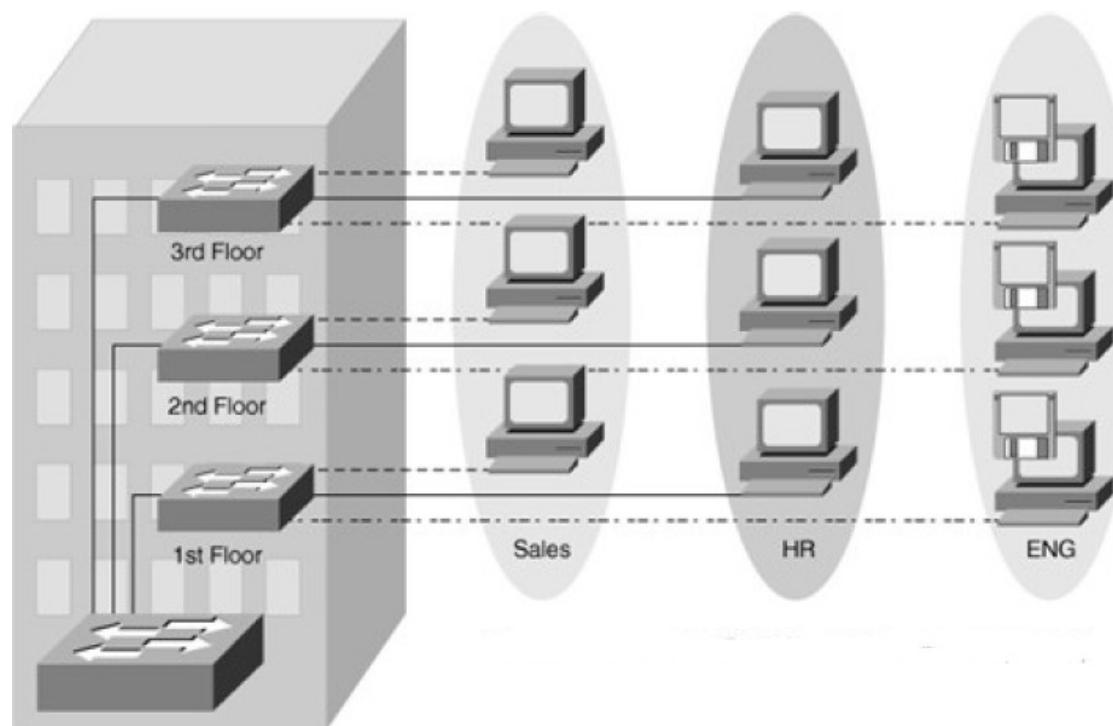
Frame P

Dest. = MAC2 | Source = MAC1



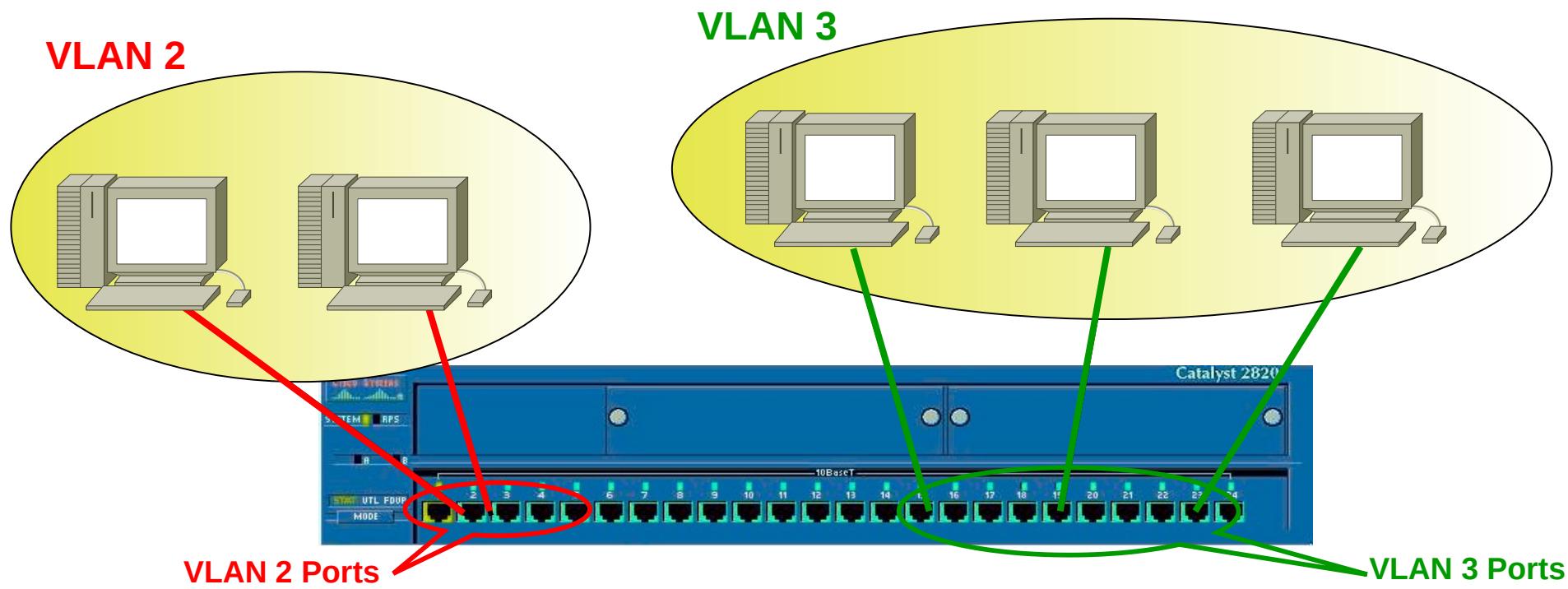
Virtual LAN (VLAN)

- A Virtual LAN (VLAN) is a group of hosts/users with a common set of requirements or characteristics in the same broadcast domain.
 - ◆ Independent of their physical location.
- Solves the scalability problems of large networks.
 - ◆ By breaking a single broadcast domain into several smaller broadcast domains.
 - ◆ Allows better/simpler network administration and security deployment.
- Hosts in different VLAN do not communicate by Layer 2.
 - ◆ Its communications are done at Layer 3 (with IP routing).



Defining Host VLAN

- The VLAN to which a host belongs depends only on the port of the switch.
 - ◆ Configured only in the switch.
 - ◆ Example: If port 1 is configured as VLAN 2, and port 20 is configured as VLAN 3:
 - ◆ If host is connected to port 1 it is on VLAN 2,
 - ◆ If host is connected to port 20 it is on VLAN 3.
- VLAN 1 is usually reserved to network administration.
 - ◆ Used to access configurations remotely via IP.



Example – VLAN

Pings sent by 10.0.0.1



```
# ping 10.0.0.2
```

```
Pinging 10.0.0.2 with 32 bytes of data:
```

```
Reply from 10.0.0.2: bytes=32 time<10ms TTL=128
```

```
Ping statistics for 10.0.0.2:
```

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

```
# ping 10.0.0.5
```

```
Pinging 10.0.0.5 with 32 bytes of data:
```

```
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

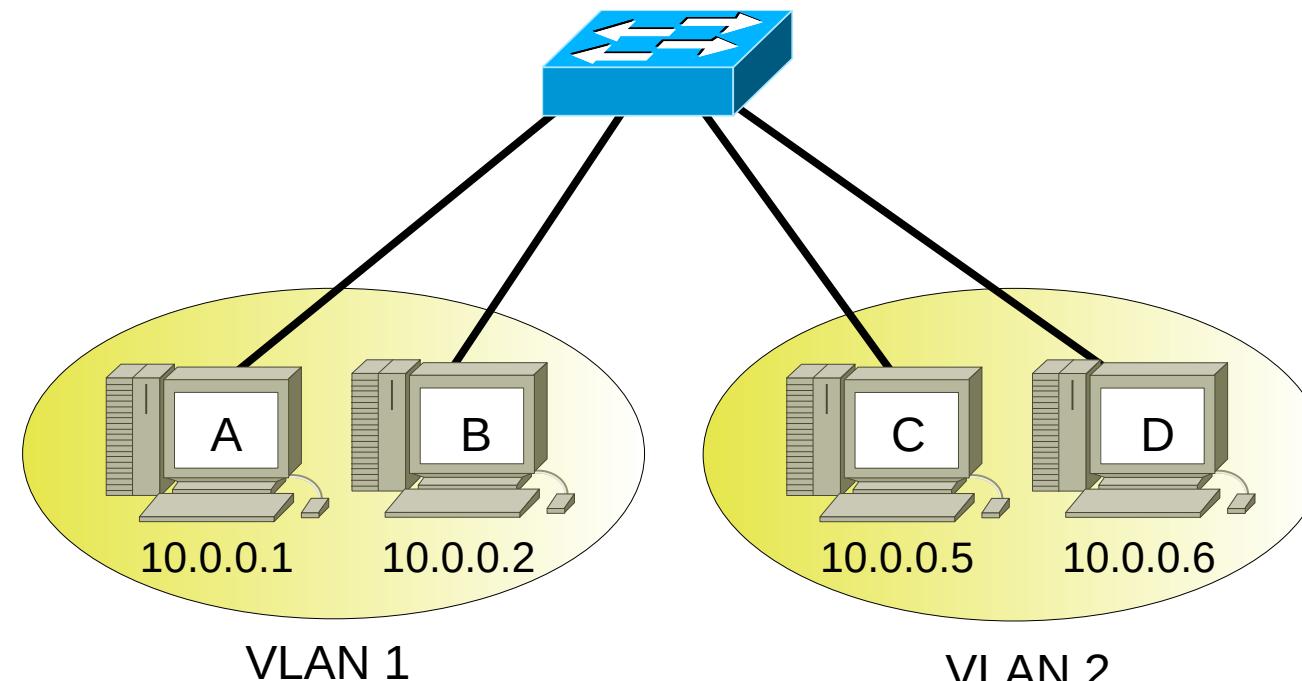
```
Ping statistics for 10.0.0.5:
```

```
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
# ping 10.0.0.6
```

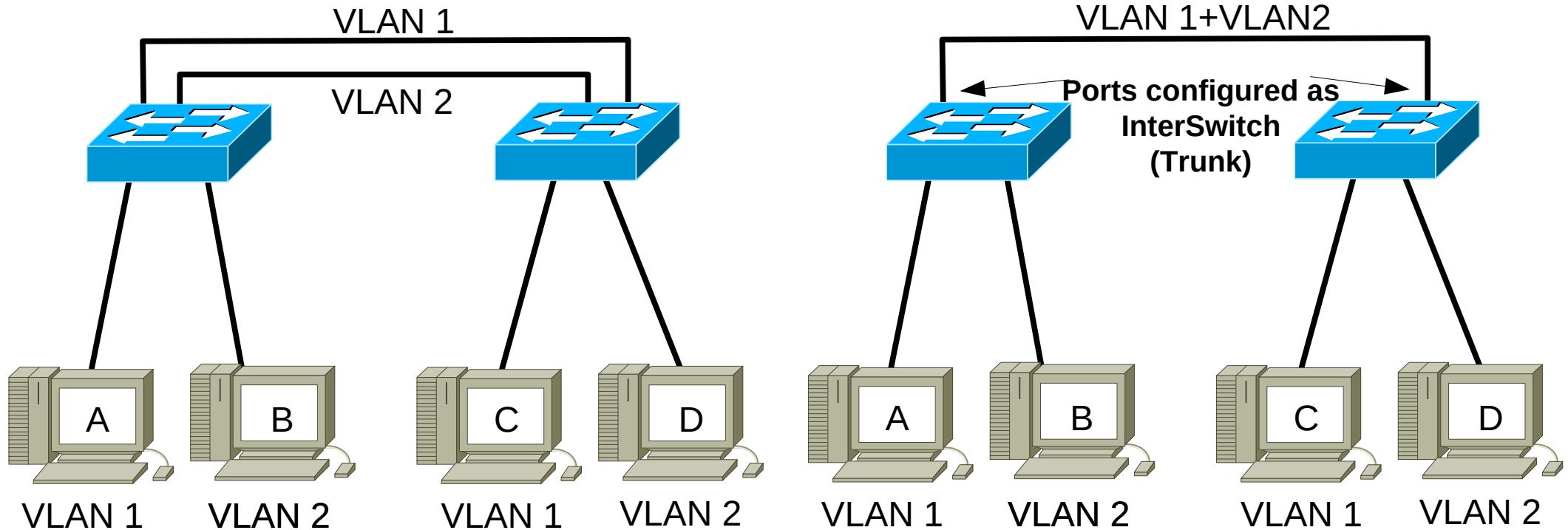
```
Pinging 10.0.0.6 with 32 bytes of data:
```

```
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```



Interconnection of Switches

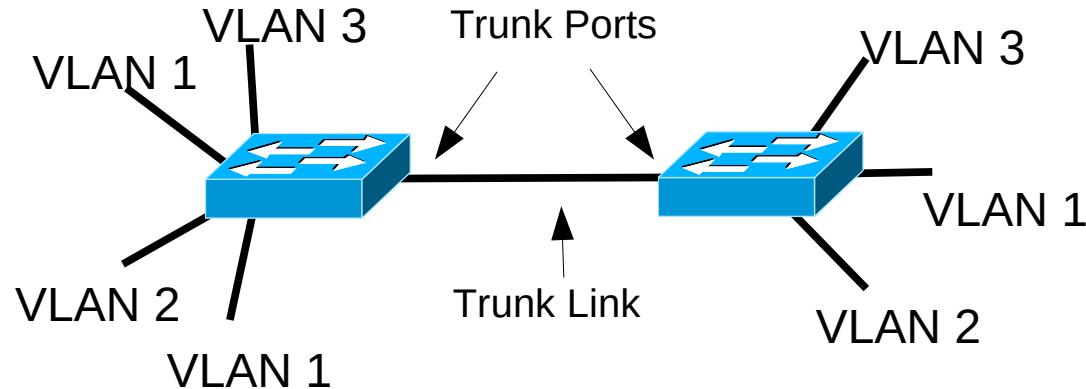
- Physical link per VLAN
 - With a single physical link.
 - Using InterSwitch/Trunk port(s).



- Using a single physical link requires a mechanism to differentiate frames from different VLAN.
 - ◆ Frames must have a tagged
 - ✚ Added when forwarding to a trunk port.
 - ✚ Read and removed when receiving a frame from a trunk port



IEEE802.1Q Standard



Ethernet frame without a VLAN tag

| 6 | 6 | 2 | | |
|-------------|--------|------|------|--|
| destination | source | type | data | |

Ethernet frame with a VLAN tag

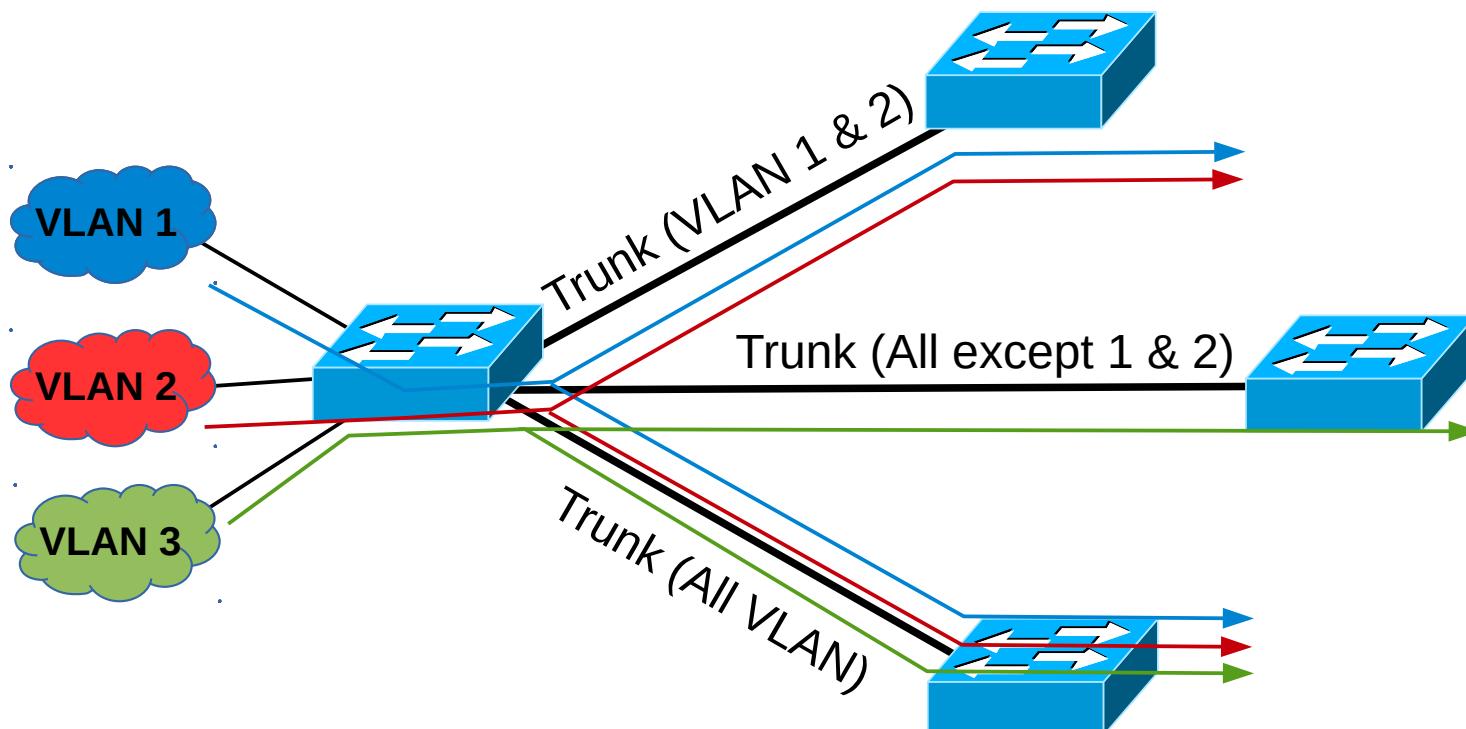
| 6 | 6 | 4 | 2 | | |
|-------------|--------|--------|----------|------|---------|
| destination | source | TAG | type | data | |
| | | 16bits | 3bits | 1bit | 12bits |
| | | 8100h | priority | CFI | VLAN ID |

- Priority: Traffic relative priority according to standard 802.1q (0 to 7 values).
- CFI: Used to guarantee compatibility with older technologies (always zero in Ethernet).
- VLAN ID: VLAN identifier.



Trunk Links

- The physical link between two Trunk ports is called a Trunk link.
- A trunk carries traffic for multiple VLANs using IEEE 802.1Q.
 - ◆ Inter-Switch Link (ISL) encapsulation is an alternative but it getting obsolete.
- Trunks may transport all VLAN or only some!

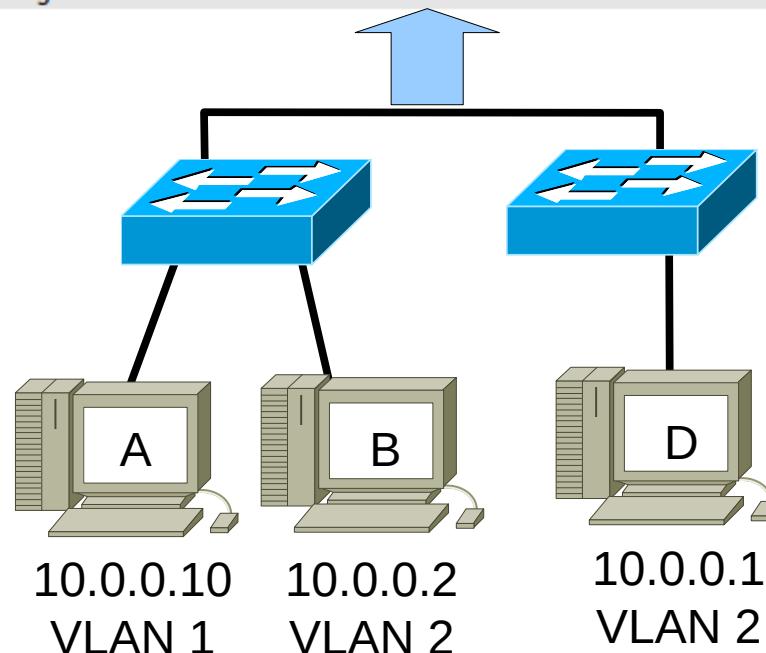


Example – InterSwitch/Trunk Ports

| Filter: icmp | | | | Expression... | Clear | Apply |
|--------------|-----------|----------|-------------|---------------|---------------------|-------|
| No.. | Time | Source | Destination | Protocol | Info | |
| 23 | 11.535990 | 10.0.0.2 | 10.0.0.1 | ICMP | Echo (ping) request | |
| 24 | 11.536995 | 10.0.0.1 | 10.0.0.2 | ICMP | Echo (ping) reply | |
| 27 | 12.538443 | 10.0.0.2 | 10.0.0.1 | ICMP | Echo (ping) request | |
| 28 | 12.539186 | 10.0.0.1 | 10.0.0.2 | ICMP | Echo (ping) reply | |

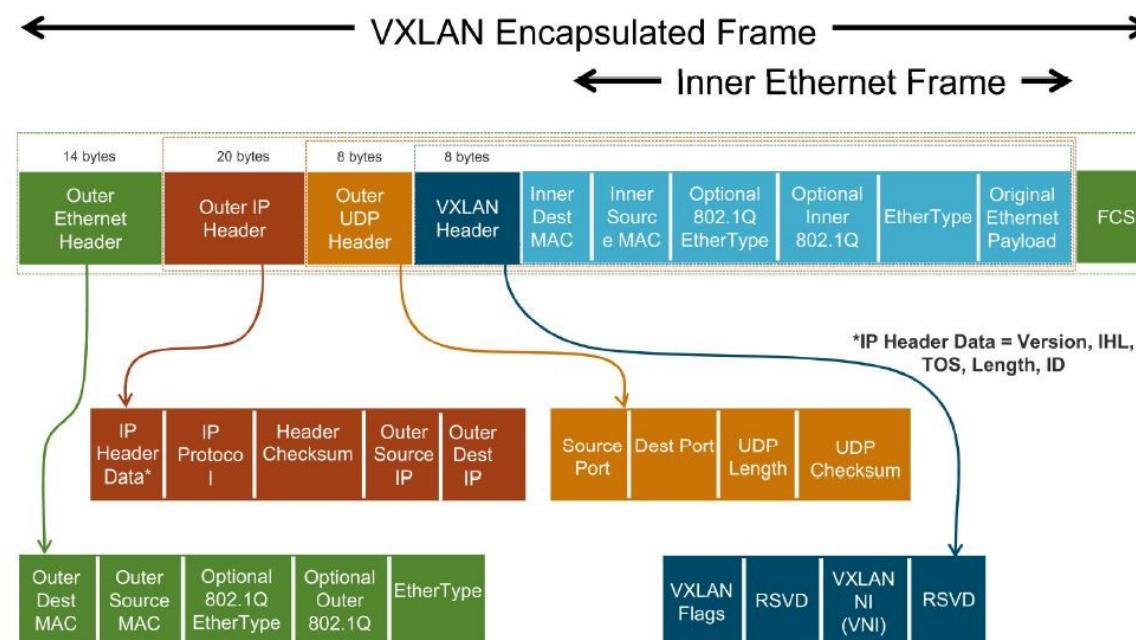
Frame 23 (102 bytes on wire, 102 bytes captured)
Ethernet II, Src: 00:aa:00:53:7c:00 (00:aa:00:53:7c:00), Dst: 00:aa:00:fa:67:00 (00:aa:00:fa:67:00)
802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 2
000. = Priority: 0
...0 = CFI: 0
.... 0000 0000 0010 = ID: 2
Type: IP (0x0800)
Internet Protocol, Src: 10.0.0.2 (10.0.0.2), Dst: 10.0.0.1 (10.0.0.1)
Internet Control Message Protocol

ID:2 == VLAN 2



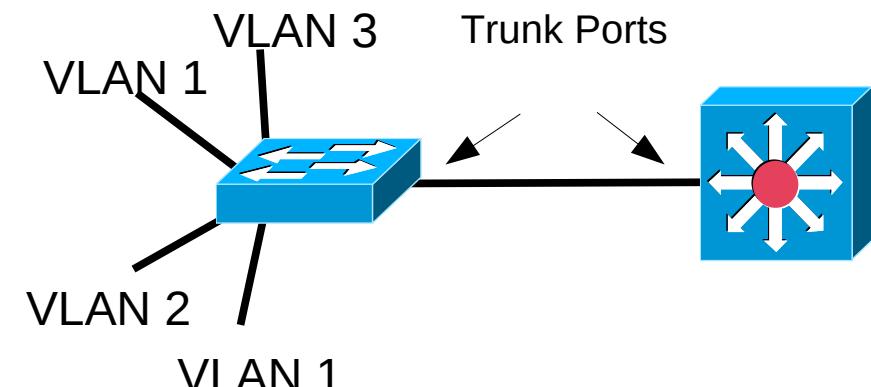
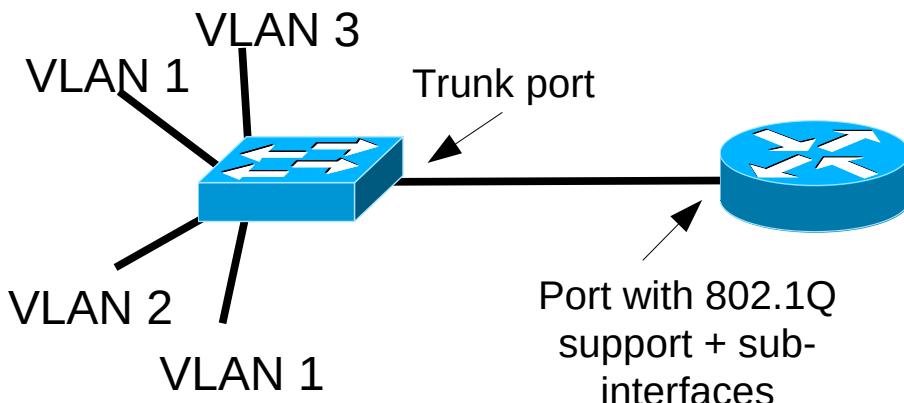
Virtual Extensible LAN (VXLAN)

- Alternative/Complement to 802.1Q in Layer3 Switches.
- Encapsulates OSI Layer 2 Ethernet frames within Layer 4 UDP/IP datagrams .
 - ◆ Default port 4789.
- VLAN may be additionally identified by a VNI field with 24 bits.
 - ◆ 802.1Q tag only as 12 bits.
 - ◆ Allows for a very large number of VLAN.
- Usually used when connecting remote VLAN (connected only via IP) in Datacenter and Cloud scenarios.

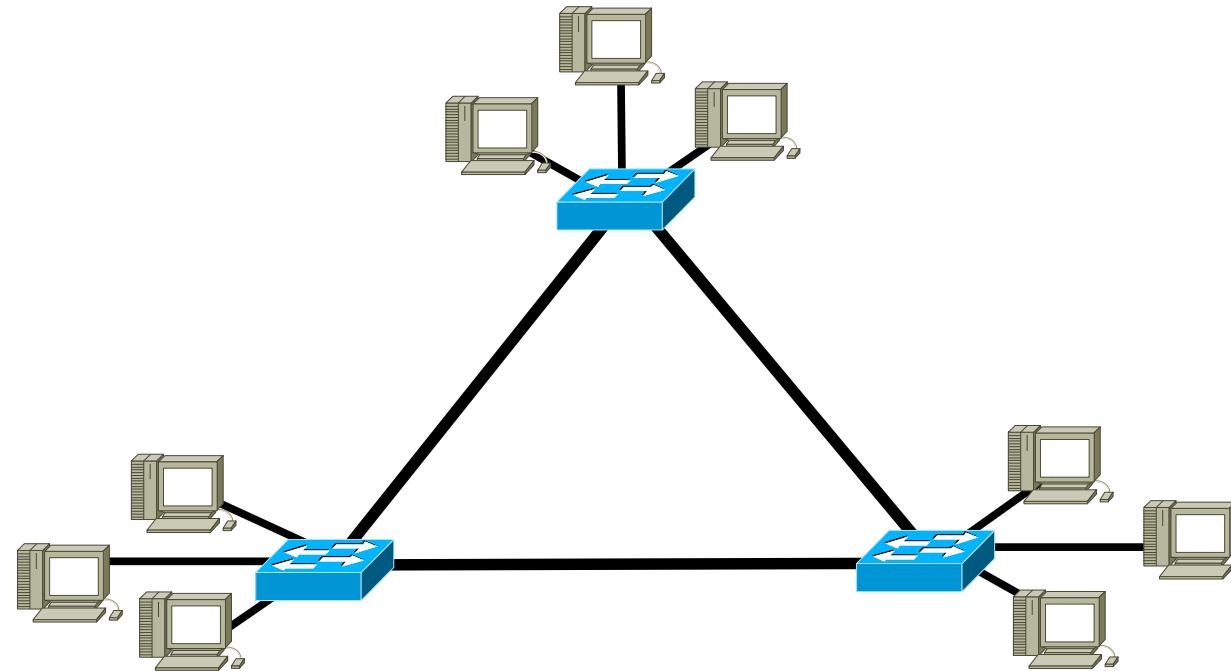


IP Connection between VLANs

- To communicate between different VLAN it is required to use Layer 3 (IP Routing).
- Common solutions:
 - ◆ A router with support to 802.1Q,
 - ◆ Connecting the physical router interface to a Trunk port.
 - ◆ The router's physical interface is sub-divided in sub-interfaces (one for each VLAN).
 - ◆ The IP gateway for a VLAN host is the IP address of the respective sub-interface in the Router.
 - ◆ A Layer 3 switch,
 - ◆ Connecting both switches (L3 and L2) using Trunk ports.
 - ◆ Each VLAN is mapped to a virtual Layer 3 interface.
 - ◆ The IP gateway for a VLAN host is the IP address of the respective virtual interface in the L3 switch.



Redundant Layer 2 Network



- Objective: Allow the network for dynamically recover from network failures.
- Problem: Link redundancy creates Layer 2 loops. Causes the collapse of communications when MAC frames with broadcast address are sent by any host due to infinite frame flooding.

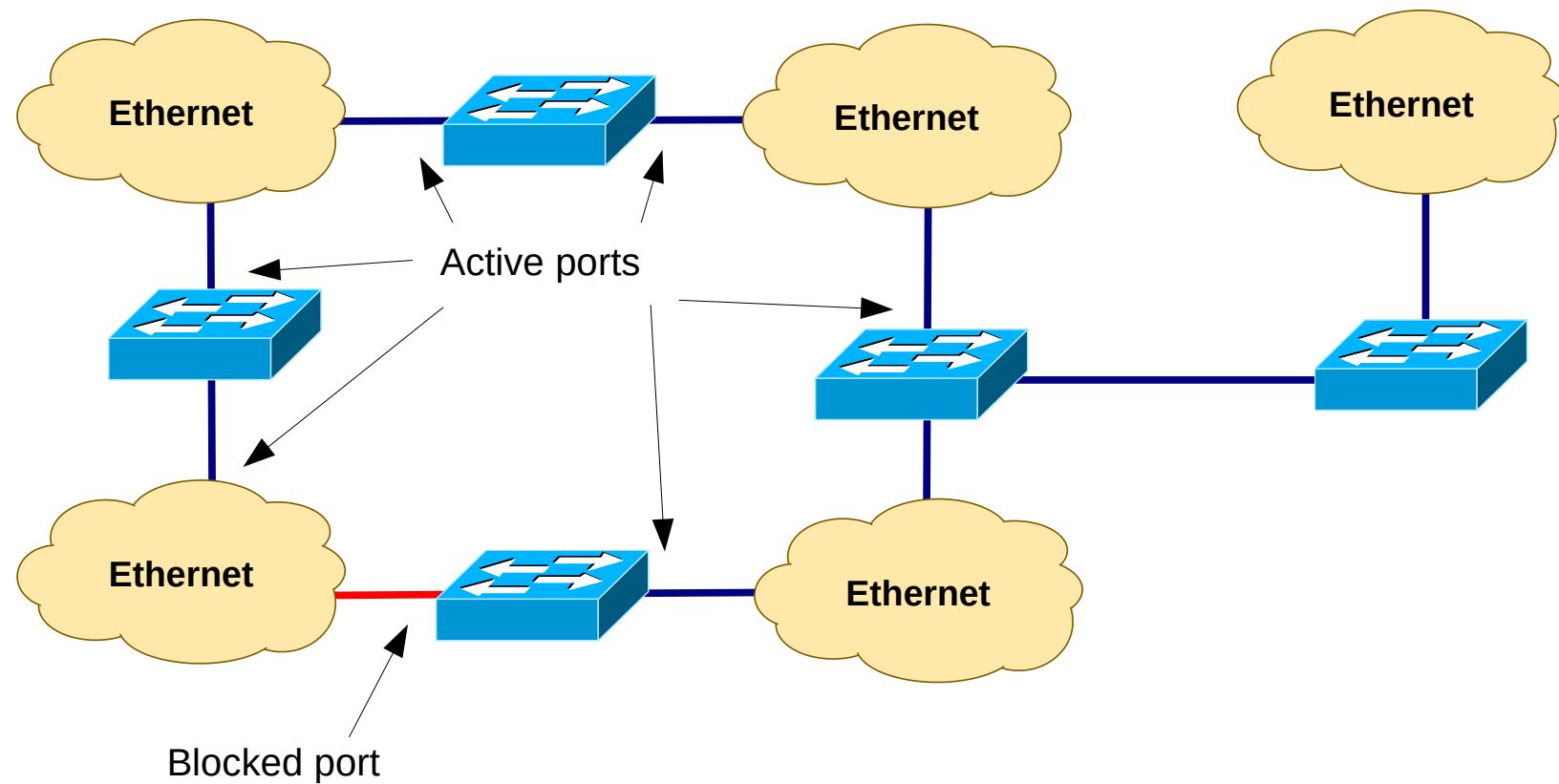


Spanning Tree Protocol (SPT)

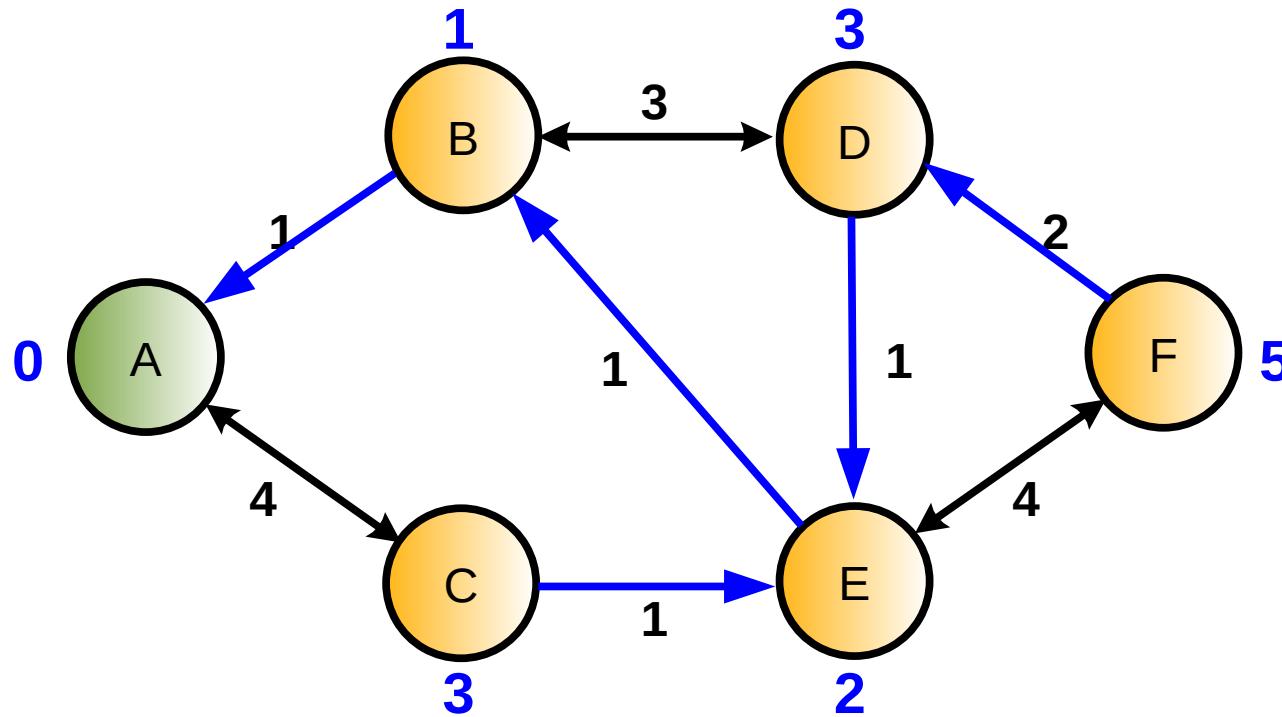
- STP enables the network to deterministically block ports and provide a loop-free topology in a network with redundant links.
- There are several STP Standards and Features:
 - STP is the original IEEE 802.1D version (802.1D-1998) that provides a loop-free topology in a network with redundant links.
 - RSTP, or IEEE 802.1W, is an evolution of STP that provides faster convergence of STP.
 - Multiple Spanning Tree (MST) is an IEEE standard. MST maps multiple VLANs into the same spanning-tree instance.
 - Per VLAN Spanning Tree Plus (PVST+) is a Cisco enhancement of STP that provides a separate 802.1D spanning-tree instance for each VLAN configured in the network.
 - RPVST+ is a Cisco enhancement of RSTP that uses PVST+. It provides a separate instance of 802.1W per VLAN.



Spanning-Tree



Bellman Equations



- When link cost are not negative, then:

Shortest path from one node X to node A

=

Cost of the link from that node X to the node that follows it in the shortest path to A

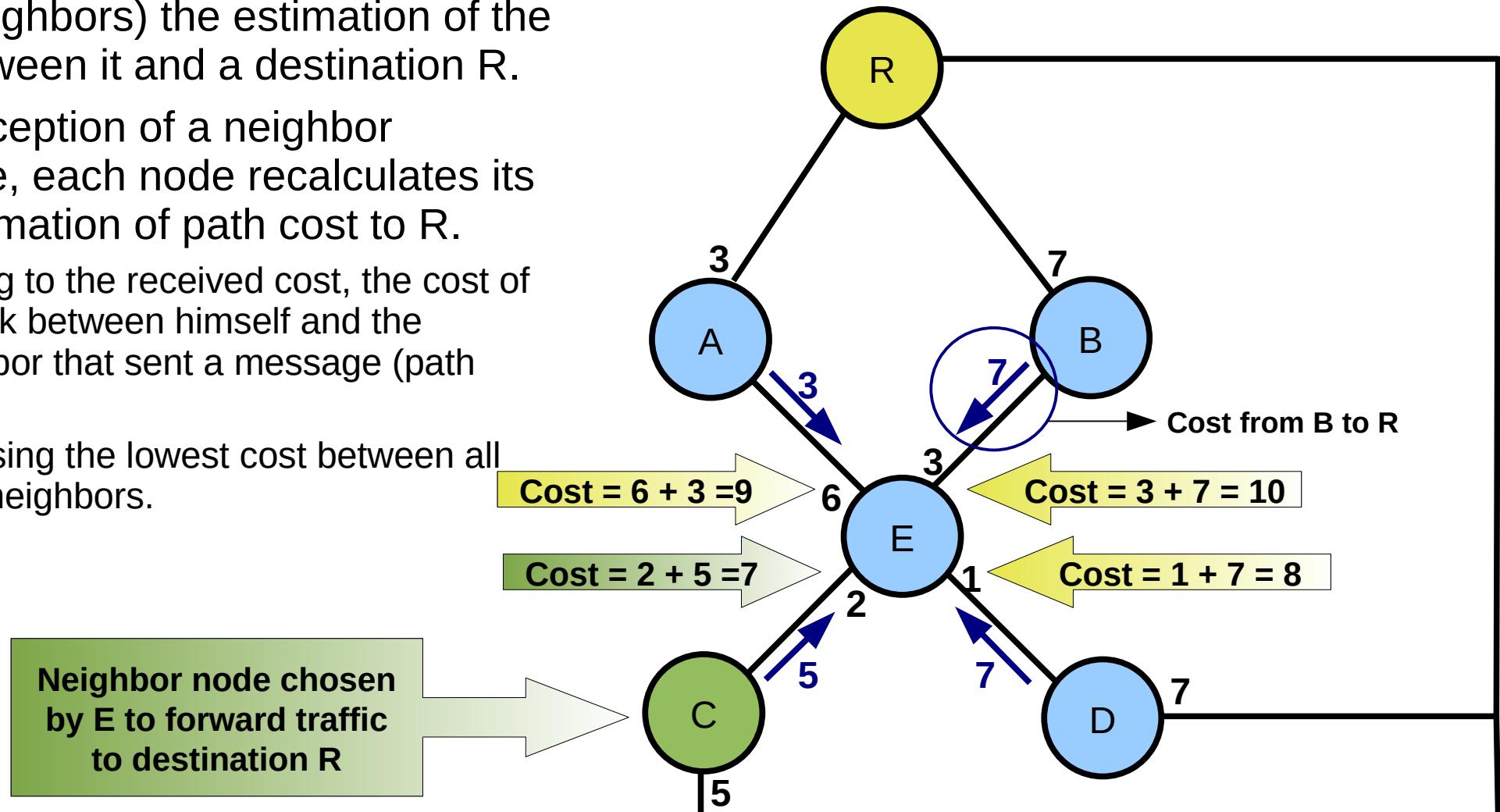
+

Shortest path from that node to node A



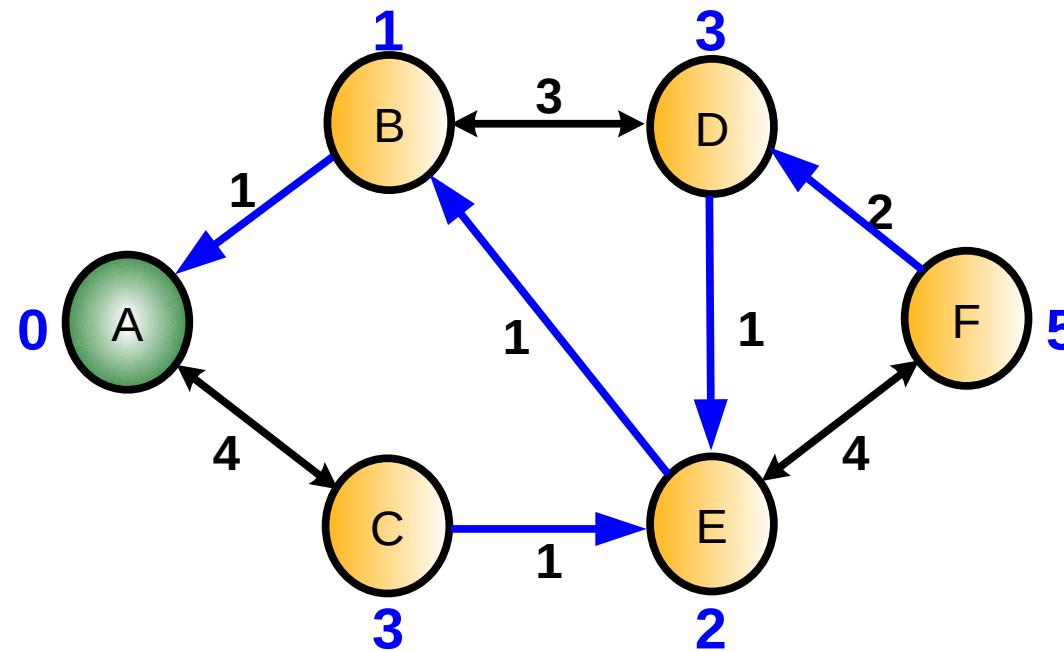
Bellman-Ford Distributed and Asynchronous Algorithm

- Each node transmits periodically (to all its neighbors) the estimation of the cost between it and a destination R.
- Upon reception of a neighbor message, each node recalculates its own estimation of path cost to R.
 - ◆ Adding to the received cost, the cost of the link between himself and the neighbor that sent a message (path cost).
 - ◆ Choosing the lowest cost between all links/neighbors.

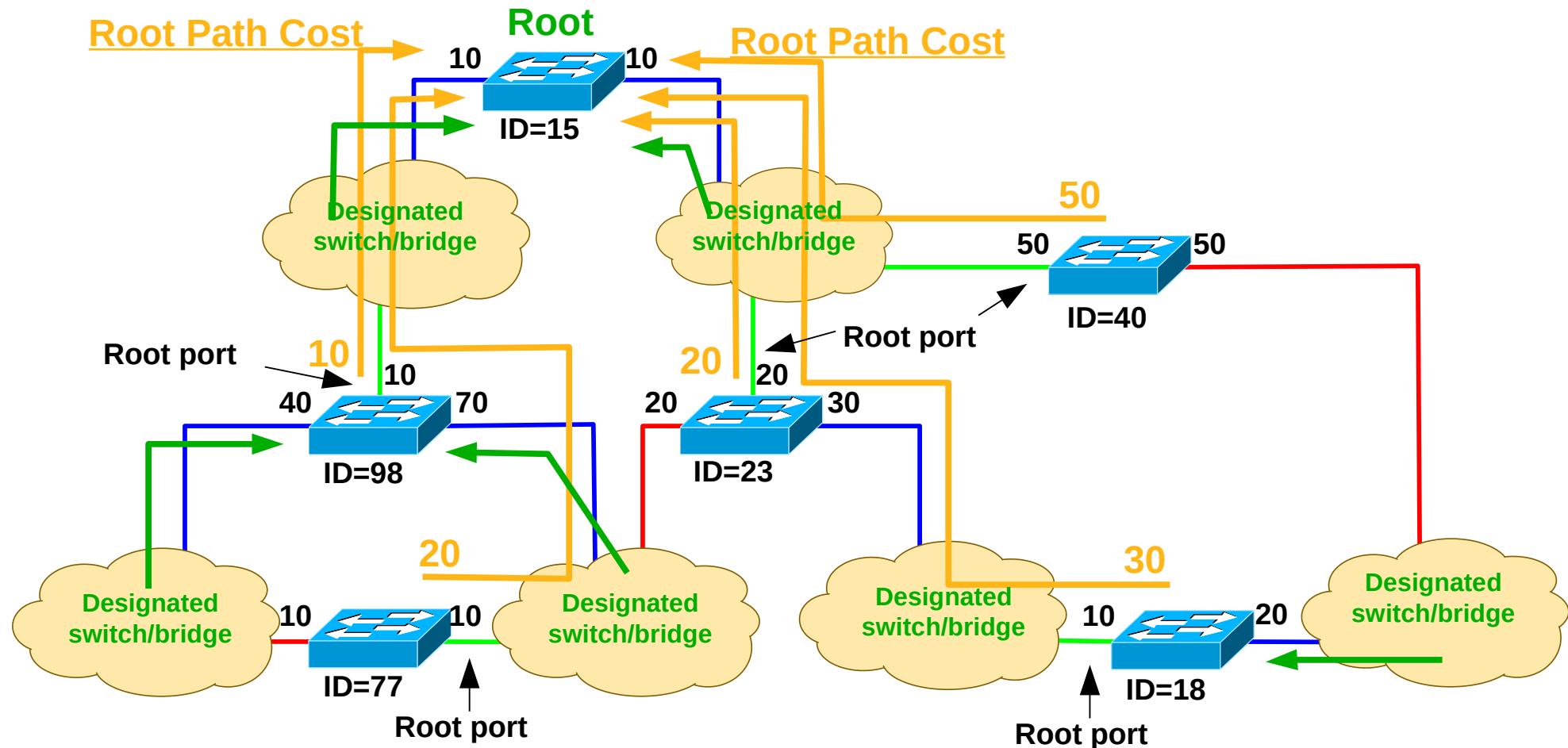


Routing based on Spanning Trees

- It is chosen an origin/root node.
- All nodes use the **Bellman-Ford Distributed and Asynchronous Algorithm** to calculate the neighbored node (and respective path cost) that provide the smallest cost to the origin/root node.
- The set of links used by all nodes to provide the shortest paths to the origin/root node is called the **Spanning Tree**.
- It is required a criteria to solve ties.

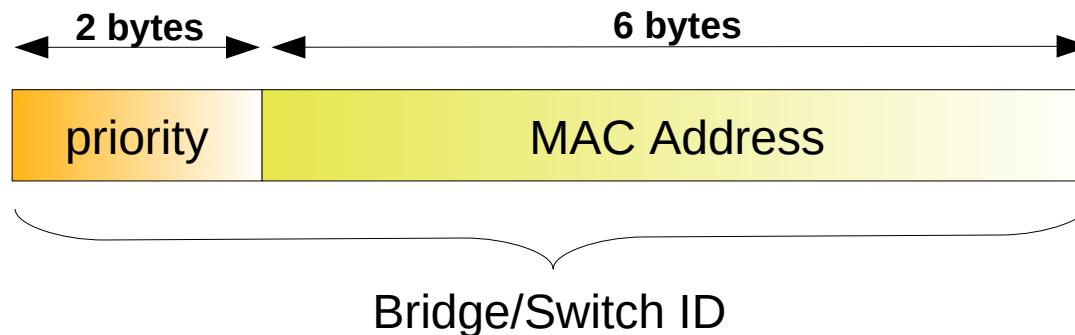


Spanning Tree Basic Concepts (1)



Spanning Tree Basic Concepts (2)

- Bridge/Switch ID – each switch is identified by an 8 bytes identifier based on:
 - ◆ 2 **Priority** bytes, defined by configuration.
 - ◆ 6 bytes (one of the **MAC Address** of the switch, or any other unique 48 bit sequence).
 - ◆ Priority has precedence over the 6 bytes sequence (usually MAC address).

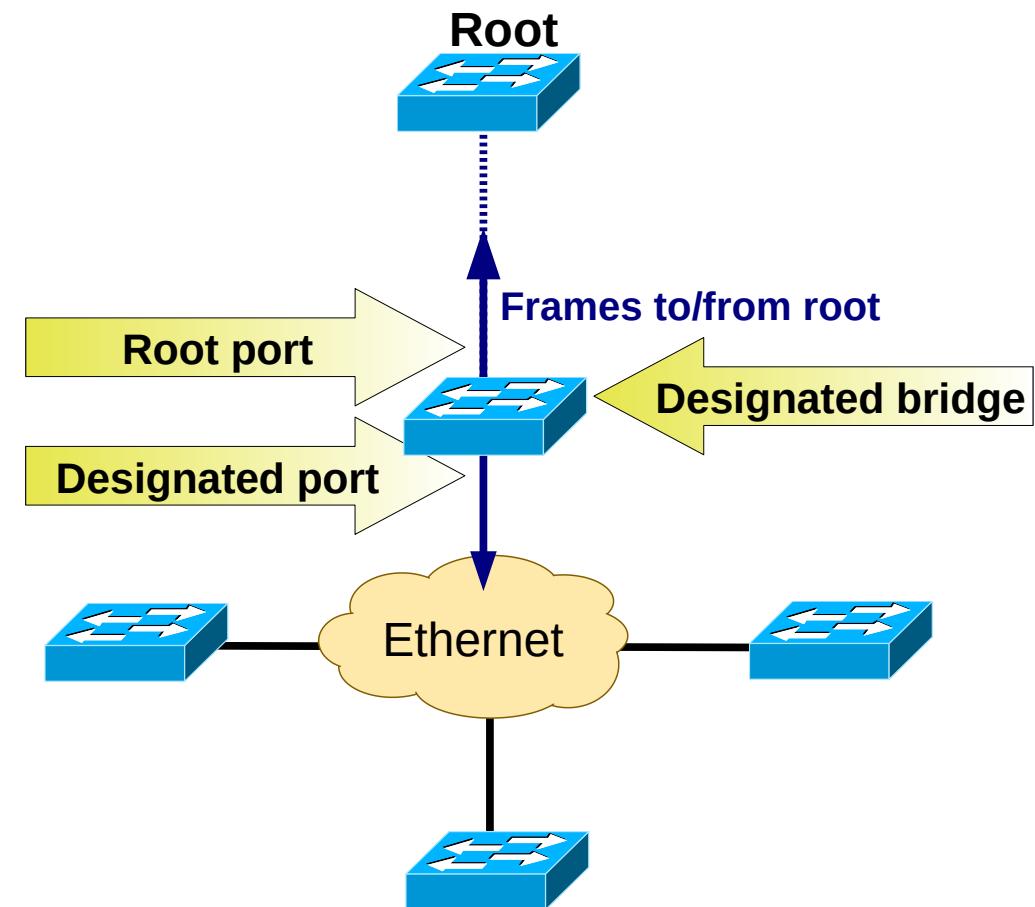


- Root Switch/bridge – Switch chosen as origin/root of the spanning tree.
 - ◆ Switch com **lowest ID**.
- Path cost – Cost associated with each port.
 - ◆ Has a default value, but can be changed by configuration.



Spanning Tree Basic Concepts (3)

- Designated Bridge – Switch responsible to forward the packets from an Ethernet segment to and from the root.
 - ◆ The root bridge is the designated bridge to all Ethernet segments connected to it.
- Designated Port – Port of the designated bridge that connects an Ethernet segment (to which is designated).
- Root Port – Port of the designated bridge that provides the path to the root.



Spanning Tree Basic Concepts (4)

- Possible Port States

- ◆ **Blocking state:**

- MAC address learning and packet forwarding are disabled;
 - Receives and processes BPDU.
 - After *MaxAge* time without receiving BPDU, it transitions to Listening state.

- ◆ **Listening state:**

- MAC address learning and packet forwarding are disabled;
 - Receives and processes BPDU.
 - When *ForwardDelay* timer expires the port transitions to Learning state.

- ◆ **Learning state:**

- Learns MAC address;
 - Packet forwarding are disabled;
 - Receives and processes BPDU.
 - When *ForwardDelay* timer expires the port transitions to Forwarding state.

- ◆ **Forwarding state:**

- MAC address learning and packet forwarding are enabled;
 - Receives and processes BPDU.

- ◆ **Disabled state:**

- MAC address learning and packet forwarding are disabled;
 - Does not receive BPDU.

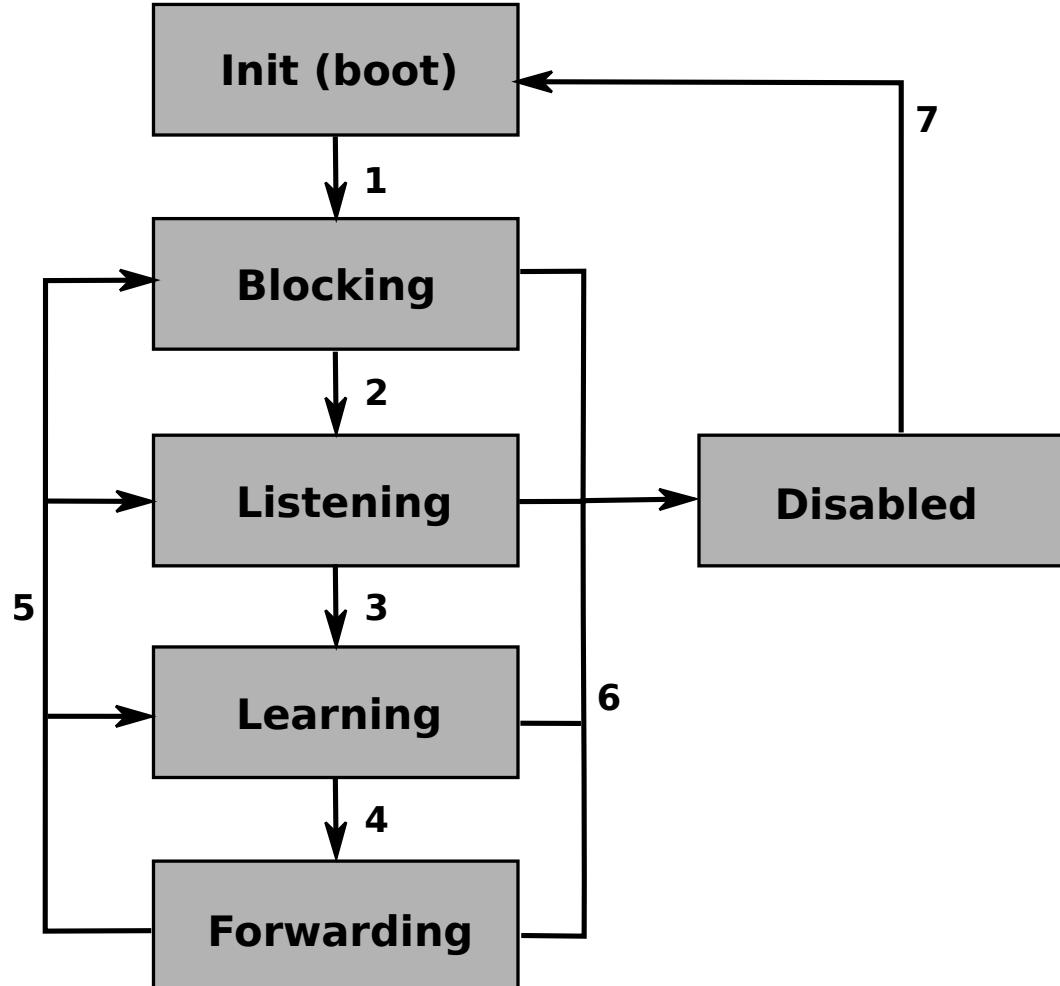


Spanning Tree Basic Concepts (5)

- Each switch has an associated cost of the shortest path to the root (Root Path Cost), given by the sum of the costs of all root ports along the path to the root.
- The Root Port, in each switch, is the port that provides the best path to the root (lowest Root Path Cost).
 - ◆ If more than one have the lowest cost, it is chosen the one with the neighbor with the lowest ID.
 - ◆ If more than one link is used to connect to the “best” neighbor it is used the one with the lowest (neighbor) port identifier.
- The Designated Bridge, from each Ethernet segment, is the switch with the lowest Root Path Cost from all connected to that segment.
 - ◆ If more than one have the lowest cost, it is chosen the one with the lowest ID.
- The Designated Port, from each Ethernet segment, is the port that connects it to its Designated Bridge.
- The root and designated ports will be in Forwarding state.
- All remaining ports will be in Blocking state.



Port States Diagram



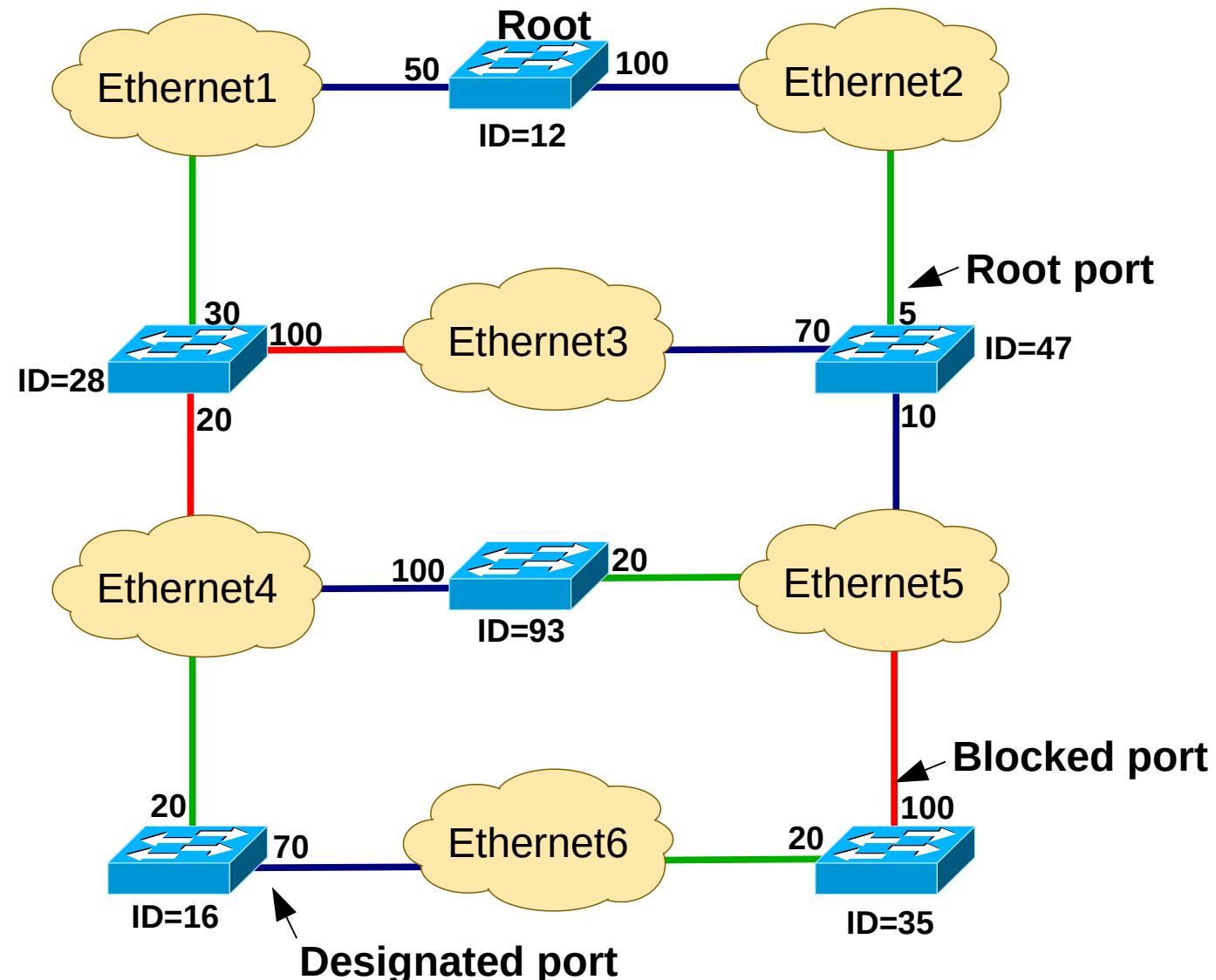
- 1) A port boots up and transitions to **Blocking** state.
- 2) When *MaxAge* timer expires the port transitions to **Listening** state.
- 3) When *ForwardDelay* timer expires the port transitions to **Learning** state.
- 4) When *ForwardDelay* timer expires the port transitions to **Forwarding** state.
- 5) After a topology change the port transitions immediately to **Blocking** state.
- 6) and 7) Administrative actions.



Example – Spanning Tree (1)

Designated bridges

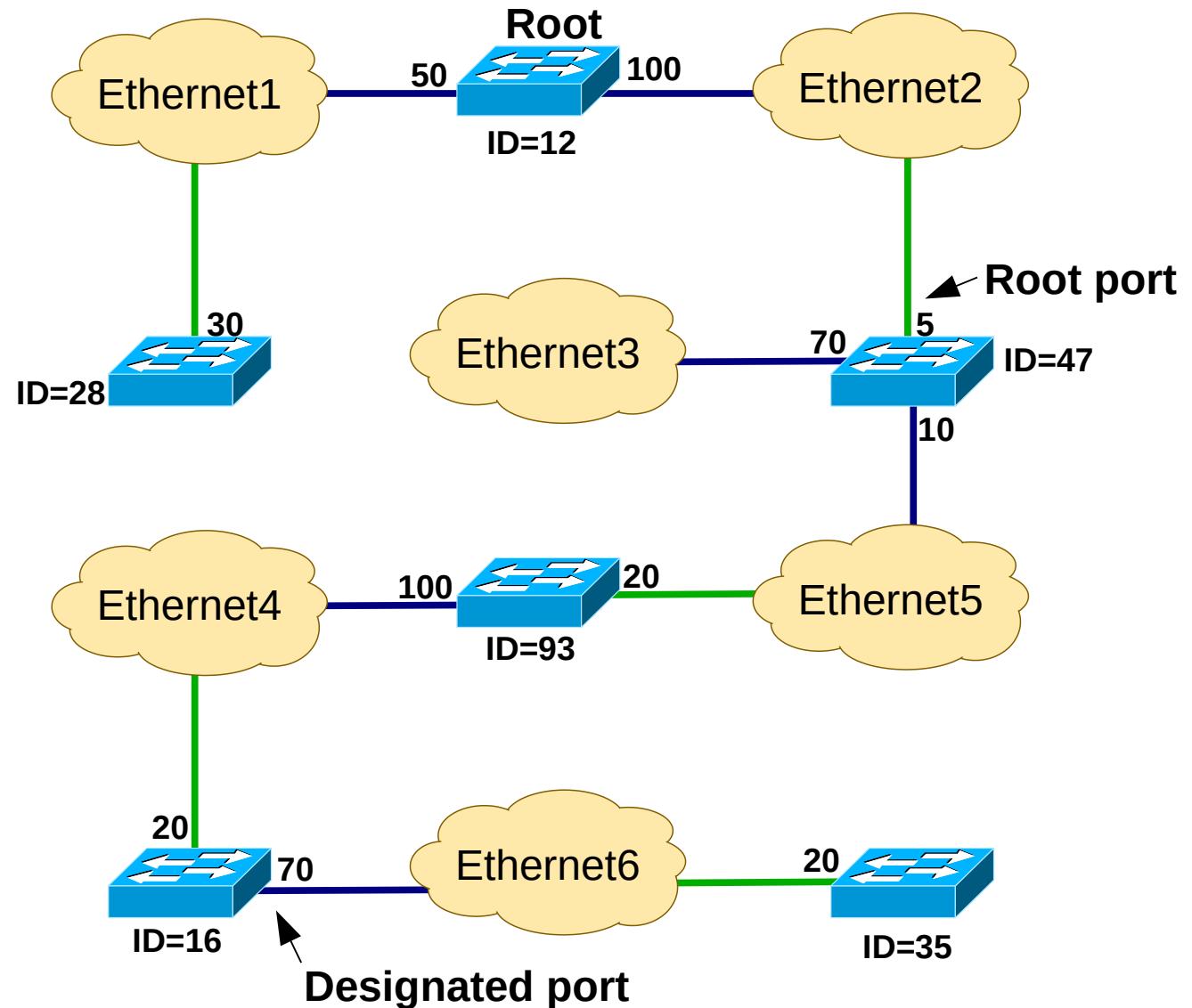
| | |
|-------|----|
| Eth1 | 12 |
| Eth 2 | 12 |
| Eth 3 | 47 |
| Eth 4 | 93 |
| Eth 5 | 47 |
| Eth 6 | 16 |



Example – Spanning Tree (2)

Designated bridges

| | |
|-------|----|
| Eth1 | 12 |
| Eth 2 | 12 |
| Eth 3 | 47 |
| Eth 4 | 93 |
| Eth 5 | 47 |
| Eth 6 | 16 |



Protocolo IEEE 802.1D

BPDUs (Bridge Protocol Data Units)

- To build the spanning tree, switches exchange special messages between them called Bridge Protocol Data Units (BPDU).
- There are two types: *Configuration e Topology Change Notification.*

IEEE 802.3 Ethernet

Destination: 01:80:c2:00:00:00 (01:80:c2:00:00:00)

Source: 00:16:e0:9a:c3:92 (00:16:e0:9a:c3:92)

Length: 39

Logical-Link Control

DSAP: Spanning Tree BPDU (0x42)

SSAP: Spanning Tree BPDU (0x42)

Control field: U, func=UI (0x03)

Spanning Tree Protocol

Protocol Identifier: Spanning Tree Protocol (0x0000)

Protocol Version Identifier: Spanning Tree (0)

BPDU Type: Configuration (0x00)

Root ID: 32768 / 00:05:1a:4e:fd:58

Root Path Cost: 200004

Bridge ID: 32768 / 00:16:e0:9a:c3:80

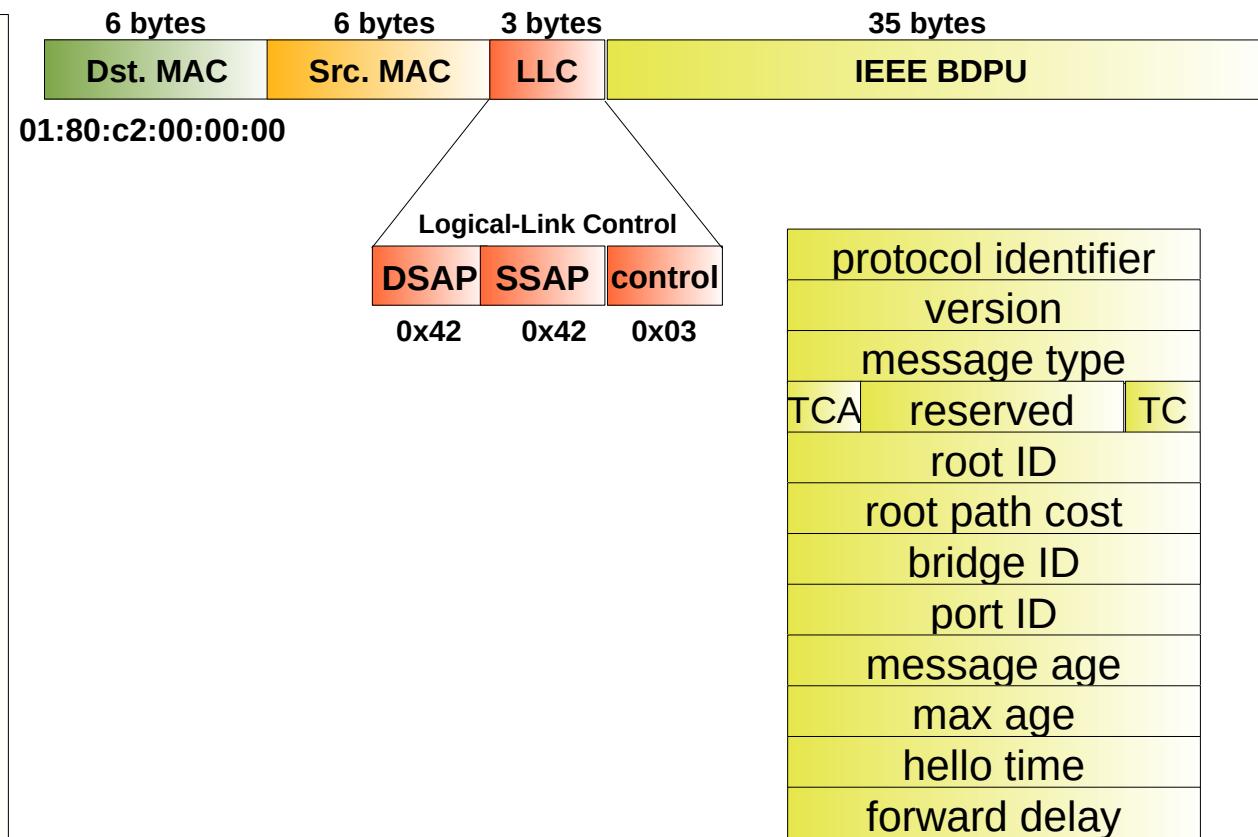
Port ID: 0x8012

Message Age: 1

Max Age: 20

Hello Time: 2

Forward Delay: 15



Configuration BPDU

- The setup of the Spanning Tree id done using Conf - BPDU (configuration messages).

IEEE 802.3 Ethernet

Destination: 01:80:c2:00:00:00 (01:80:c2:00:00:00)
Source: 00:16:e0:9a:c3:92 (00:16:e0:9a:c3:92)
Length: 39

Logical-Link Control

DSAP: Spanning Tree BPDU (0x42)
SSAP: Spanning Tree BPDU (0x42)
Control field: U, func=UI (0x03)

Spanning Tree Protocol

Protocol Identifier: Spanning Tree Protocol (0x0000)
Protocol Version Identifier: Spanning Tree (0)
BPDU Type: Configuration (0x00)

Root ID: 32768 / 00:05:1a:4e:fd:58

Root Path Cost: 200004

Bridge ID: 32768 / 00:16:e0:9a:c3:80

Port ID: 0x8012

Message Age: 1

Max Age: 20

Hello Time: 2

Forward Delay: 15

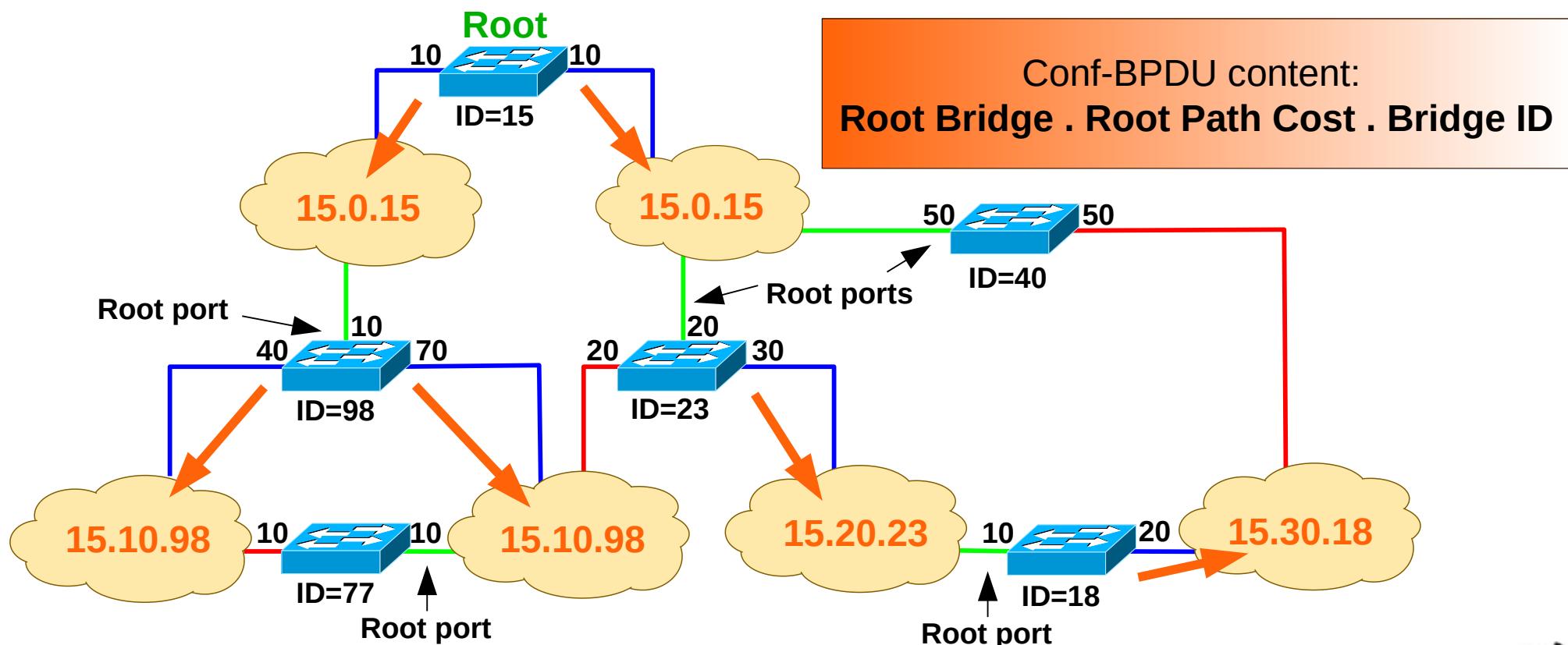
- More relevant fields:

- Root ID: ID of the current root bridge.
- Root Path Cost: estimation of the cost to the root.
- Bridge ID: own bridge identifier.
- Port ID: identifier of the port by which the BPDU was sent.
 - Port priority (1 byte) + Port number



Spanning Tree Maintenance

- Periodically switches sent Conf-BPDUs by its Designated Ports.
 - Periodicity of Conf-BPDU messages = hello time
 - Recommended Hello time: 2 seconds.
 - Defined at the root bridge.



Sorting of Best BPDU

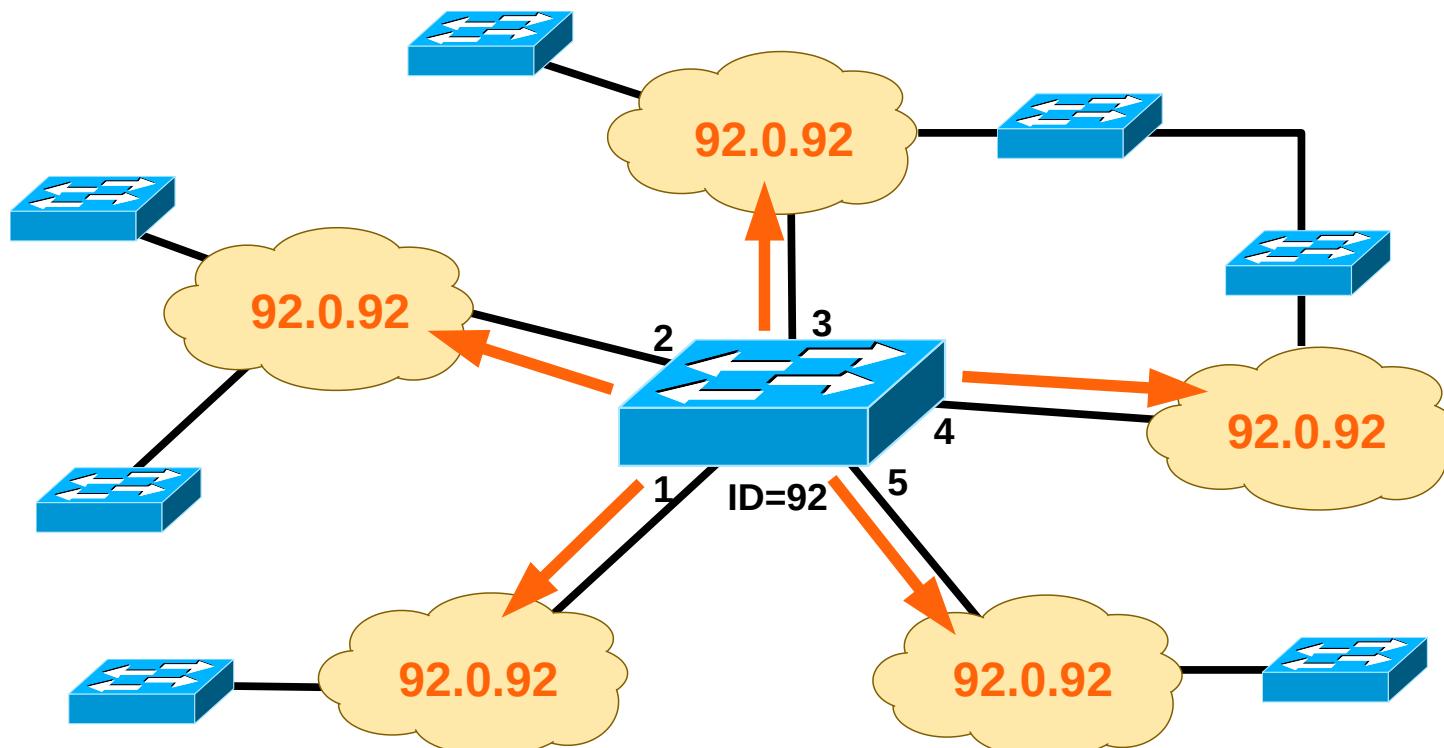
- A Conf-BPDU C1 is considered better than a Conf-BPDU C2 if:
 - ◆ The Root ID of C1 is lower than the one in C2,
 - ◆ With equal Root ID, if Root Path Cost of C1 is lower than the one in C2,
 - ◆ With equal Root ID and Root Path Cost, if the Bridge ID of C1 is lower than the one in C2,
 - ◆ With equal Root ID, Root Path Cost and Bridge ID, if the Port ID of C1 is lower than the one in C2.

| Root ID | Root Path Cost | Bridge ID | Port ID |
|---------|----------------|-----------|---------|
| 18 | 27 | 32 | 2 |
| 18 | 27 | 32 | 4 |
| 18 | 27 | 43 | 1 |
| 18 | 35 | 23 | 3 |
| 23 | 31 | 45 | 2 |

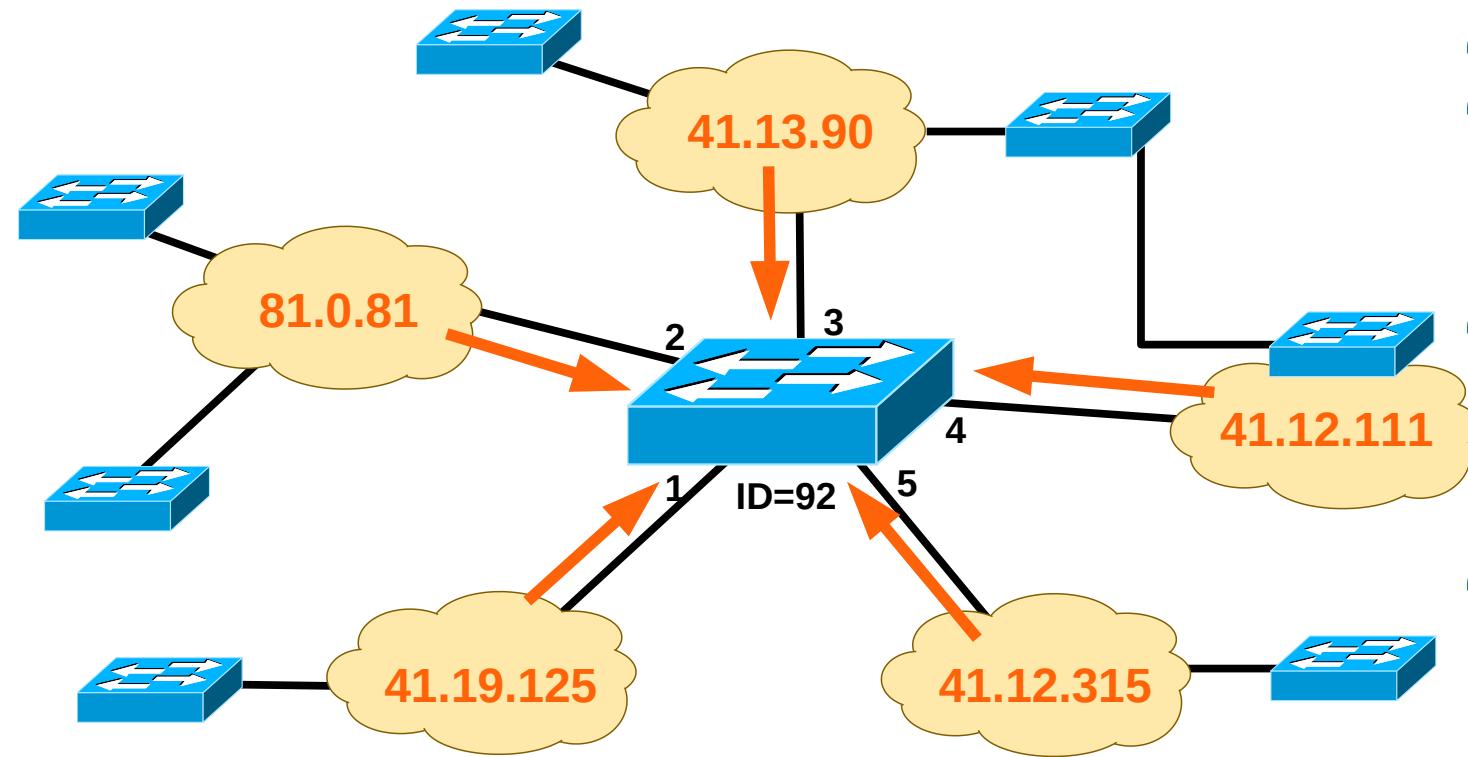


Building the Spanning Tree (1)

- Each switch initially assumes to be the Root Bridge.
 - ◆ Assumes Root Path Cost = 0,
 - ◆ Sends Conf-BPDU to all its ports.



Building the Spanning Tree (2)



Best Conf-BPDU received by Bridge 92 (until now)

Estimations of Bridge 92 (assuming port costs equal to 1).

- Bridge92 is not root (BridgeID 92>41)
- Bridge 92 Root Port is 4.
 - Lowest RootID (41).
 - Lowest Root Path Cost ($12+1=13$).
 - Lowest Neighbor BridgeID ($111 < 315$)
- Bridge 92 is Designated Bridge via ports 1 and 2
 - Port 2, Lowest RootID (41).
 - Port 1, Same RootID (41) and Lowest Root Path Cost ($13 < 19$).
- Bridge 92 ports 3 and 5 are blocked.
 - Neighbors have the same RootID (41).
 - Via port 3, Neighbor has the same Root Path Cost (13), but lower BridgeID ($90 < 92$).
 - Via port 5, Neighbor has lower Root Path Cost (12).

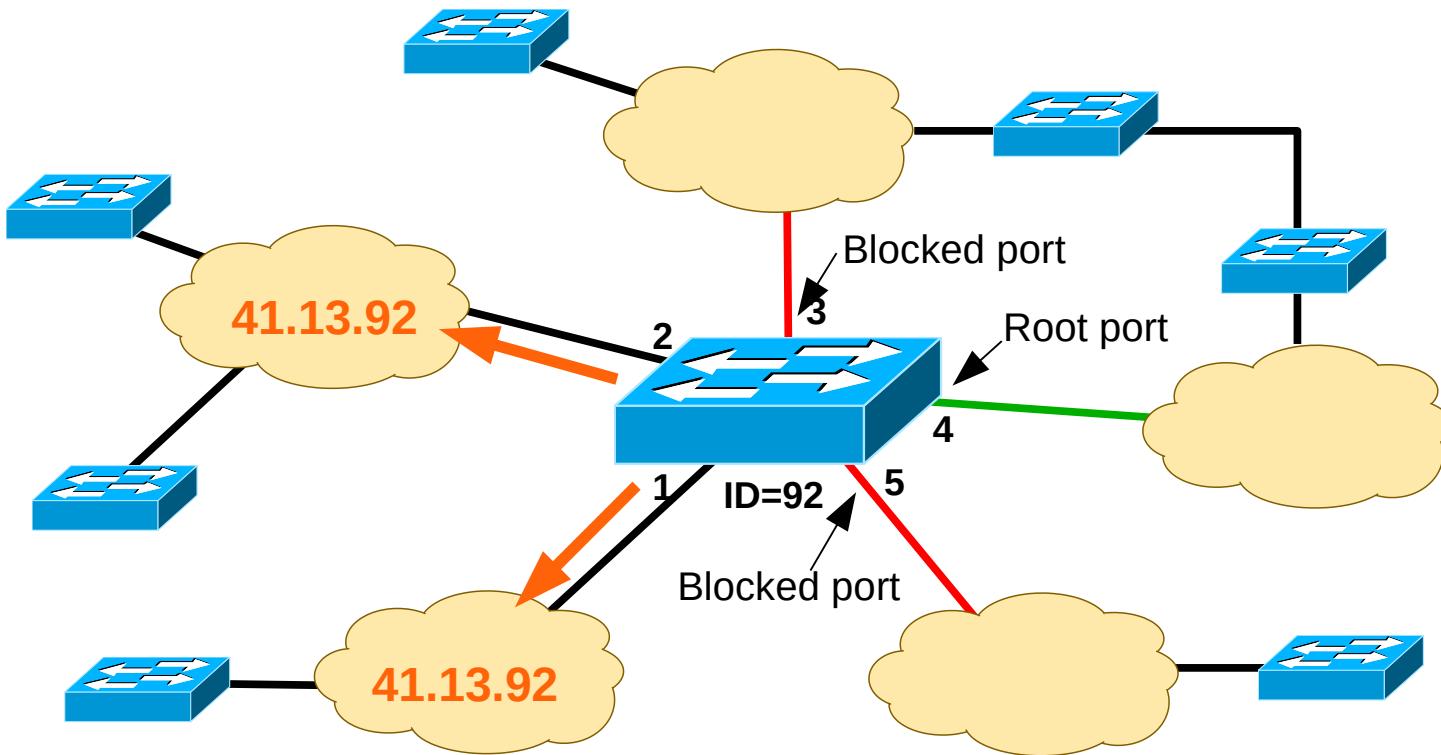
Root Bridge = 41

Root port = 4

Root Path Cost = $12 + 1 = 13$



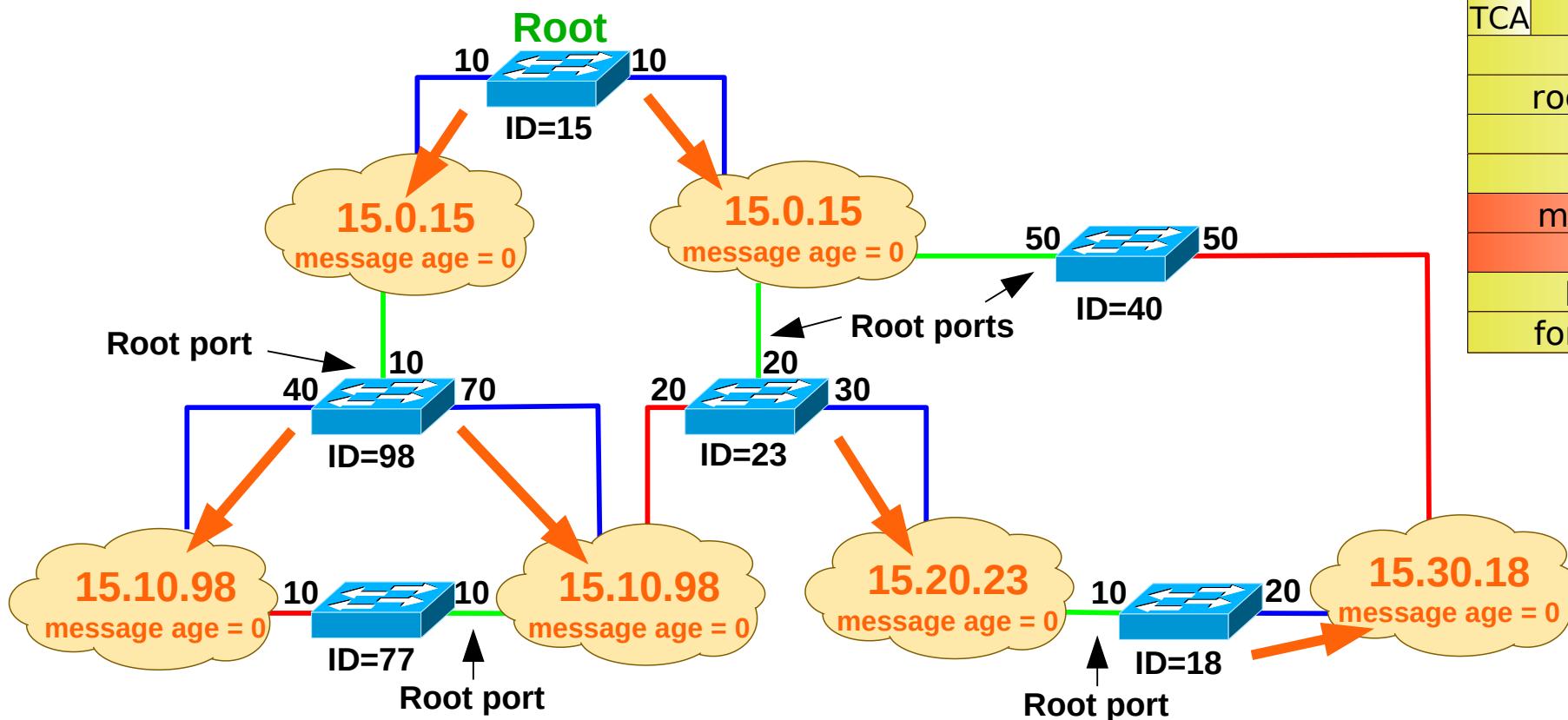
Building the Spanning Tree (3)



Conf-BPDU sent by Bridge 92 - **41.13.92**



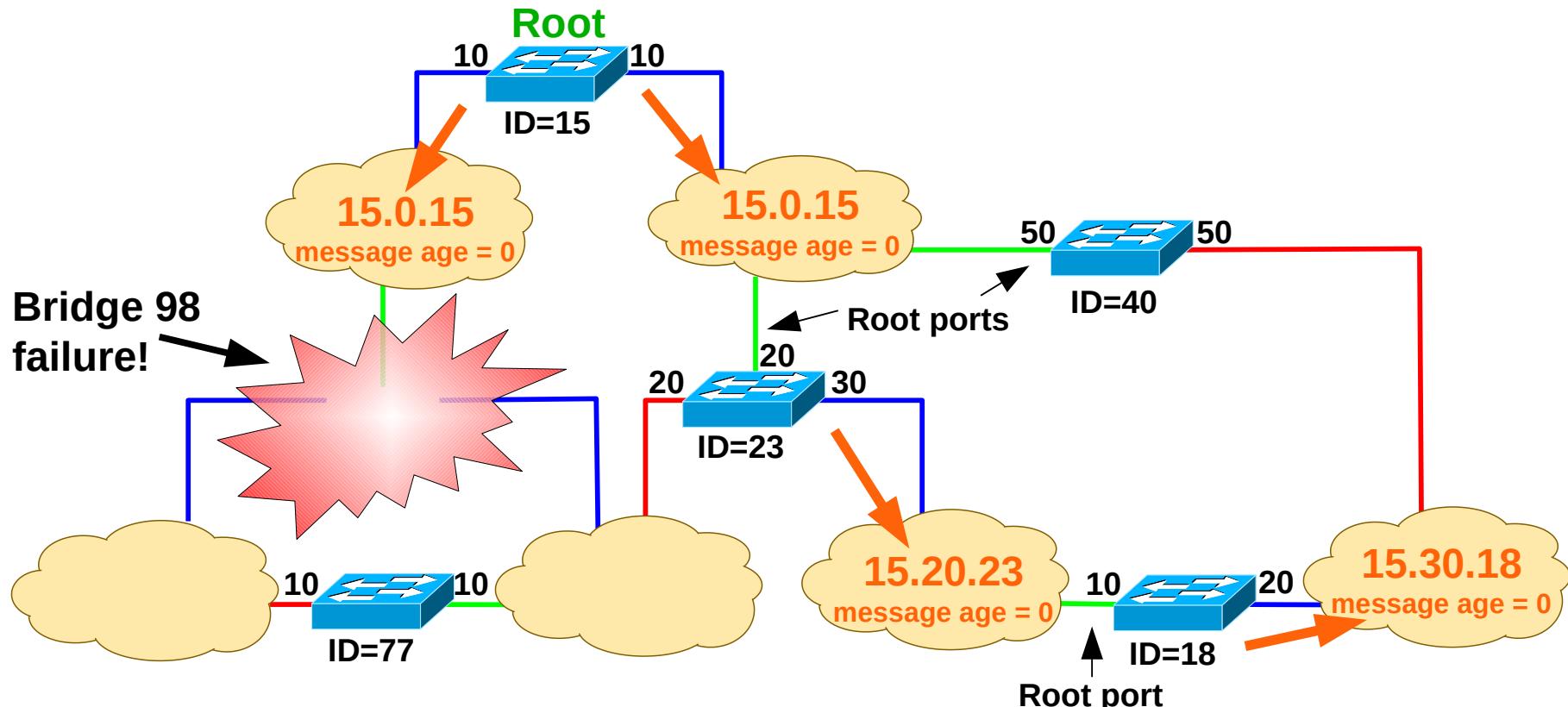
Network Failures (1)



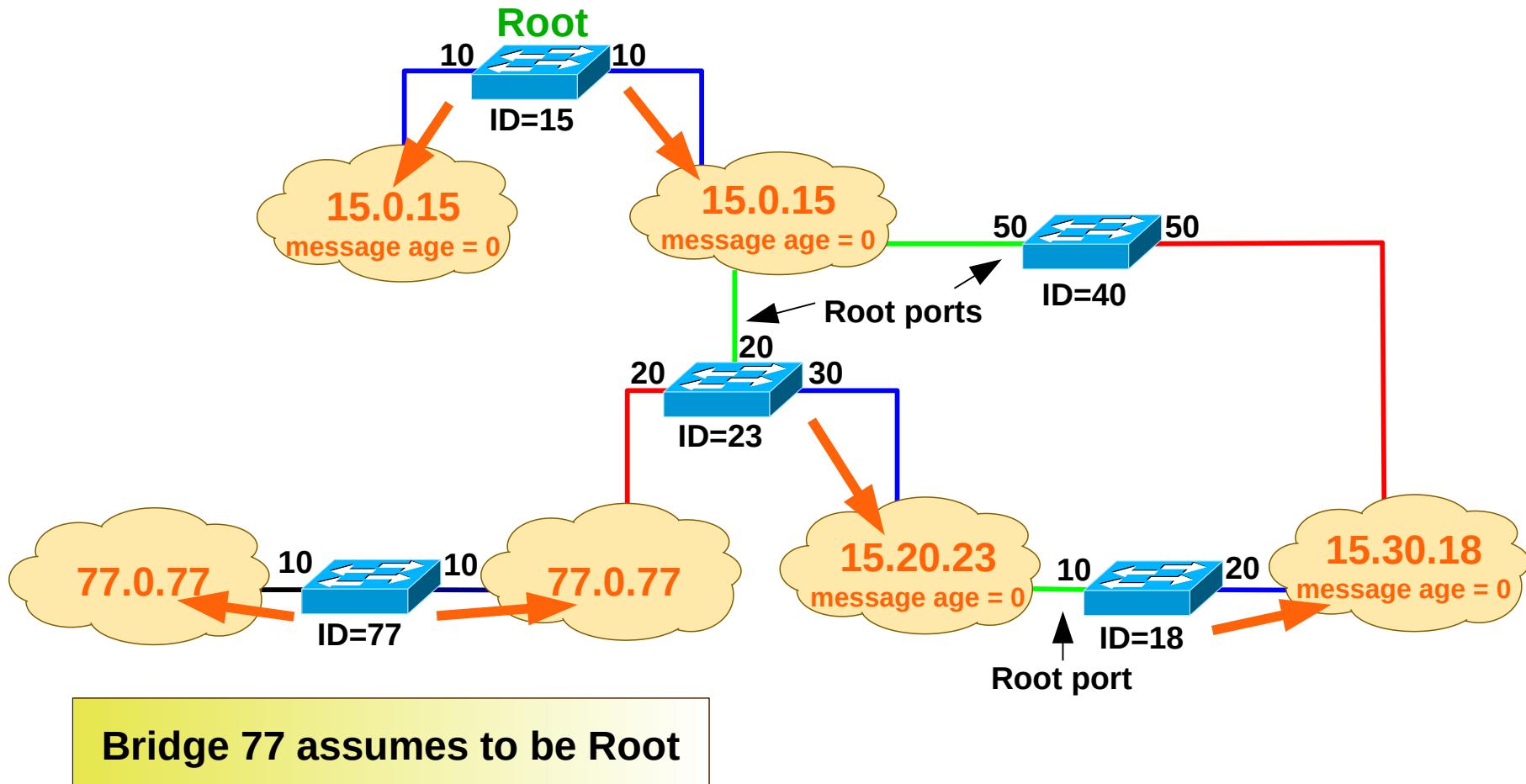
| Conf-BPDU | | |
|---------------------|----------|----|
| protocol identifier | | |
| version | | |
| message type | | |
| TCA | reserved | TC |
| root ID | | |
| root path cost | | |
| bridge ID | | |
| port ID | | |
| message age | | |
| max age | | |
| hello time | | |
| forward delay | | |



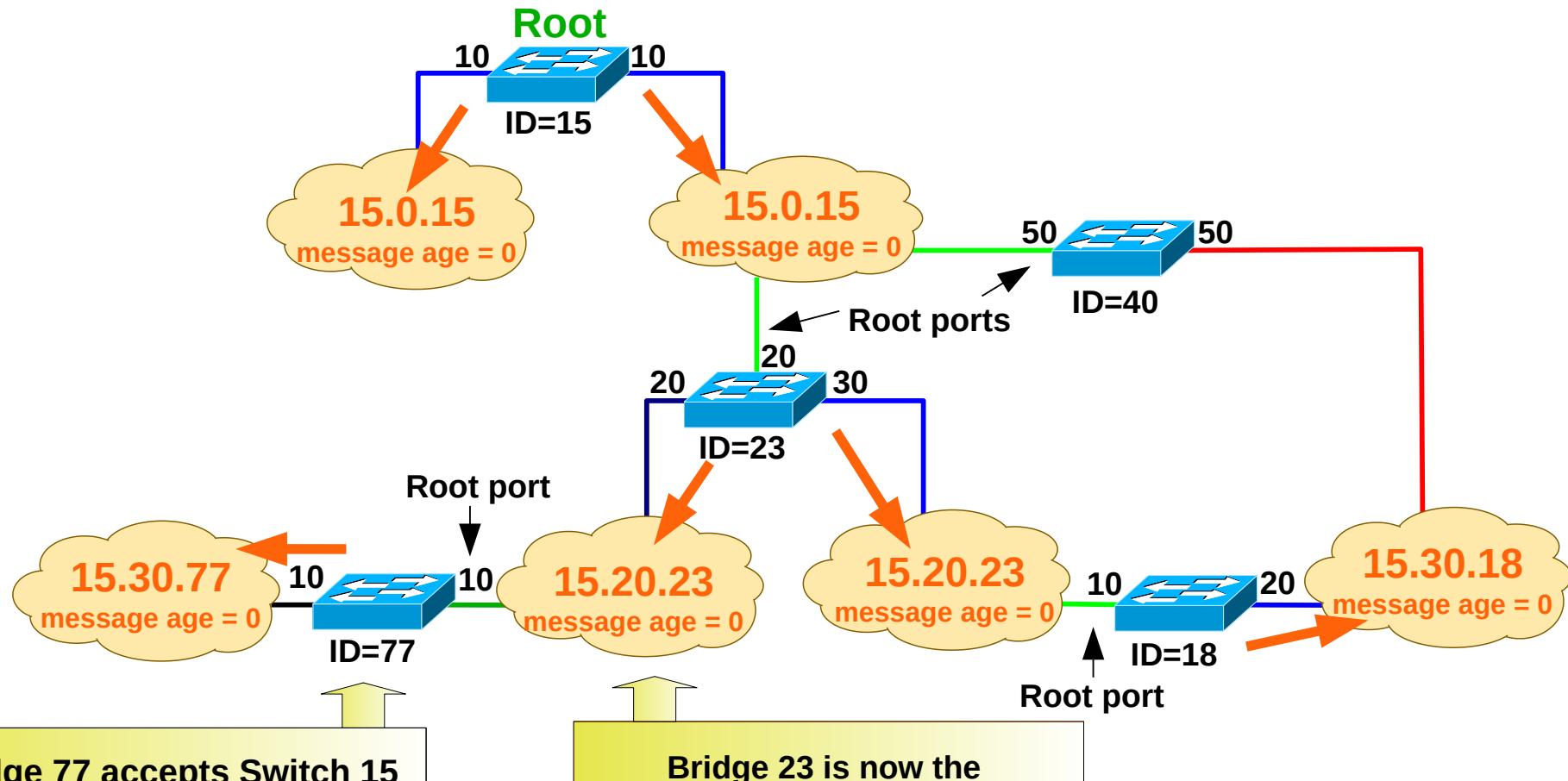
Network Failures (2)



Network Failures (3)



Network Failures (4)



Forwarding Tables Entries Lifetimes

- Forwarding Tables Long Lifetime – Many frames will be lost when network is changing topology.
- Forwarding Tables Short Lifetime – Creates too much traffic due to frequent flooding.
- There are two forwarding tables lifetimes:
 - ◆ **Long**: used by default (recommended value = 300 seconds)
 - ◆ **Short**: used when SPT is re-configuring (recommended value = 15 seconds)



Topology Change Notification

Conf (Configuration) BPDU

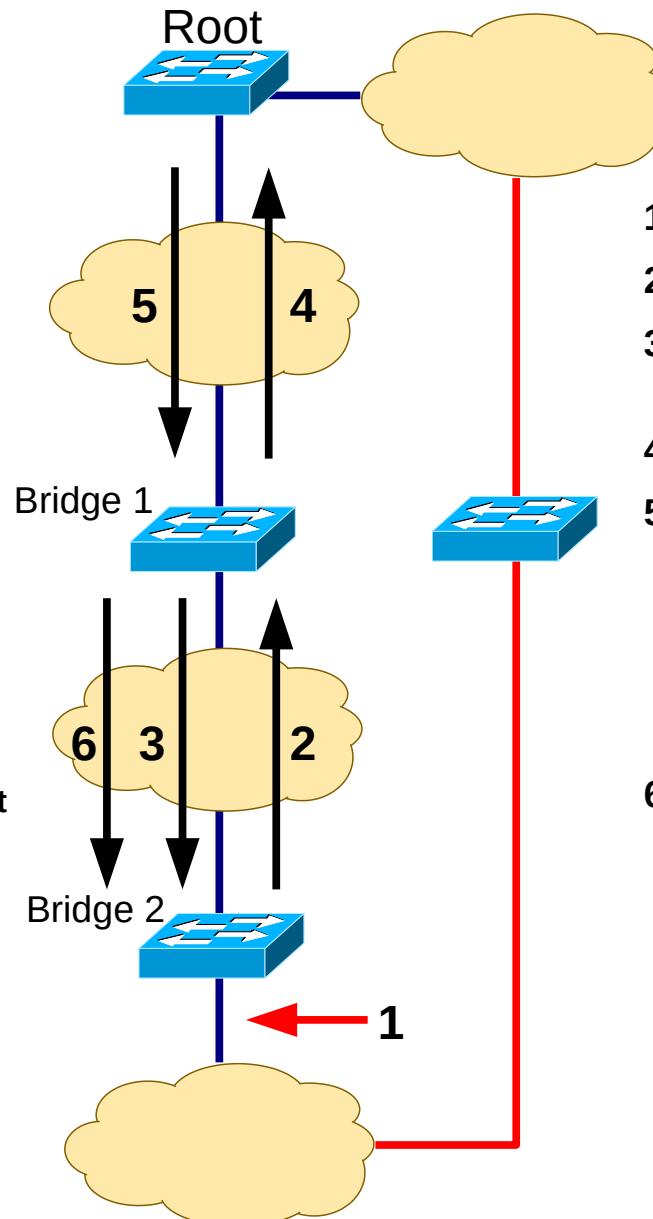
| | | |
|---------------------|----------|----|
| protocol identifier | | |
| version | | |
| message type = 0 | | |
| TCA | reserved | TC |
| root ID | | |
| root path cost | | |
| bridge ID | | |
| port ID | | |
| message age | | |
| max age | | |
| hello time | | |
| forward delay | | |

TCA - flag Topology Change Acknowledgment

TC - flag Topology Change

TCN (Topology Change Notification)
BPDU

| |
|---------------------|
| protocol identifier |
| version |
| message type = 1 |



1. Port changes state to disabled or blocking
2. Sends TCN-BPDU (periodicity = hello time)
3. Sends Conf-BPDU with TCA = 1 while receiving TCN-BPDU
4. Sends TCN-BPDU (periodicity = hello time)
5. Sends Conf-BPDU with TCA = 1 while receiving TCN-BPDU and with TC=1 for a period of time equal to *ForwardDelay* + *MaxAge*

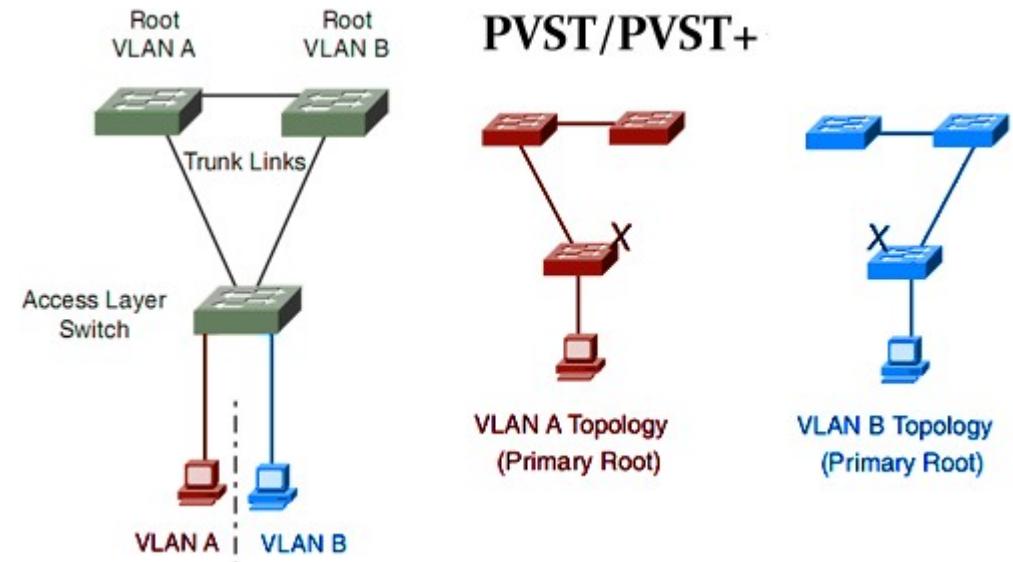
Root bridge uses the forwarding table short lifetime during this period

6. Sends Conf-BPDU with TC=1
- Bridge 1 uses the forwarding table short lifetime while receiving Conf-BPDU with TC=1
- Bridge 2 uses the forwarding table short lifetime while receiving Conf-BPDU with TC=1



Other Protocols (1)

- Cisco's proprietary versions of SPT are:
 - ↳ Per-VLAN Spanning Tree (PVST).
 - ↳ Per-VLAN Spanning Tree Plus (PVST+).
- ↳ Create a different spanning tree for each VLAN.
 - ↳ Different roots, costs, blocked ports, etc...
 - ↳ In a complex switching network some switches may not have ports of all VLAN.



```
Ethernet II, Src: c2:00:05:7f:f1:01 (c2:00:05:7f:f1:01), Dst: PVST+ (01:00:0c:cc:cc:cd)
802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 1
    000. .... .... = Priority: 0
    ...0 .... .... = CFI: 0
    .... 0000 0000 0001 = ID: 1
Length: 50
Logical-Link Control
Spanning Tree Protocol
    Protocol Identifier: Spanning Tree Protocol (0x0000)
    Protocol Version Identifier: Spanning Tree (0)
    BPDU Type: Configuration (0x00)
    BPDU flags: 0x00
    Root Identifier: 32768 / 0 / c2:00:05:7f:00:00
    Root Path Cost: 0
    Bridge Identifier: 32768 / 0 / c2:00:05:7f:00:00
    Port identifier: 0x802a
    Message Age: 0
    Max Age: 20
    Hello Time: 2
```

Identificador da VLAN



Other Protocols (2)

- IEEE 802.1p
 - ◆ Extension of IEEE 802.1Q.
 - ◆ Provides QoS based on relative priorities.
 - ◆ Defines the field *User Priority* (3 bits) that allows 8 levels of priority.
 - ◆ The standard recommends:
 - ✚ Priority 7 : Critical traffic,
 - ✚ Priorities 5–6 : Delay sensitive traffic (voice and live video),
 - ✚ Priorities 1–4 : Delay variation sensitive traffic (*streaming*),
 - ✚ Priority 0 : Other traffic.



Other Protocols (3)

- IEEE 802.1w Rapid Spanning Tree Protocol

- Extension of IEEE 802.1D.
- Speeds up the convergence time of the Spanning Tree in case of topology changes
 - There are only three port states in RSTP that correspond to the three possible operational states.
 - Adds two additional port roles to a port when in blocking state
 - Alternate port: possible alternative Root port.
 - Backup port: possible alternative Designated port.
- Adds a negotiated mechanism between switches.
 - Uses the reserved bits in the Conf-BPDU.

| STP (802.1D) Port State | RSTP (802.1w) Port State | Is Port Included in Active Topology? | Is Port Learning MAC Addresses? |
|-------------------------|--------------------------|--------------------------------------|---------------------------------|
| Disabled | Discarding | No | No |
| Blocking | Discarding | No | No |
| Listening | Discarding | Yes | No |
| Learning | Learning | Yes | Yes |
| Forwarding | Forwarding | Yes | Yes |

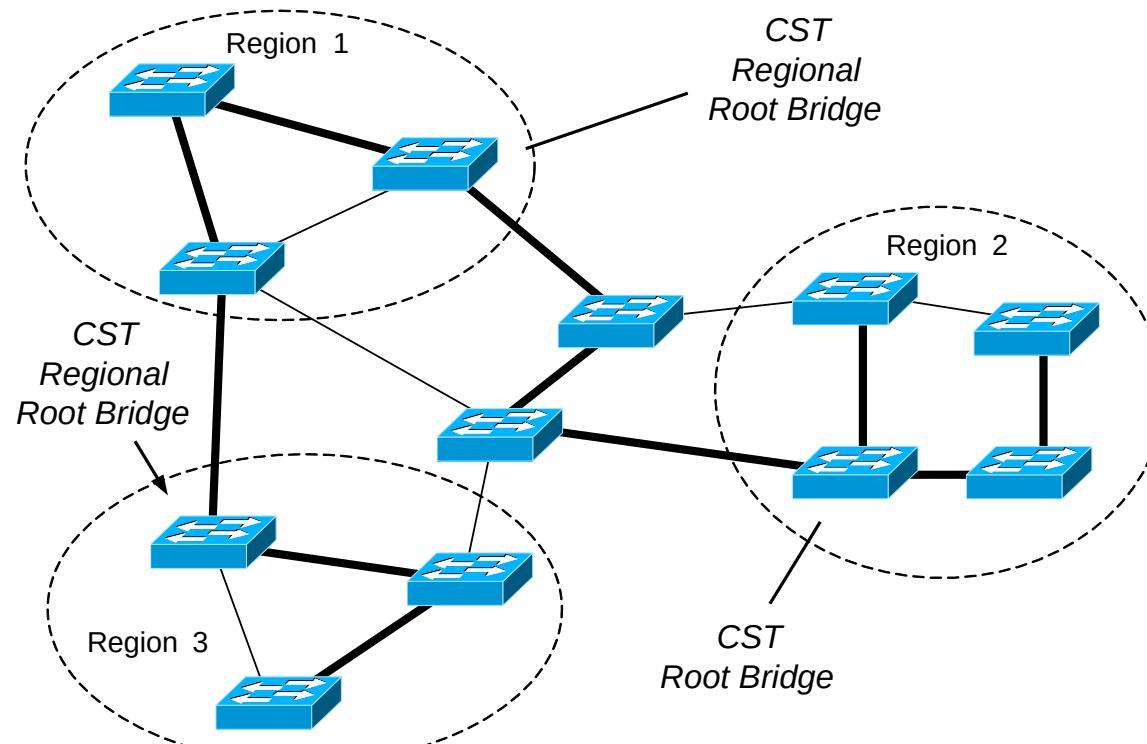
Conf (Configuration) BPDU

| | | |
|---------------------|----------|----|
| protocol identifier | | |
| version | | |
| message type = 0 | | |
| TCA | reserved | TC |
| root ID | | |
| root path cost | | |
| bridge ID | | |
| port ID | | |
| message age | | |
| max age | | |
| hello time | | |
| forward delay | | |



Other Protocols (4)

- IEEE 802.1s Multiple Spanning Tree Protocol
 - Creates multiple Spanning Trees.
 - Allows the assignment of a set of several VLAN to a specific Common Spanning Tree (CST).
 - CST are usually mapped to regions of the network.

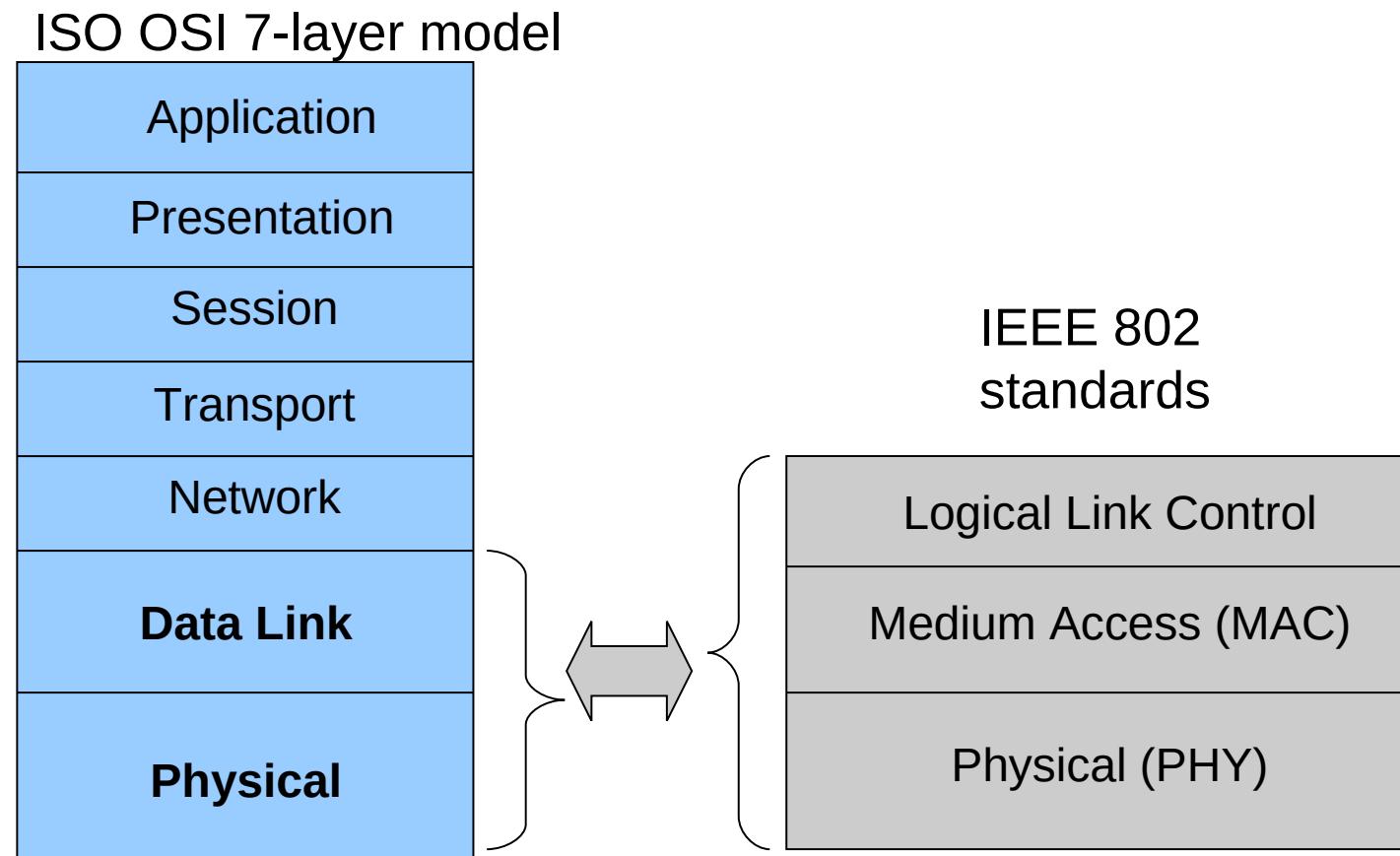


Wi-Fi



Standardization of Wireless Networks

- Wireless networks are standardized by the IEEE under the 802 LAN MAN standards committee.



Wireless Networks

- Networks are designed according to the number of users and coverage area
- There are several scales on the number of users and coverage area
 - ◆ Local: LANs → IEEE 802.11
 - ◆ Personal: PANs → e.g. Bluetooth, ZigBee
 - ◆ Regional: WANs → GSM, UMTS, LTE, 5G, LoRa,...
 - ◆ Worldwide : Satellite → Iridium, SpaceX Starlink?



Wireless LAN: Overview

- Two Types
 - ◆ Infra-structured,
 - ◆ Ad-hoc.
- Advantages
 - ◆ Flexible installation (minimum cables).
 - ◆ More robust (no cable problems).
 - ◆ One-time installation (conferences, historic buildings).
- Problems
 - ◆ Many proprietary solutions.
 - ◆ Restrictions on the electromagnetic spectrum.
 - ◆ Subject to frame collision when accessing the transmission medium.
 - ◆ More on this later.
 - ◆ Lower bandwidths than cabled networks.



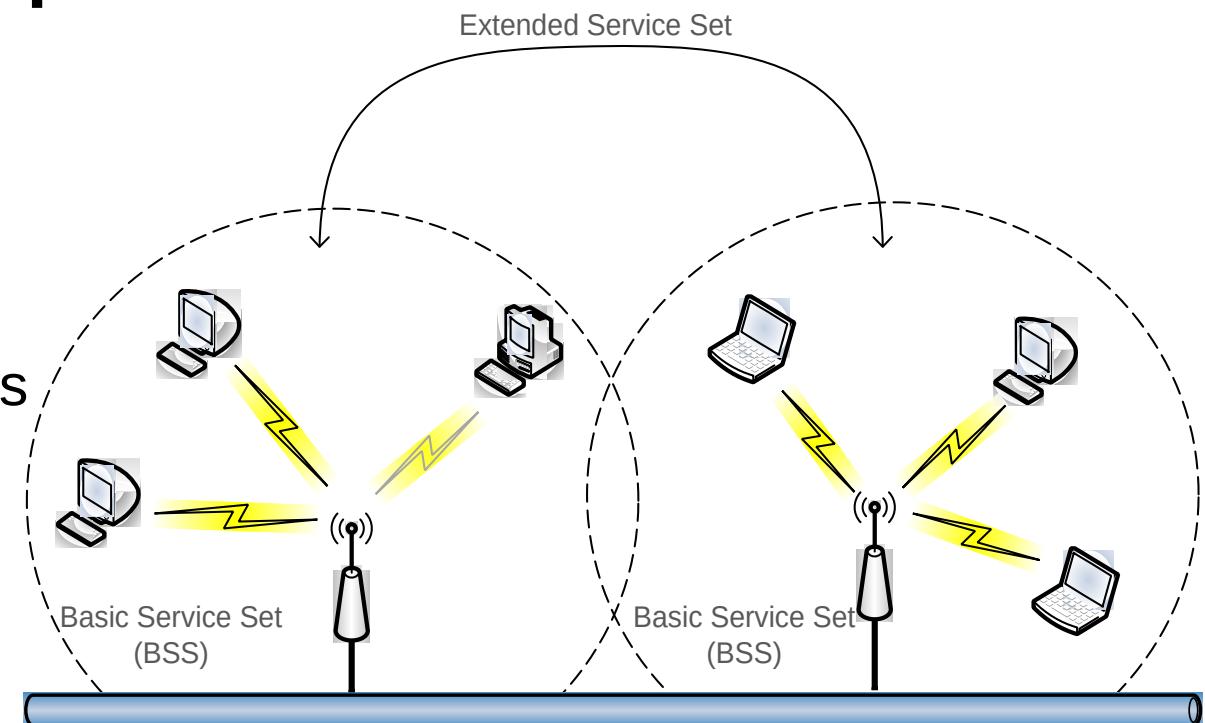
Evolution of WLAN standards

- WiFi 1 - 802.11b, 1999, 2.4 GHz band, 11 Mbps data rate
- WiFi 2 - 802.11a, 1999, 5 GHz band, 54 Mbps data rate
- WiFi 3 - 802.11g, 2003, 2.4 GHz band, 54 Mbps data rate
- WiFi 4 - 802.11n, 2009, 2.4 and 5 GHz bands, ~600 Mbps data rate
- WiFi 5 - 802.11ac, 2013, 5 GHz band, ~1.3 Gbps data rate
- WiFi 6 - 802.11ax, 2019, 1 to 7GHz bands, >11Gbps data rate



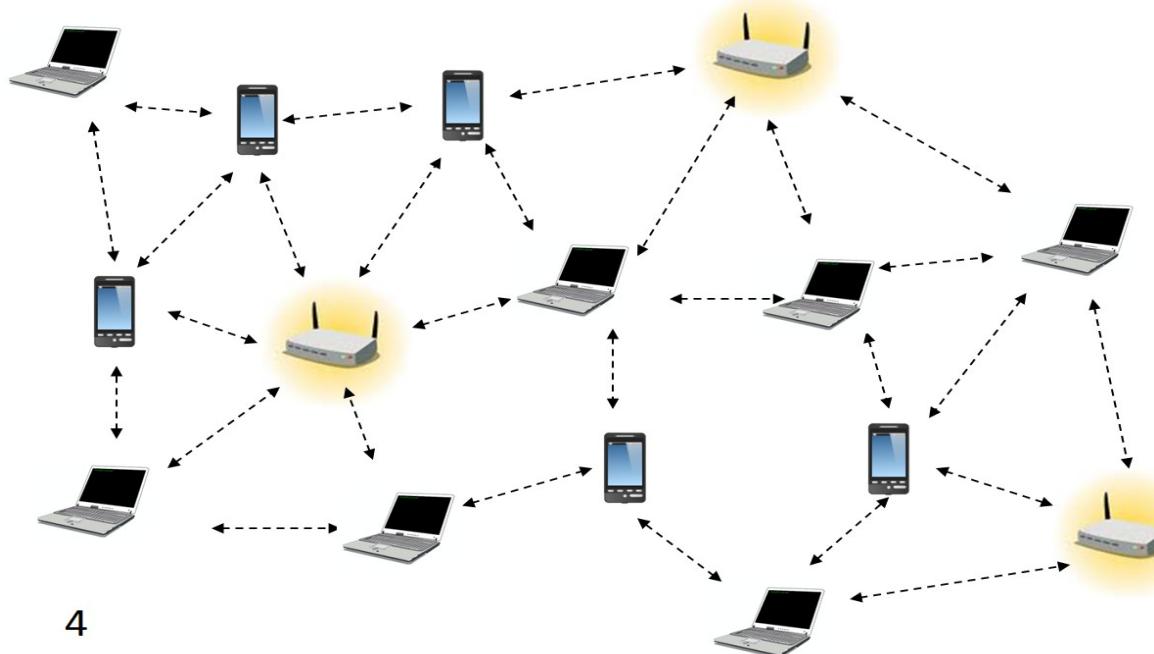
Components

- Station (STA)
 - ◆ Mobile terminal
- Access Point (AP)
 - ◆ STA connect to access points (infra-structured networks)
- Basic Service Set (BSS)
 - ◆ STA and AP with same coverage form a BSS
 - ◆ Group of IEEE 802.11 stations associated to an Access Point (AP)
 - ◆ Known through the SSID
- Extended Service Set (ESS)
 - ◆ Several BSSs interconnected by APs form a ESS



Ad-hoc Networks (IBSS)

- Temporary set of stations
- Forming an ad-hoc network – an independent BSS (IBSS), means that there is no connection to a wired network
- No AP
- No relay function (direct connection)
- Simple setup



4



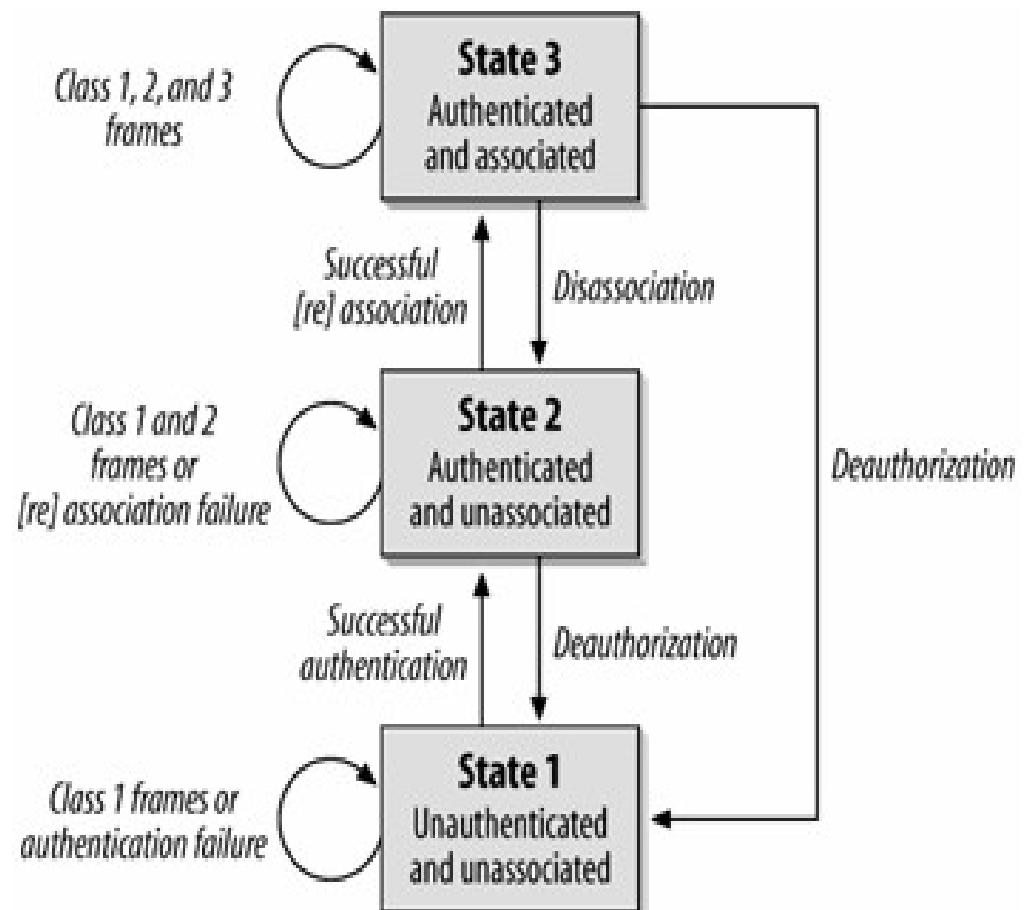
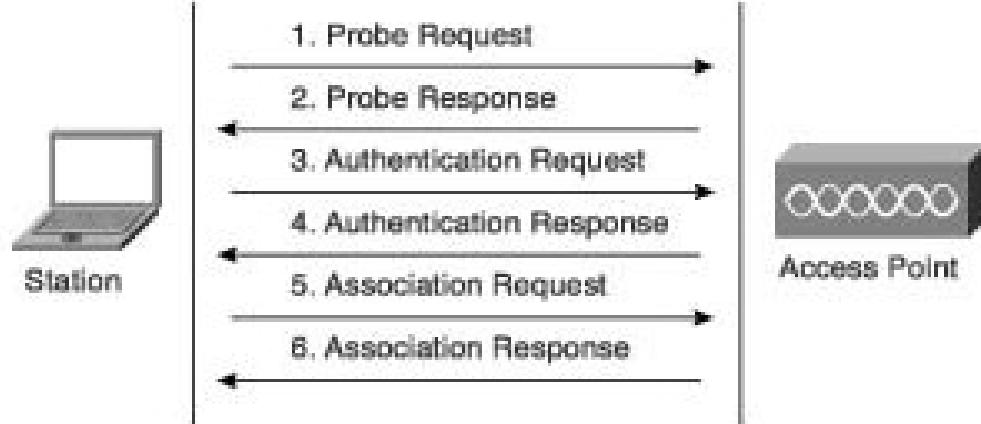
IEEE 802.11 services

- Station services (similar to wired network)
 - ◆ Authentication (login)
 - ◆ De-authentication (logout)
 - ◆ Privacy
 - ◆ Data delivery
- Distribution services
 - ◆ Association
 - ✚ Make logical connection between the AP and the station – the AP will not receive any data from a station before association
 - ◆ Re-association (similar to association)
 - ✚ Send repeatedly to the AP.
 - ✚ Help the AP to know if the station has moved from/to another BSS.
 - ✚ After Power Save
 - ◆ Disassociation
 - ✚ Manually disconnect (PC is shutdown or adapter is ejected)



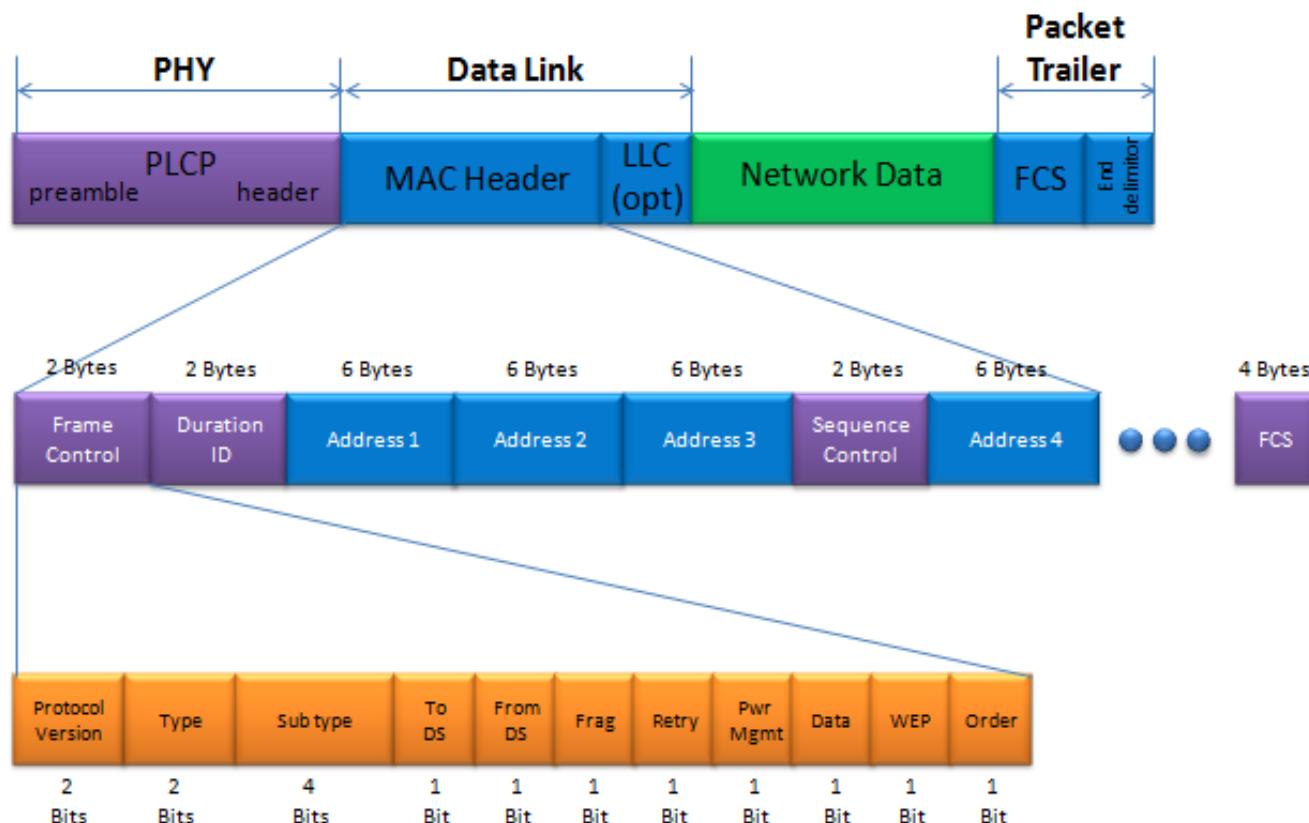
Joining a BSS

- Station finds BSS/AP by **Scanning/Probing**.
- BSS with AP: both **Authentication** and **Association** are necessary for joining a BSS.



WLAN Frames

- Three types of frames
 - ◆ Control: RTS, CTS, ACK
 - ◆ Management
 - ◆ Data
- Header is different for the different types of frames.



Joining BSS with AP: Scanning

- A station willing to join a BSS must get in contact with the AP. This can happen through:
 - 1. Passive scanning
 - The station scans the channels for a Beacon frame that is sent periodically from an AP to announce its presence and provide the SSID, and other parameters for WNICs within range
 - 2. Active scanning (the station tries to find an AP)
 - The station sends a Probe Request frame - Sent from a station when it requires information from another station
 - All AP's within reach reply with a Probe Response frame - Sent from an AP containing capability information, supported data rates, etc., after receiving a probe request frame



Beacon Frame

- IEEE 802.11 Beacon frame, Flags:c
 - Type/Subtype: Beacon frame (0x0008)
 - › Frame Control Field: 0x8000
 - .000 0000 0000 0000 = Duration: 0 microseconds
 - Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
 - Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
 - Transmitter address: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
 - Source address: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
 - BSS Id: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
 - 0000 = Fragment number: 0
 - 1001 1000 1010 = Sequence number: 2442
 - Frame check sequence: 0x6f0b825c [unverified]
 - [FCS Status: Unverified]
- IEEE 802.11 wireless LAN
 - › Fixed parameters (12 bytes)
 - Timestamp: 660070796
 - Beacon Interval: 0.102400 [Seconds]
 - › Capabilities Information: 0x0421
 - › Tagged parameters (123 bytes)
 - › Tag: SSID parameter set: LABC0M
 - › Tag: Supported Rates 1(B), 2(B), 5.5(B), 6, 9, 11(B), 12, 18, [Mbit/sec]
 - › Tag: DS Parameter set: Current Channel: 13
 - › Tag: Traffic Indication Map (TIM): DTIM 0 of 0 bitmap
 - › Tag: ERP Information
 - › Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]
 - › Tag: Cisco CCX1 CKIP + Device Name
 - › Tag: Vendor Specific: Microsoft Corp.: WMM/WME: Parameter Element
 - › Tag: Vendor Specific: Cisco Systems, Inc.: Aironet Unknown (1) (1)
 - › Tag: Vendor Specific: Cisco Systems, Inc.: Aironet CCX version = 5
 - › Tag: Vendor Specific: Cisco Systems, Inc.: Aironet Unknown (11) (11)
 - › Tag: Vendor Specific: Cisco Systems, Inc.: Aironet Client MFP Disabled



Probe Request/Response Frames

- IEEE 802.11 Probe Request, Flags:C

Type/Subtype: Probe Request (0x0004)
Frame Control Field: 0x4000
.000 0000 0000 = Duration: 0 microseconds
Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
Transmitter address: Microsoft_0a:43:e3 (c0:33:5e:0a:43:e3)
Source address: Microsoft_0a:43:e3 (c0:33:5e:0a:43:e3)
BSS Id: Broadcast (ff:ff:ff:ff:ff:ff)
.... 0000 = Fragment number: 0
1100 1011 0001 = Sequence number: 3249
Frame check sequence: 0xc7056d0a [unverified]
[FCS Status: Unverified]

- IEEE 802.11 wireless LAN

- Tagged parameters (62 bytes)
 - › Tag: SSID parameter set: TD_WIFI_GUEST
 - › Tag: Supported Rates 1, 2, 5.5, 6, 9, 11, 12, 18, [Mbit/sec]
 - › Tag: DS Parameter set: Current Channel: 13
 - › Tag: HT Capabilities (802.11n D1.10)
 - › Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]

- IEEE 802.11 Probe Response, Flags:C

Type/Subtype: Probe Response (0x0005)
Frame Control Field: 0x5000
.000 0001 0011 1010 = Duration: 314 microseconds
Receiver address: IntelCor_d2:98:58 (28:b2:bd:d2:98:58)
Destination address: IntelCor_d2:98:58 (28:b2:bd:d2:98:58)
Transmitter address: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
Source address: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
BSS Id: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
.... 0000 = Fragment number: 0
1010 0010 1001 = Sequence number: 2601
Frame check sequence: 0x80831320 [unverified]
[FCS Status: Unverified]

- IEEE 802.11 wireless LAN

- Fixed parameters (12 bytes)
 - Timestamp: 664064263
 - Beacon Interval: 0.102400 [Seconds]
 - Capabilities Information: 0x0421
- Tagged parameters (117 bytes)
 - › Tag: SSID parameter set: LABCOM
 - › Tag: Supported Rates 1(B), 2(B), 5.5(B), 6, 9, 11(B), 12, 18, [Mbit/sec]
 - › Tag: DS Parameter set: Current Channel: 13
 - › Tag: ERP Information
 - › Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]
 - › Tag: Cisco CCX1 CKIP + Device Name
 - › Tag: Vendor Specific: Microsoft Corp.: WMM/WME: Parameter Element
 - › Tag: Vendor Specific: Cisco Systems, Inc.: Aironet Unknown (1) (1)
 - › Tag: Vendor Specific: Cisco Systems, Inc.: Aironet CCX version = 5
 - › Tag: Vendor Specific: Cisco Systems, Inc.: Aironet Unknown (11) (11)
 - › Tag: Vendor Specific: Cisco Systems, Inc.: Aironet Client MFP Disabled



Joining BSS with AP: Authentication

- Once an AP is found/selected, a station goes through authentication
- Open system authentication (default, 2-step process)
 - Station sends authentication frame with its identity
 - AP sends frame as an Ack / NAck
- Shared key authentication
 - Stations receive shared secret key through secure channel independent of 802.11
 - After the WNIC sends its initial authentication request, it will receive an authentication frame from the AP containing a challenge text
 - The WNIC sends an authentication frame containing the encrypted version of the challenge text to the AP.
 - The AP ensures the text was encrypted with the correct key by decrypting it with its own key.
 - The result of this process determines the WNIC's authentication status.



Authentication Frames

- Nowadays, WPA* secure networks use “Open System”.
- Non-“Open System” authentication was used for WEP protected networks (unsecured and functionally deprecated).

- IEEE 802.11 Authentication, Flags:

Type/Subtype: Authentication (0x000b)
Frame Control Field: 0xb000
.000 0001 0011 1010 = Duration: 314 microseconds
Receiver address: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
Destination address: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
Transmitter address: D-LinkIn_6a:cc:6e (84:c9:b2:6a:cc:6e)
Source address: D-LinkIn_6a:cc:6e (84:c9:b2:6a:cc:6e)
BSS Id: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
.... 0000 = Fragment number: 0
0001 0100 1011 = Sequence number: 331

← From Station

- IEEE 802.11 Authentication, Flags:c

Type/Subtype: Authentication (0x000b)
Frame Control Field: 0xb000
.000 0001 0011 1010 = Duration: 314 microseconds
Receiver address: D-LinkIn_6a:cc:6e (84:c9:b2:6a:cc:6e)
Destination address: D-LinkIn_6a:cc:6e (84:c9:b2:6a:cc:6e)
Transmitter address: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
Source address: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
BSS Id: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
.... 0000 = Fragment number: 0
1010 1001 0000 = Sequence number: 2704
Frame check sequence: 0x9f8350e1 [unverified]
[FCS Status: Unverified]

- IEEE 802.11 wireless LAN

- Fixed parameters (6 bytes)
Authentication Algorithm: Open System (0)
Authentication SEQ: 0x0002
Status code: Successful (0x0000)

From AP →

Joining BSS with AP: Association

- Once a station is authenticated, it starts the association process, i.e., information exchange about the AP/station capabilities and roaming
 - STA → AP: Associate Request frame
 - Enables the AP to allocate resources and synchronize. The frame carries information about the WNIC, including supported data rates and the SSID of the network the station wishes to associate with.
 - AP → STA: Association Response frame
 - Acceptance or rejection to an association request. If it is an acceptance, the frame will contain information such as association ID and supported data rates.
 - New AP informs old AP (if it is a handover).
- Only after association is completed, a station can transmit and receive data frames.



Association Request/Response Frames

- IEEE 802.11 Association Request, Flags:

Type/Subtype: Association Request (0x0000)
Frame Control Field: 0x0000
.000 0001 0011 1010 = Duration: 314 microseconds
Receiver address: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
Destination address: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
Transmitter address: D-LinkIn_6a:cc:6e (84:c9:b2:6a:cc:6e)
Source address: D-LinkIn_6a:cc:6e (84:c9:b2:6a:cc:6e)
BSS Id: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
.... 0000 = Fragment number: 0
0001 0100 1100 = Sequence number: 332

← From Station

- IEEE 802.11 wireless LAN

- Fixed parameters (4 bytes)
 - › Capabilities Information: 0x0421
 - Listen Interval: 0x000a
- Tagged parameters (43 bytes)
 - › Tag: SSID parameter set: LABCOM
 - › Tag: Supported Rates 1, 2, 5.5, 11, 6, 9, 12, 18, [Mbit/sec]
 - › Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]
 - › Tag: Extended Capabilities (8 octets)
 - › Tag: Vendor Specific: Microsoft Corp.: WMM/WME: Information E

- IEEE 802.11 Association Response, Flags:C

Type/Subtype: Association Response (0x0001)
Frame Control Field: 0x1000
.000 0001 0011 1010 = Duration: 314 microseconds
Receiver address: D-LinkIn_6a:cc:6e (84:c9:b2:6a:cc:6e)
Destination address: D-LinkIn_6a:cc:6e (84:c9:b2:6a:cc:6e)
Transmitter address: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
Source address: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
BSS Id: Cisco_61:ee:d0 (00:1c:f6:61:ee:d0)
.... 0000 = Fragment number: 0
1010 1001 0001 = Sequence number: 2705
Frame check sequence: 0xe7103b15 [unverified]
[FCS Status: Unverified]

- IEEE 802.11 wireless LAN

- Fixed parameters (6 bytes)
 - › Capabilities Information: 0x0421
 - Status code: Successful (0x0000)
 - ..00 0000 0000 0001 = Association ID: 0x0001
- Tagged parameters (42 bytes)
 - › Tag: Supported Rates 1(B), 2(B), 5.5(B), 6, 9, 11(B), 12, 18, [Mbit/sec]
 - › Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]
 - › Tag: Vendor Specific: Microsoft Corp.: WMM/WME: Parameter Element

From AP →

Data Frame

- IEEE 802.11 QoS Data, Flags: .p.....TC
 - Type/Subtype: QoS Data (0x0028)
- Frame Control Field: 0x8841
 - .000 0001 0011 1010 = Duration: 314 microseconds
 - Receiver address: Cisco_61:ee:d1 (00:1c:f6:61:ee:d1) ← Node that will receive frame (AP)
 - Transmitter address: IntelCor_e8:14:53 (b8:8a:60:e8:14:53) ← Node that send frame
 - Destination address: D-LinkIn_6a:cc:6e (84:c9:b2:6a:cc:6e) ← Station to receive data
 - Source address: IntelCor_e8:14:53 (b8:8a:60:e8:14:53) ← Station who sent data
 - BSS Id: Cisco_61:ee:d1 (00:1c:f6:61:ee:d1)
 - STA address: IntelCor_e8:14:53 (b8:8a:60:e8:14:53)
 - 0000 = Fragment number: 0
 - 0000 0000 0011 = Sequence number: 3
 - Frame check sequence: 0xc72771e8 [unverified]
 - [FCS Status: Unverified]
- Qos Control: 0x0000
- CCMP parameters
- Data (1244 bytes)
 - Data: f8002648417037bc923106ead1717d4821fde0989beb08b1...
[Length: 1244]

- Station “IntelCor*” sending data to station “D-LinkIn*” (via AP).
- Frame captured between station “IntelCor*” and AP (“Cisco*”).



Authentication and authorization mechanisms

- Changing according to the organization and the security level
 - ◆ Open network
 - ◆ Open network + MAC authentication
 - ◆ Open network + VPN-gateway
 - ◆ Open network + web-gateway
 - ◆ SSID
 - ◆ Shared key: WEP
 - ◆ Wi-Fi Protected Access (WPA)
 - ◆ IEEE 802.11i (WPA2)
 - ◆ IEEE 802.1X
 - ◆ Virtual Private Networks (VPNs)



Open Network(s)

- Open network
 - ◆ Network is open, providing IP addresses with DHCP
 - ◆ There is no authentication and access is free
 - ◆ Does not require specific software
 - ◆ Access control is complicated
 - ◆ It is possible to 'see' all traffic in the network (sniffing)
- Open network + MAC authentication
 - ◆ The control of the station MAC address is added
 - ◆ Larger management load
 - ✚ ... But MAC addresses can be falsified
 - ✚ ... Difficult to support guests
 - ✚ ... Impossible to use in public environments



Open Network + Gateways

- Open Network + VPN gateway.
 - ◆ Open network, with the client being authenticated in an IP VPN (L3) in order to be able to access its network from outside.
 - ✚ Requires VPN client software.
 - ✚ Difficult to use by guests.
 - ✚ Scalability is being enhanced.
 - ✚ VPN controllers can be expensive.
- Open network + web gateway.
 - ◆ Open network, with the client being authenticated in web server (L3), providing “credentials”.
 - ✚ Easy to use by guests.
 - ✚ Standardization is being enhanced.
 - ✚ Scalability is being enhanced.
 - ✚ A browser needs to be working during the session.



Service Set ID (SSID)

- **SSID – name of the network.**
- Identifies the BSS, emitted in the beacon.
- Networks can block beacon and force the AP to be directly specified by its name.
- This is not very efficient.
 - ◆ Operating systems are smarter.
 - ◆ The change of SSID requires a new advertisement to all stations.
 - ◆ With the increasing number of stations, security will decrease.
 - ◆ SSID is only useful to the self-organization of the stations, not to security.



WEP Protocol

- Wired Equivalent Privacy → shared key scheme.
- Part of basic 802.11 standard.
- Security protocol at link layer (L2).
- Designed to be computationally efficient and self-synchronized.
- The station has to know the key (like a password) to access the AP.
- With passive monitoring, it can be broken (in seconds)
 - Header is not ciphered, all destinations and origins are visible.
 - Control frames are not ciphered, and then they can be changed.
 - AP is not authenticated and can be falsified.
 - **Should not be implemented!**



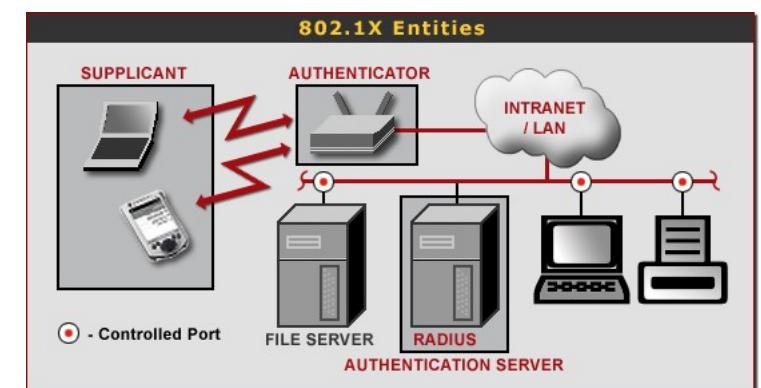
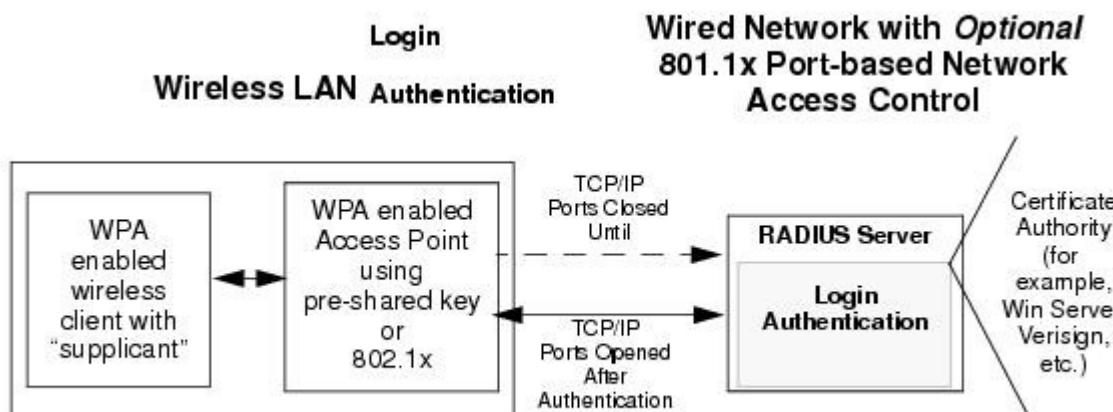
WPA and 802.11i (WPA2)

- IEEE 802.11i - IEEE 802.11 task group “MAC enhancement for wireless security”.
- Wi-Fi Protected Access (WiFi Alliance), WPA, is a subset internal in 802.11i.
 - ◆ Compatible with work developed in 802.11i.
 - ◆ Only supports BSS.
 - ◆ Defined to work in actual equipment.
 - Firmware update only.
 - ◆ Pass-phrase constant and shared, but keys are generated per session.
 - ◆ Used in the AP and station.
 - ◆ Uses “Open System” during authentication phase.
- WPA has two distinct components.
 - ◆ Authentication, based on 802.1X.
 - ◆ Ciphering based on TKIP (Temporal Key Integrity Protocol).



IEEE 802.1X

- Layer 2 solution between station and AP.
 - Available in many equipments (e.g. IEEE 802.xx).
 - Web systems frequently use 802.1X.
- Several authentication-mechanisms available (EAP-MD5, EAP-TLS, EAP-TTLS, PEAP)
- Multiple standard ciphering algorithms .
- Can cipher data with dynamic keys.
- Resorts to RADIUS servers.



WPA* Key Exchange

- Done during the Association process.

- ◆ After Association Request/response frames.

```
205 595.669409767 IntelCor_e8:14:53 Cisco_61:ee:d1      802.11  110 Association Request, SN=38, FN=0, Flags=....., SSID=LABCOM_SEC
206 595.671214291 Cisco_61:ee:d1 IntelCor_e8:14:53      802.11  128 Association Response, SN=14, FN=0, Flags=.....
207 595.673042781 Cisco_61:ee:d1 IntelCor_e8:14:53      EAPOL   211 Key (Message 1 of 4)
208 595.678333124 IntelCor_e8:14:53 Cisco_61:ee:d1      EAPOL   168 Key (Message 2 of 4)
209 595.681795313 Cisco_61:ee:d1 IntelCor_e8:14:53      EAPOL   269 Key (Message 3 of 4)
210 595.683690439 IntelCor_e8:14:53 Cisco_61:ee:d1      EAPOL   146 Key (Message 4 of 4)

Frame 207: 211 bytes on wire (1688 bits), 211 bytes captured (1688 bits) on interface 0
Radiotap Header v0, Length 56
  802.11 radio information
  IEEE 802.11 QoS Data, Flags: ....F.
    Type/Subtype: QoS Data (0x0028)
    Frame Control Field: 0x8802
      .000 0001 0011 1010 = Duration: 314 microseconds
    Receiver address: IntelCor_e8:14:53 (b8:8a:60:e8:14:53)
    Transmitter address: Cisco_61:ee:d1 (00:1c:f6:61:ee:d1)
    Destination address: IntelCor_e8:14:53 (b8:8a:60:e8:14:53)
    Source address: Cisco_61:ee:d1 (00:1c:f6:61:ee:d1)
    BSS Id: Cisco_61:ee:d1 (00:1c:f6:61:ee:d1)
    STA address: IntelCor_e8:14:53 (b8:8a:60:e8:14:53)
      .... .... 0000 = Fragment number: 0
    0000 0001 1100 .... = Sequence number: 28
  Qos Control: 0x0007
Logical-Link Control
  802.1X Authentication
    Version: 802.1X-2004 (2)
    Type: Key (3)
    Length: 117
    Key Descriptor Type: EAPOL RSN Key (2)
      [Message number: 1]
    Key Information: 0x008a
    Key Length: 16
    Replay Counter: 1
    WPA Key Nonce: 4f65d0b4e9e77b88f2ccb135749eefb105a3aa1ef65de66a...
    Key IV: 00000000000000000000000000000000
    WPA Key RSC: 0000000000000000
    WPA Key ID: 0000000000000000
    WPA Key MIC: 00000000000000000000000000000000
    WPA Key Data Length: 22
    WPA Key Data: dd14000fac046616ebb59b83e8cc1816ced0e542a935
```



Layer 3 - Routing

IPv4 and IPv6 Routing

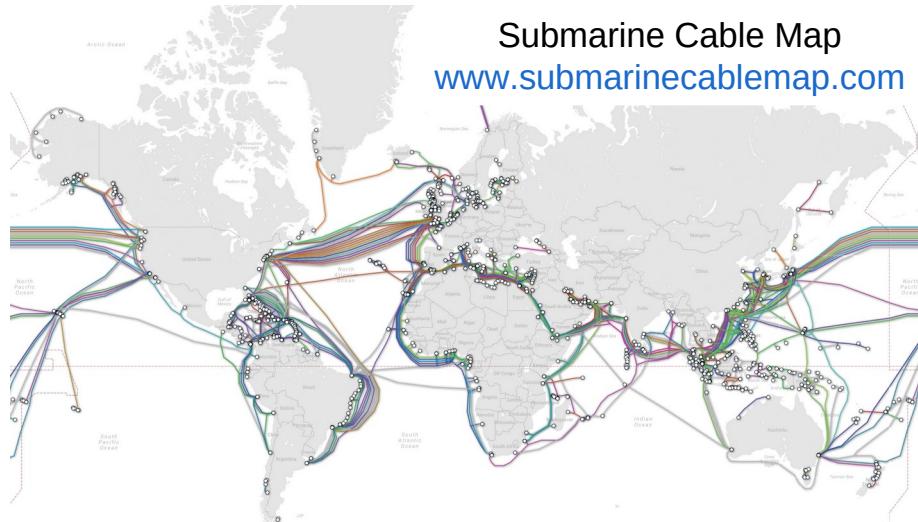
Fundamentos de Redes

Mestrado Integrado em
Engenharia de Computadores e Telemática
DETI-UA



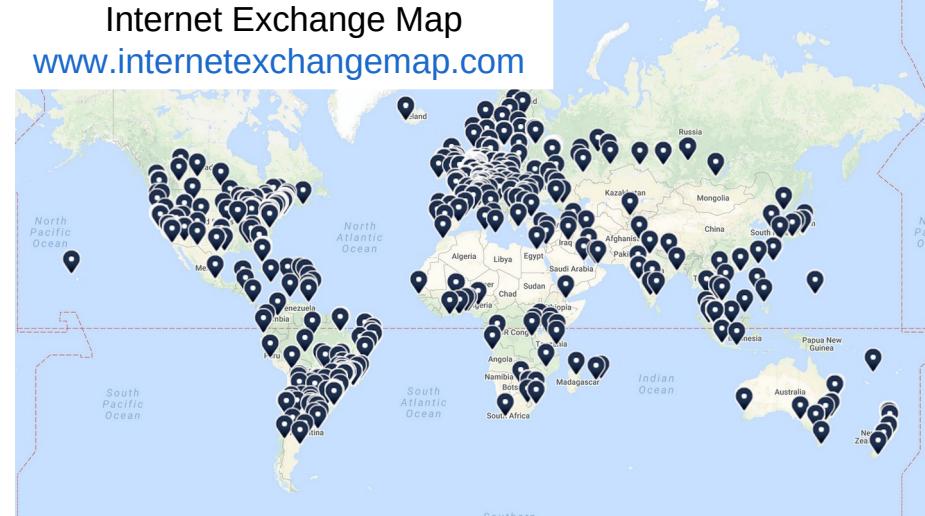
Internet Physical Structure (1)

- Submarine Cables and IXs (majority of) information is public.
 - Internet Exchange (IX): Place where ISP exchange networking information/traffic.

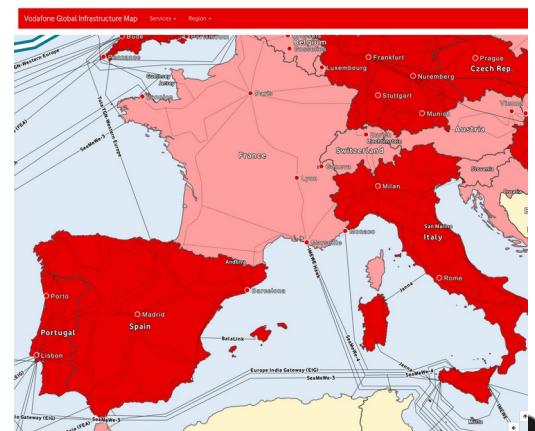
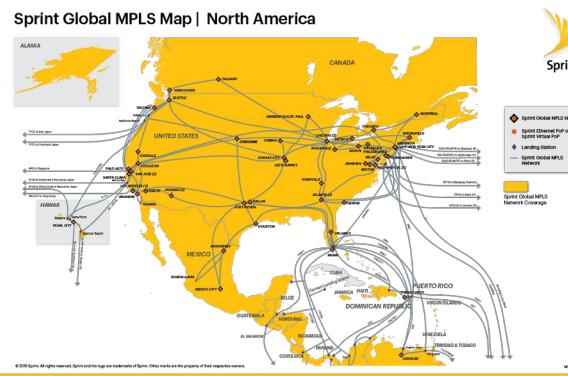
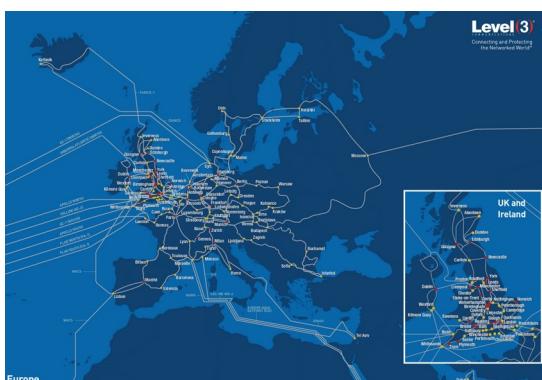


Submarine Cable Map
www.submarinecablemap.com

Internet Exchange Map
www.internetexchangemap.com

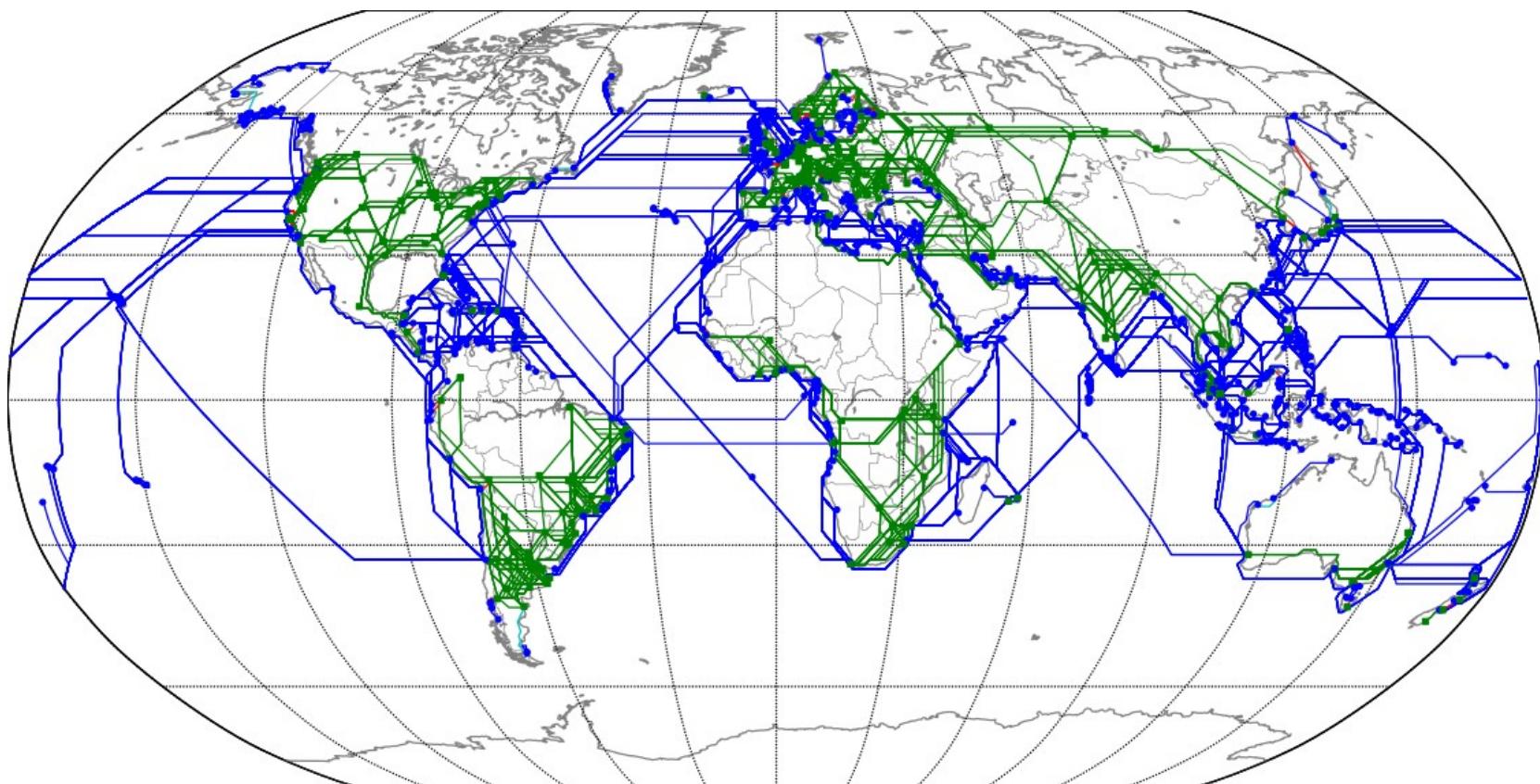


- Land Cables information is sparse (sometimes secret) and provider specific.



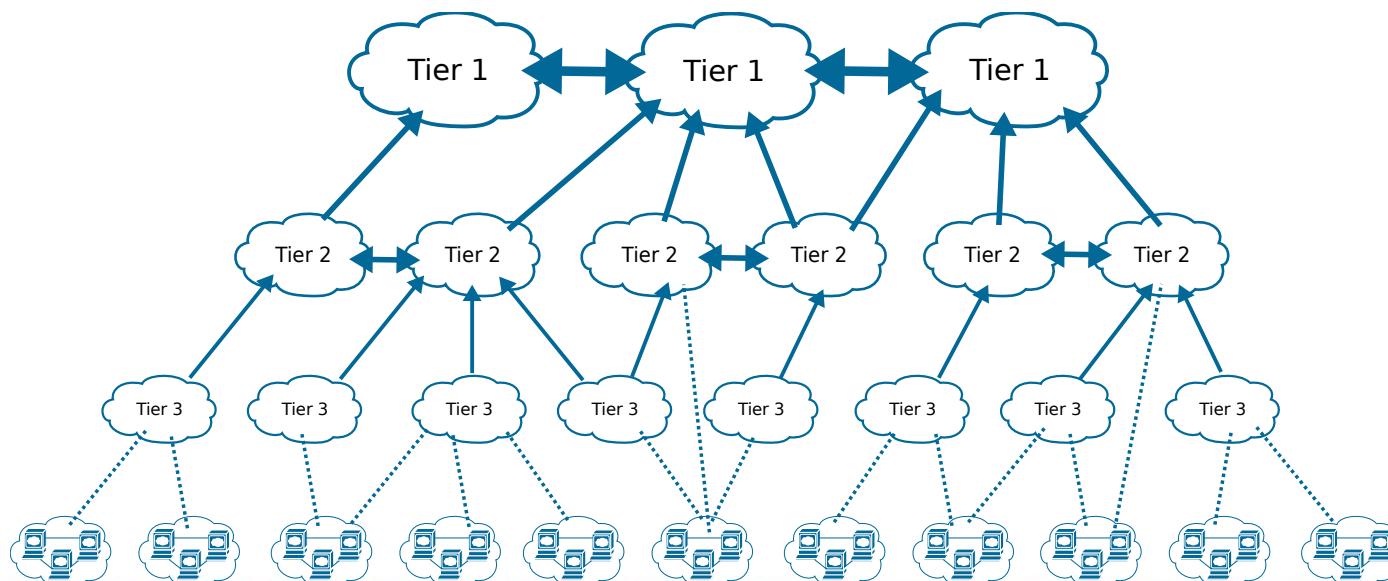
Internet Physical Structure (2)

- It is only possible to have an approximated overview of the Internet physical structure based on extrapolations of publicly available information.



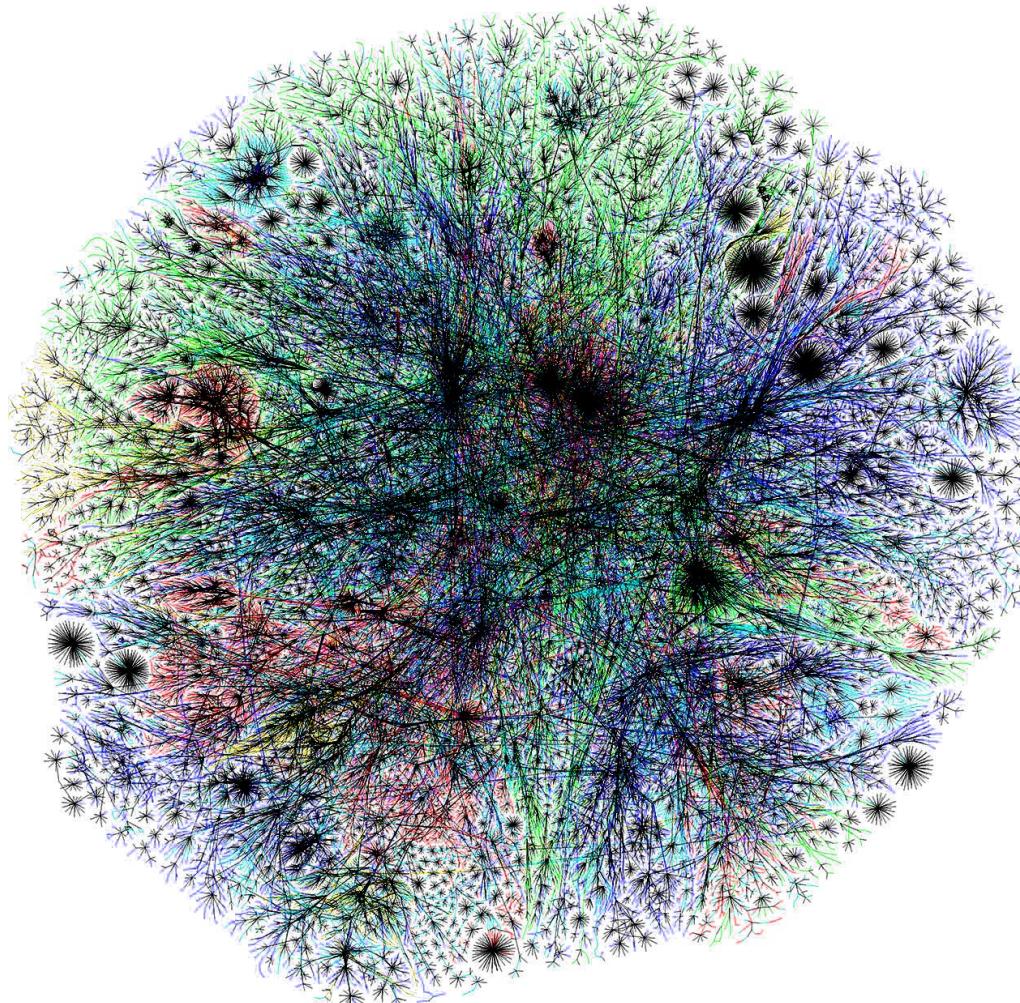
Internet Logical Structure (1)

- ISPs must agree on how there are going to exchange traffic (Peering agreement).
 - One ISP may only transport traffic from/to the other peer internal networks (non-transit), or
 - May transport traffic from/to any network the other peer sends towards it (transit).
- Tier ISP classification depends only how big the ISP is, in terms of geographical scope and how inter-operate with other ISPs.
 - Typically operate large high-capacity networks.
- Tier levels
 - Tier 1 ISPs often do not provide services to end-users, instead they provide Internet transit (transport of traffic from other ISPs networks).
 - Tier 2 ISPs also provide Internet transit at a non-global level, but require Tier1 ISPs to achieve global connectivity. May provide services to end-users.
 - Tier 3 operate at a regional level, provide services to end-users, and require Tier1 and Tier2 ISPs to achieve global connectivity.



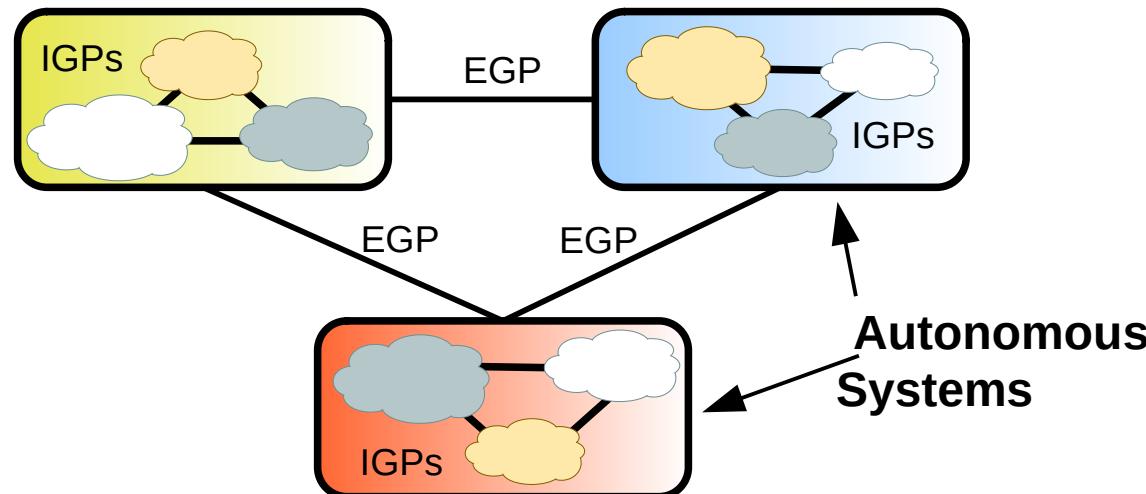
Internet Logical Structure (2)

- Internet complexity:



Autonomous Systems

- AS (Autonomous System) – set of routers/networks with a common routing policy and under the same administration.
- Routing inside an AS is performed by IGPs (Interior Gateway Protocols) such as RIPv1, RIPv2, RIPng, OSPFv2, OSPFv3, IS-IS and EIGRP.
 - ◆ Called Internal Routing
- Routing between AS is performed by EGPs (Exterior Gateway Protocols) such as BGP and MP-BGP.
- IGPs and EGPs have different objectives:
 - ◆ IGPs: optimize routing performance
 - ◆ EGPs: optimize routing performance obeying political, economic and security policies.

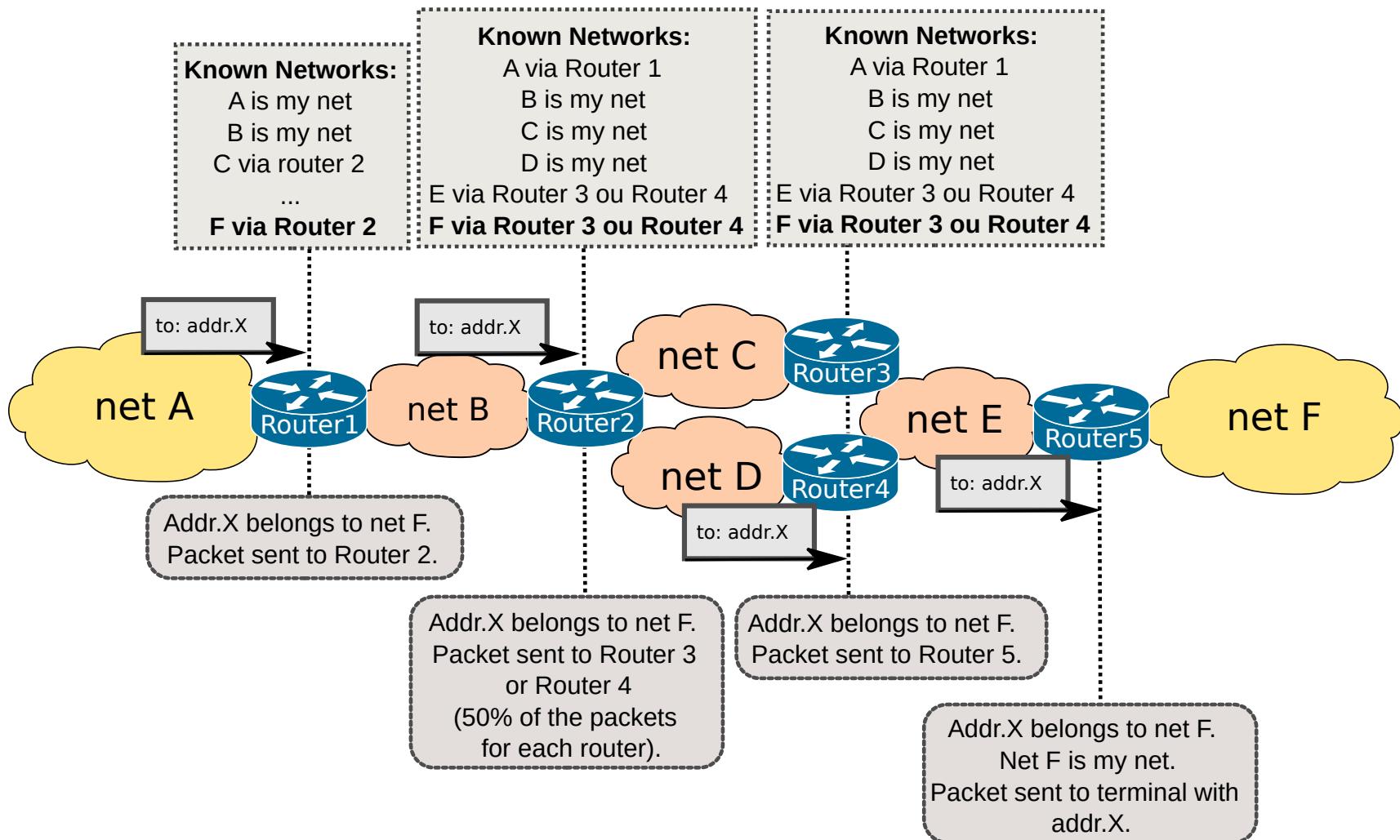


IP Routing Overview (1)

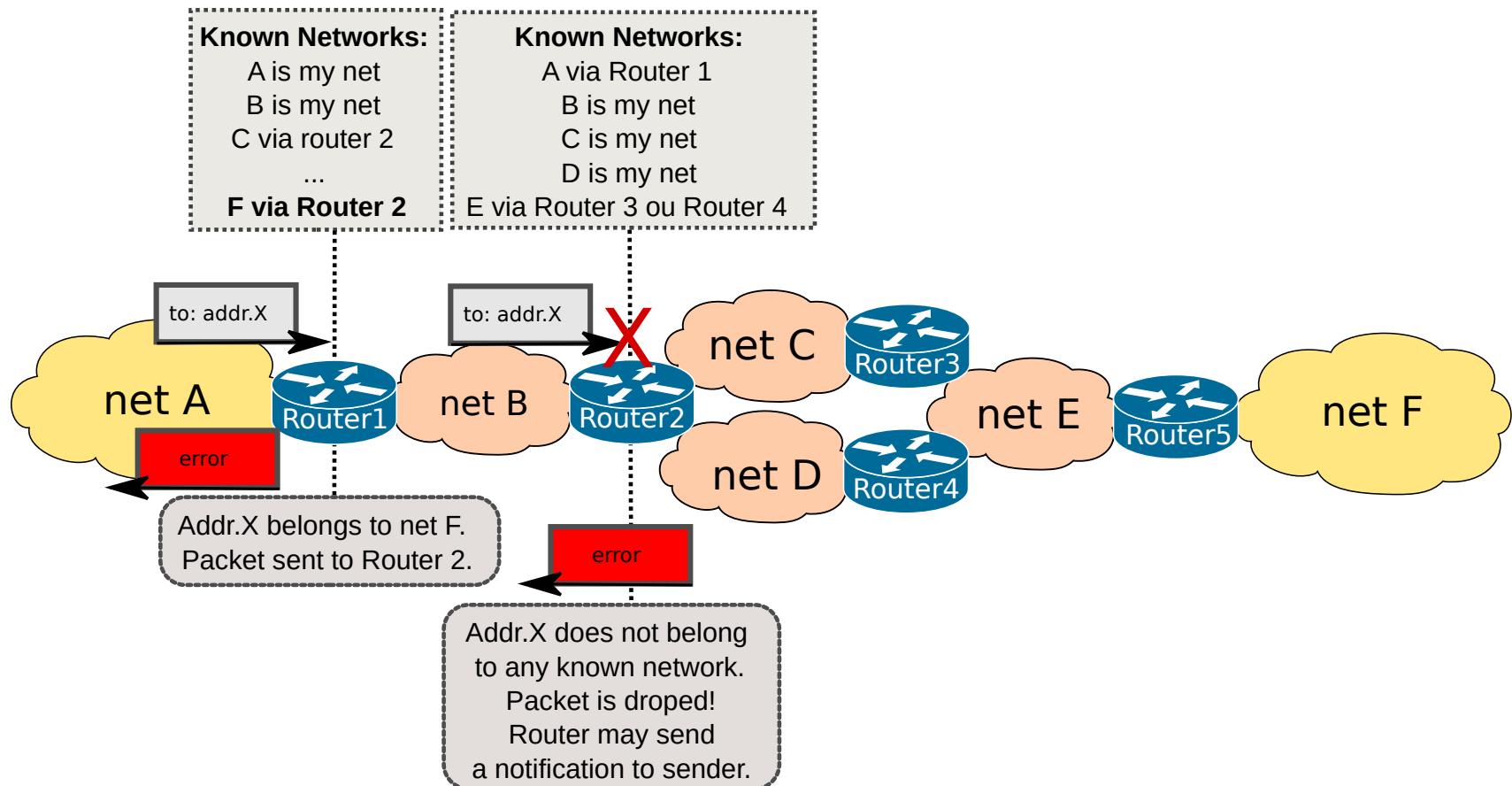
- Routers forward packets toward destination networks.
- Routers must be aware of destination networks to be able to forward packets to them.
- A router knows about the networks directly attached to its interfaces
- For networks not directly connected to one of its interfaces, however, the router must rely on outside information.
- A router can be made aware of remote networks by:
 - ◆ **Static routing:** An administrator manually configure the information.
 - ◆ **Dynamic routing:** Learns from other routers.
 - ◆ **Routing policies:** Manually routing rules that outweigh static/dynamic routing.
- If a packet for un unknown network reaches the router this will drop the packet, and MAY notify the sender about the routing error.



IP Routing Overview (2)



IP Routing Overview (3)



Routing Tables (1)

Cisco IOS

- Define how a remote network is reachable:

- Next-hop (identified by its address), and
- Local interface that provides connection.

- A network may be reachable using more than one path: (next-hop,local interface) pair.

- Mandatory elements

- Destination prefix
- Destination mask
- Metric
 - Could be defined by key tags.
 - e.g., Directly Connected

- One or both
 - Next-hop address
 - Output interface

- Optional elements

- Administrative distance
- Protocol
- Entry age (last time information received)
- Scope
- Flags
- Source-specific

- The next path hop (next hop address) may be found using more than one table entry (recursive resolution).

- e.g., Network A is reachable through address from network B, Network B is reachable through address from network C, ...

- The next-hop address may be obtained from external information (configurations or other mechanisms).

- e.g., Tunnels, Point-to-point connections, etc...

- When an entry uses a next-hop address from an unknown network, that entry is removed.

- All entries obtain by dynamic methods may have an entry age (time since last update/confirmation).

- After a timeout value without an update/confirmation the entry is removed.

| | |
|---|--|
| R | 200.1.1.0/24 [120/1] via 200.19.14.10, 00:00:16, FastEthernet0/1 |
| | 200.19.14.0/24 is variably subnetted, 2 subnets, 2 masks |
| C | 200.19.14.0/24 is directly connected, FastEthernet0/1 |
| L | 200.19.14.4/32 is directly connected, FastEthernet0/1 |
| R | 200.38.0.0/24 [120/1] via 200.43.0.8, 00:00:03, FastEthernet1/1 |
| | 200.43.0.0/24 is variably subnetted, 2 subnets, 2 masks |
| C | 200.43.0.0/24 is directly connected, FastEthernet1/1 |
| L | 200.43.0.1/32 is directly connected, FastEthernet1/1 |

Linux: route -n

| Destination | Gateway | Genmask | Flags | Metric | Ref | Use | Iface |
|--------------|--------------|---------------|-------|--------|-----|-----|----------|
| 0.0.0.0 | 193.136.92.1 | 0.0.0.0 | UG | 100 | 0 | 0 | enp5s0f1 |
| 169.254.0.0 | 0.0.0.0 | 255.255.0.0 | U | 1000 | 0 | 0 | enp5s0f1 |
| 193.136.92.0 | 0.0.0.0 | 255.255.254.0 | U | 100 | 0 | 0 | enp5s0f1 |

Linux: ip route

```
default via 193.136.92.1 dev enp5s0f1 proto static metric 100
169.254.0.0/16 dev enp5s0f1 scope link metric 1000
193.136.92.0/23 dev enp5s0f1 proto kernel scope link src 193.136.93.104 metric 100
```

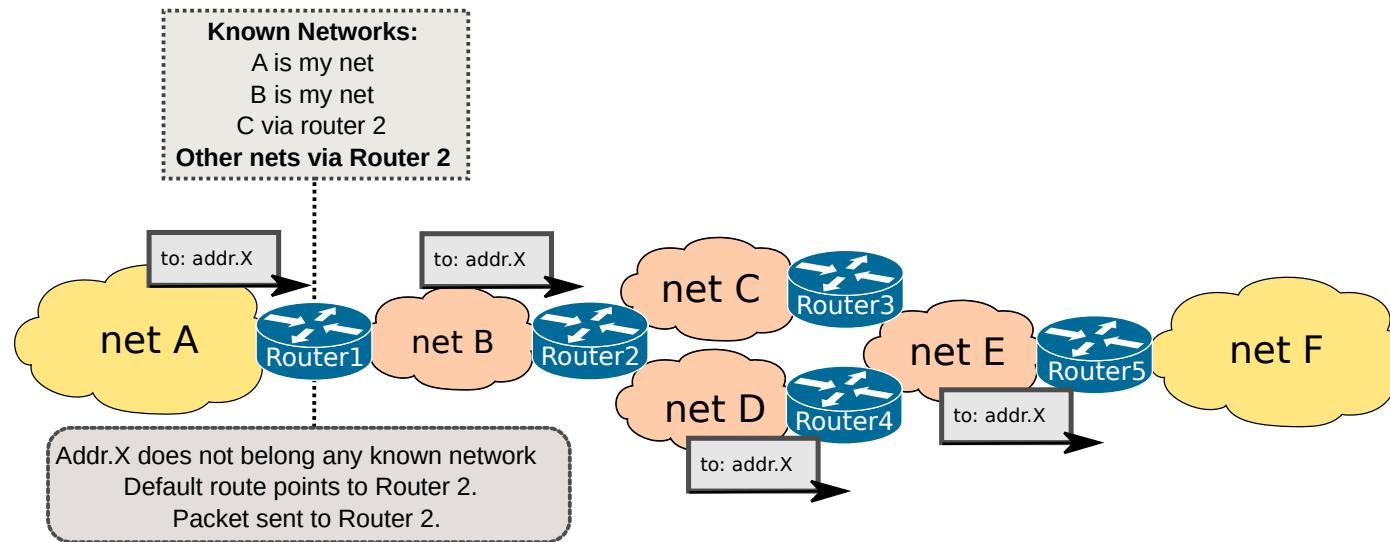


Routing Tables (2)

- An IP address may have multiple matches on a Routing Table:
 - ◆ Example: 192.168.1.12
 - ◆ Will match:
 - 192.168.1.0/25 via ...
 - 192.168.1.0/24 via ...
 - 192.168.0.0/23 via ...
 - 192.168.0.0/16 via ...
 - ...
 - ◆ Router will choose entry with the largest network prefix (most specific network).
 - i.e., 192.168.1.0/25 via ...
- Load balancing
 - ◆ Routing tables may have more than one path for each network
 - ◆ Traffic will be divided by all entries.
 - ◆ By packet, flow (TCP session, UDP IPs/port), etc...
 - E.g, packet 1 path 1, packet 2 path 2, packet 3 path 1, ...
 - Flow 1 path 1, flow 2 path 2, flow 3 path 3, flow 4 path 1, flow 5 path 2, ...



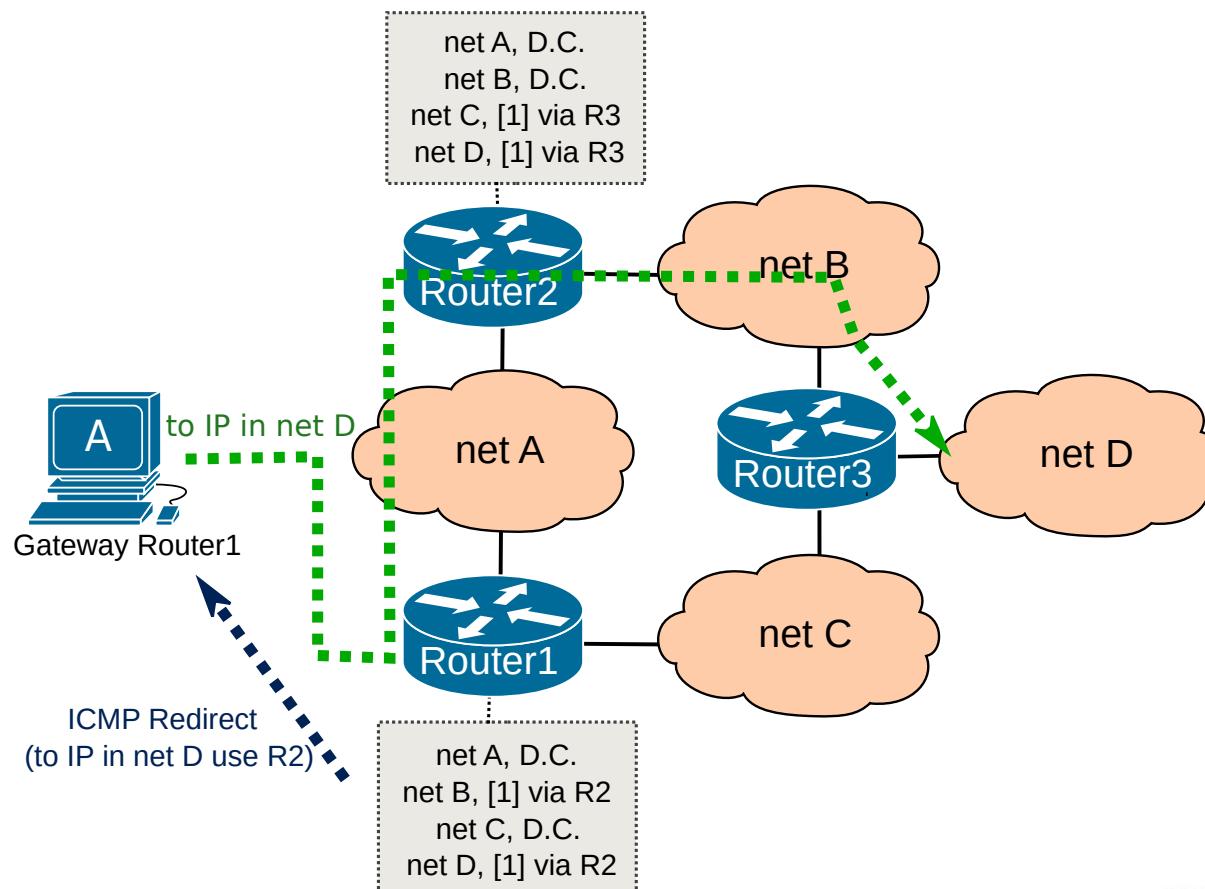
Default Routes



- In some circumstances, a router does not need to recognize the details of remote networks.
- The router can be configured to send all traffic (or all traffic for which there is not a more specific entry in the routing table) to a specific neighbor router.
- This is known as a default route.
- Default routes are either dynamically advertised using routing protocols or statically configured.
- IPv4 default route - $0.0.0.0/0$
- IPv6 default route - $::/0$

ICMP Redirect

- When a Router has a path defined via the interface from which received a packet:
 - The router sends an **ICMP Redirect** to the sender, defining a new gateway for that specific destination (IP address),
 - The sender updates its routing table for that specific destination address.



Administrative Distance

- Most routing protocols have metric structures and algorithms that are incompatible with other protocols.
- It is critical that a network using multiple routing protocols be able to seamlessly exchange route information and be able to select the best path across multiple protocols.
- Routers use a value called administrative distance to select the best path when they learn from different routing protocols the same destination (same network prefix and mask length).
- The Protocol/Method with the lowest Administrative Distance is preferred
 - ◆ The Administrative Distance value is configurable.
- Example:
 - ◆ Static [1/0] 192.168.1.0/24 via ... ← Chosen!
 - ◆ RIP [120/1] 192.168.1.0/24 via ...
 - ◆ OSPF [110/1] 192.168.1.0/24 via ...

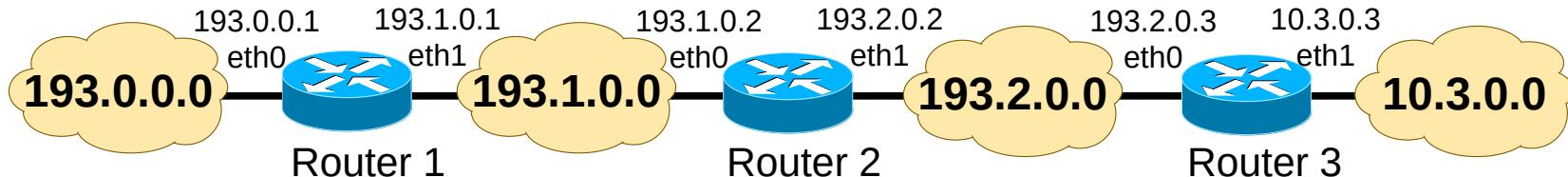


Static Routing

- Stating routing do not react to network topology changes.
 - ◆ If a link fails, the static route is no longer valid if it is configured to use that failed link, so a new static route must be configured.
 - ◆ Connectivity may be lost until intervention of an administrator.
- Static routing does not scale well when network grows.
 - ◆ Administrative burden to maintain routes may become excessive.
- Static routes can be used in the following circumstances:
 - ◆ When the administrator needs total control over the routes used by the router.
 - ◆ When a backup to a dynamically recognized route is necessary.
 - ◆ When it is used to reach a network accessible by only one path (a stub network).
 - ◆ There is no backup link, so dynamic routing has no advantage.
 - ◆ When a router connects to its ISP and needs to have only a default route pointing toward the ISP router, rather than learning many routes from the ISP.
 - ◆ Again, a single path of access without backup.
 - ◆ When a router is underpowered and does not have the CPU or memory resources necessary to handle a dynamic routing protocol.
 - ◆ When it is undesirable to have dynamic routing updates forwarded across low bandwidth links.



Static Routing Examples



- Example 1

- Router2 do not know networks 193.0.0.0/24 and 10.3.0.0/24
- Necessary static routes:
 - 193.0.0.0/24 accessible through 193.1.0.1 (eth1, Router1)
 - 10.3.0.0/24 accessible through 193.2.0.3 (eth0, Router3)

- Example 2

- Router1 do not know networks 193.2.0.0/24 and 10.3.0.0/24
- Necessary static routes:
 - 193.2.0.0/24 accessible through 193.1.0.2 (eth0, Router2)
 - 10.3.0.0/24 accessible through 193.1.0.2 (eth0, Router2)
- OR
- Using default route: 0.0.0.0/0 accessible through 193.1.0.2 (eth0, Router2)

Dynamic Routing

- Dynamic routing allows the network to adjust to changes in the topology automatically, without administrator involvement.
- Routers exchange information about the reachable networks and the state of each network/link.
 - ◆ Routers exchange information only with other routers running the same routing protocol.
 - ◆ When the network topology changes, the new information is dynamically propagated throughout the network, and each router updates its routing table to reflect the changes.



Distance Vector vs. Link State

- *Distance vector*

- Each router knows only the distance/cost to all network destinations.
- Information (distances to all known destinations) is periodically sent by routers to its neighbors
- Each router determines the lowest cost paths to all destinations based on a distributed and asynchronous version of the Bellman-Ford algorithm
- Examples: RIP, IGRP, EIGRP

- *Link state*

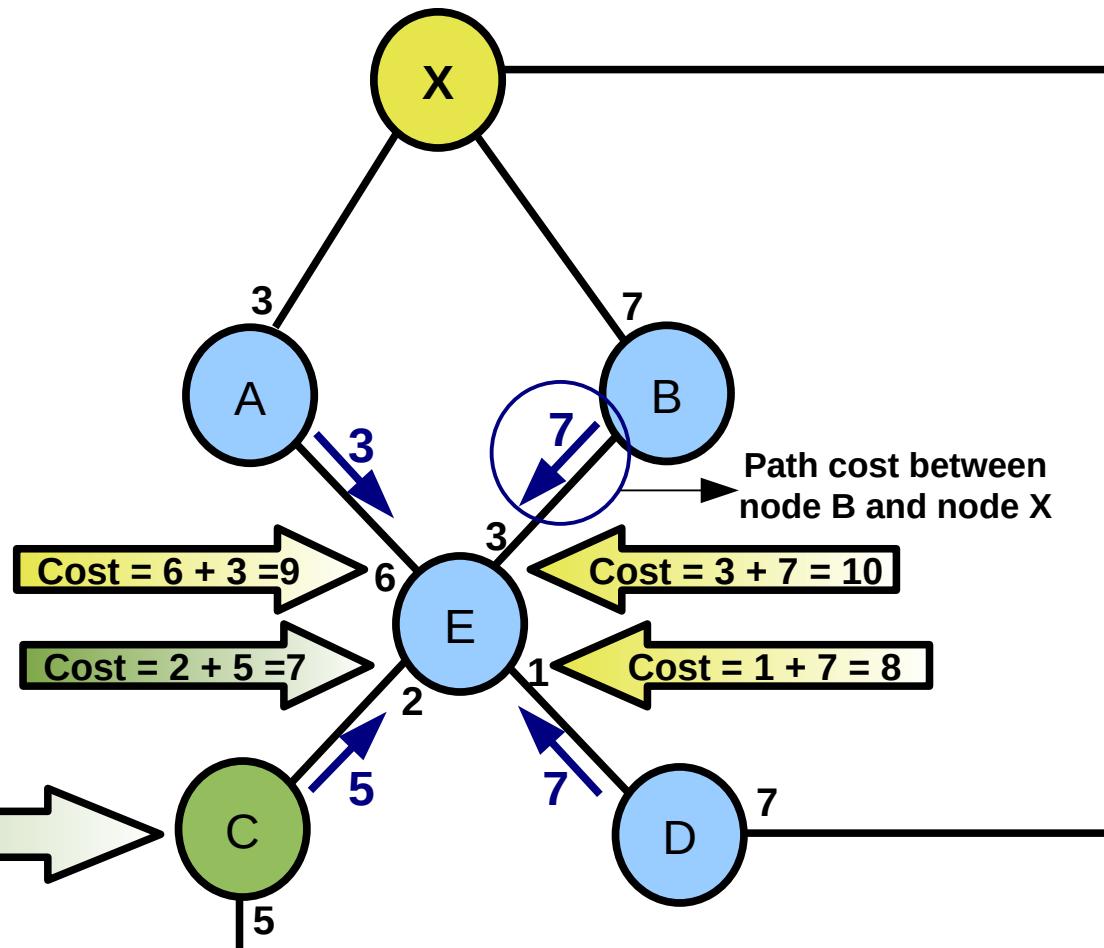
- Routers know the complete network topology.
- Use a centralized algorithm to determine the lowest cost path to all destinations.
- The required information to build and maintain the network topology in each
- Examples: OSPF, IS-IS



Distributed and Asynchronous Bellman-Ford Algorithm

- Each node periodically transmits to its neighboring nodes (its estimation of) the cost to reach a destination node.
- Each node recalculates its own estimation of the cost to reach a destination node
 - ◆ Adds the received estimated cost to the destination to the cost of the connection/port where it received the neighbor information.
 - ◆ Chooses the lowest cost.

Neighbor chosen by node E to route traffic to node X



RIP (Routing Information Protocol)

- Is a *distance vector* protocol
 - ◆ Each router maintains a list of known networks and, for each network, an estimation of the cost to reach it – this is called a distance vector.
 - ◆ Each router periodically send to its neighboring routers its own distance vector (partially or complete) – announcement/update.
 - ◆ Each router uses the distance vector sent by its neighbors to update its own distance vector.
- The path cost to a destination is given by the number of routers/hops in the path.
 - ◆ Maximum cost is 15.
 - ◆ A cost of 16 is considered infinite (or unattainable destination).
- Each router determines the entries in its own routing table, based on the constructed distance vector.
 - ◆ For each destination (network) learned, it adds an entry to that network that uses the path (or paths) with the lowest cost, using as next-hop the neighboring router(s) that announced that network with that lowest cost path.



RIP Version 1

- RIP Version 1 (RIPv1) is a classfull protocol.
 - ◆ Does not announces (sub-)networks masks, only network prefixes.
 - ◆ Network masks are assumed based on the incoming interface mask.
 - ◆ If all networks have the same mask it works perfectly, however, when networks with different masks exist it is problematic.
- RIPv1 uses the broadcast address 255.255.255.255 to send announcements/updates.
 - ◆ All network devices must process the packets.
- Does not support authentication.
 - ◆ Messages may be forged by an attacker.



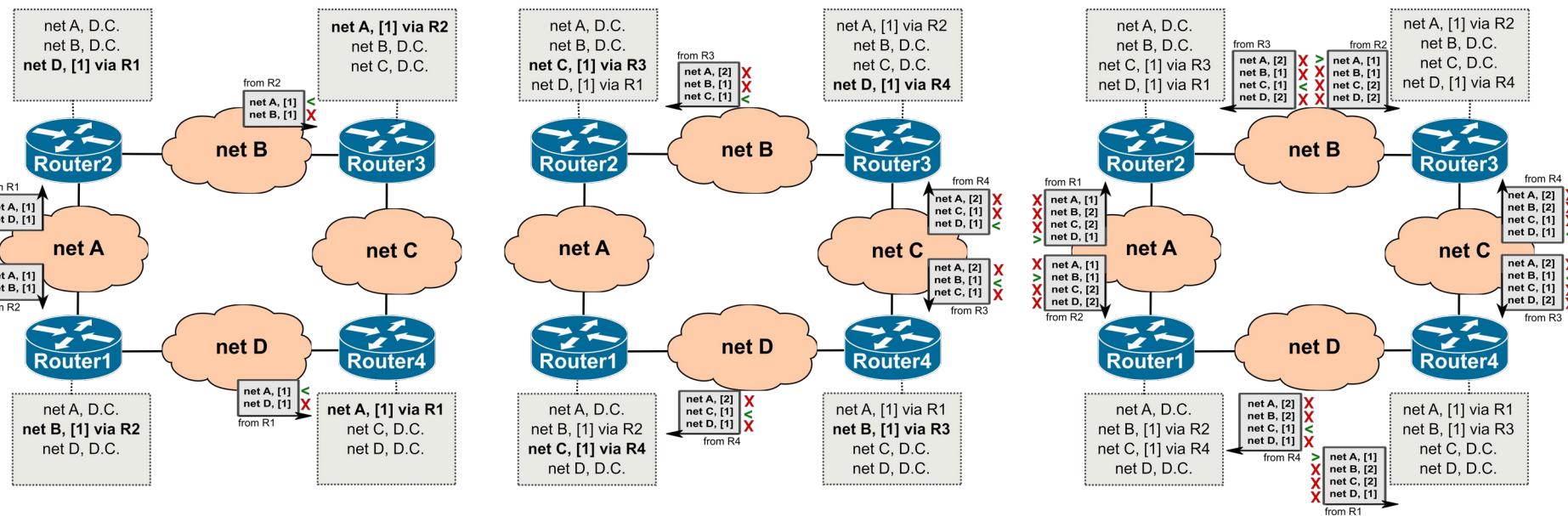
RIP Version 2

- RIP Version 2 (RIPv2) is a *classless* protocol.
 - ◆ RIPv2 announcements include network prefix and mask.
 - ◆ Supports variable length masks.
- RIPv2 used the multicast address 224.0.0.9 to send announcements/updates only to routers running RIPv2.
- RIPv2 supports authentication using message-digest and clear text password.
 - ◆ Clear text password authentication should not be used!



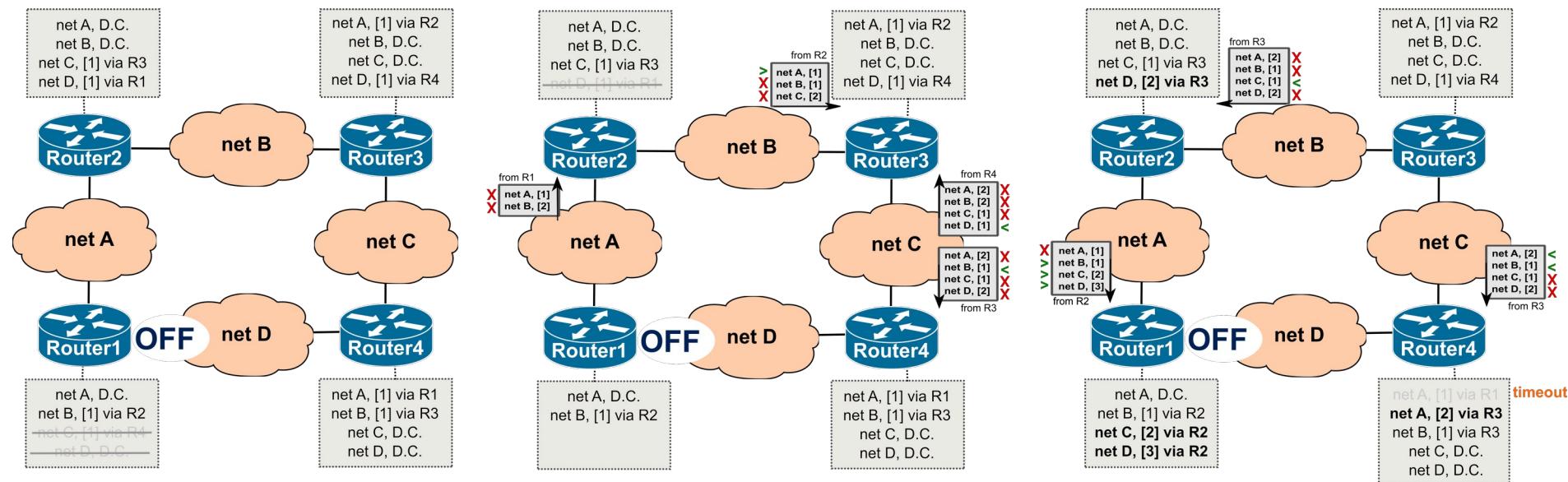
RIP Algorithm (1)

- Assuming that Router1 and Router2 send announcements first.
 - With split-horizon disabled.



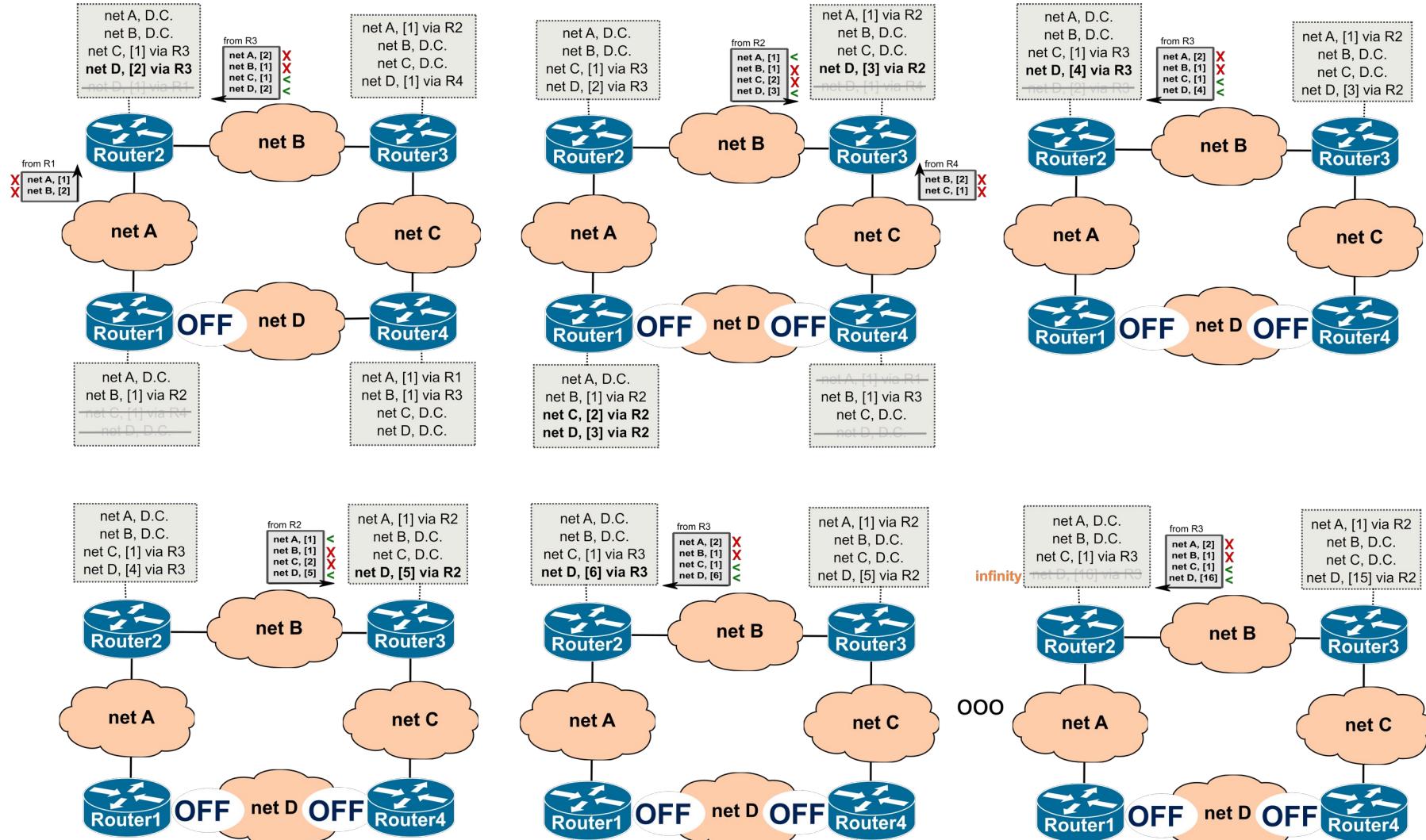
RIP Algorithm (2)

- Assuming Router1 connection to network D goes down.
 - No triggered updates.



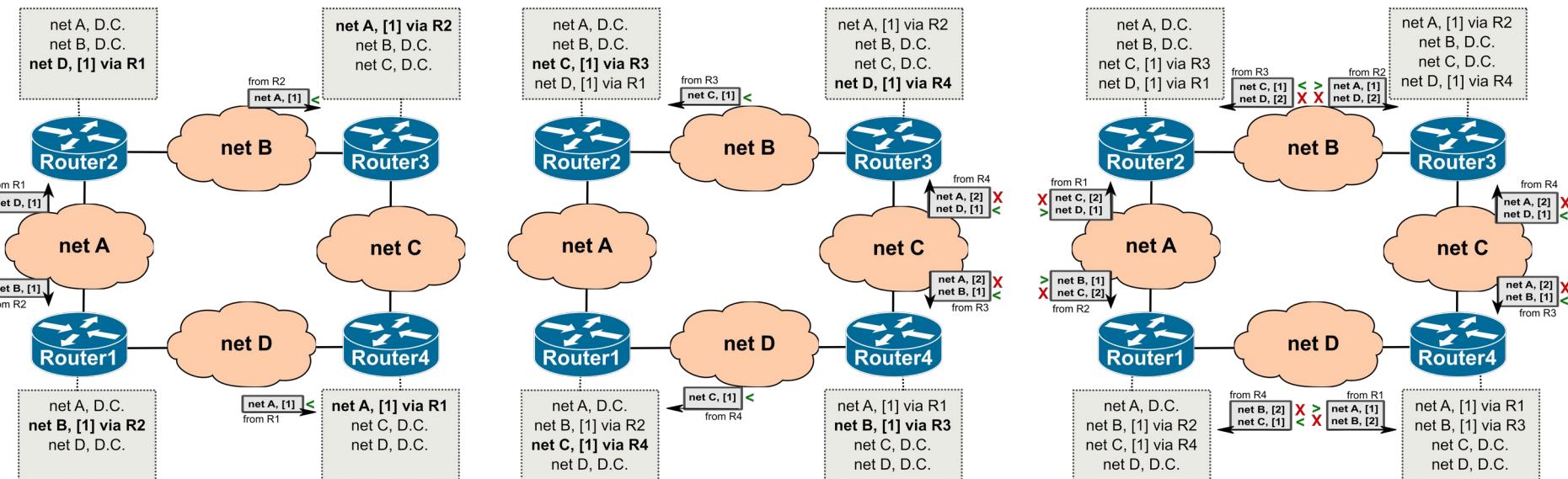
Count to Infinity Problem

- When multiple failures occur before algorithm convergence!



Split-Horizon (1)

- Solution for the count to infinity problem.
- Each Router, in each interface, announces only the networks in which that interface is not used to provide the best path to that destination.
- Split horizon lowers the convergence time of the routing tables when there is a topology change.
 - ◆ RIPv1 e RIPv2 supports it.
- Assuming Router1 and Router2 start sending announcements first:

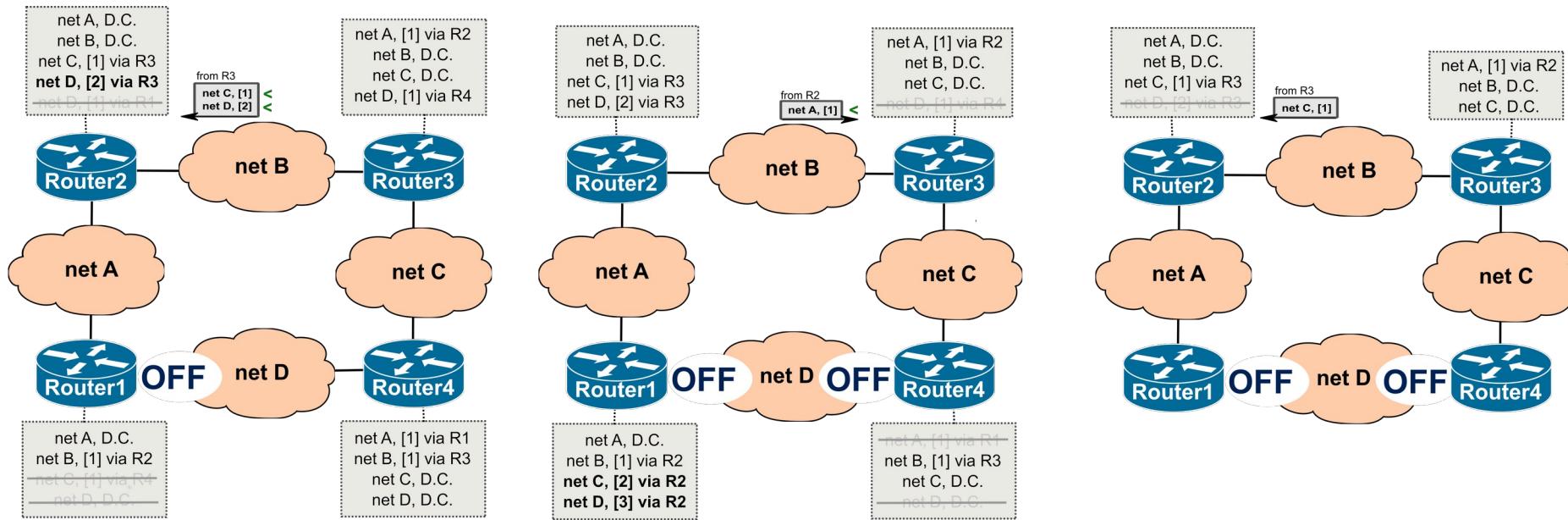


- In Split horizon with Poisoned Reverse, routers announce all networks but set metric to infinity (16) for networks learned by the interface by which they are sending the announcement.
 - ◆ Larger update messages.



Split-Horizon (2)

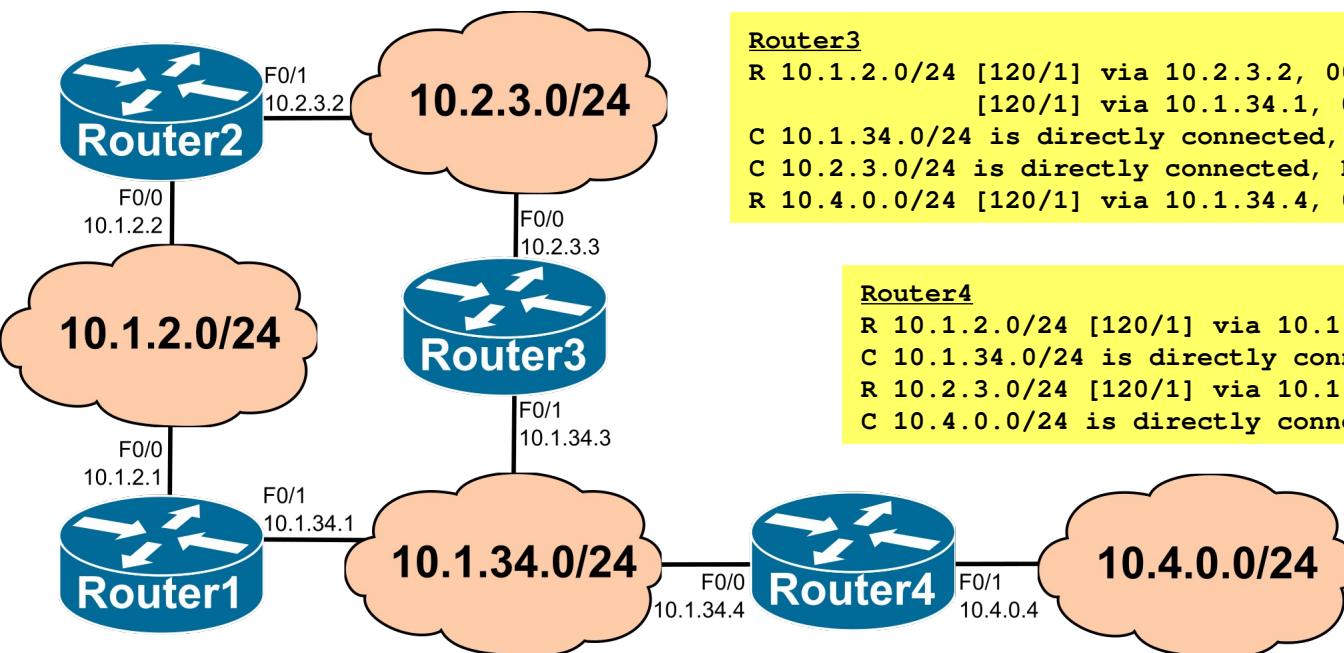
- Solution for the count to infinity problem.
- Prevents any routing loops that involve two routers.
 - ◆ It is possible to end up with patterns in which three or more routers are engaged in mutual deception.
- Assuming Router1 and Router4 loose connection to network D almost simultaneously:



Routing Tables with RIP

Router2

```
C 10.1.2.0/24 is directly connected, FastEthernet0/0
R 10.1.34.0/24 [120/1] via 10.2.3.3, 00:00:21, FastEthernet0/1
          [120/1] via 10.1.2.1, 00:00:11, FastEthernet0/0
C 10.2.3.0/24 is directly connected, FastEthernet0/1
R 10.4.0.0/24 [120/2] via 10.2.3.3, 00:00:21, FastEthernet0/1
          [120/2] via 10.1.2.1, 00:00:11, FastEthernet0/0
```



Router3

```
R 10.1.2.0/24 [120/1] via 10.2.3.2, 00:00:08, FastEthernet0/0
          [120/1] via 10.1.34.1, 00:00:28, FastEthernet0/1
C 10.1.34.0/24 is directly connected, FastEthernet0/1
C 10.2.3.0/24 is directly connected, FastEthernet0/0
R 10.4.0.0/24 [120/1] via 10.1.34.4, 00:00:24, FastEthernet0/1
```

Router4

```
R 10.1.2.0/24 [120/1] via 10.1.34.1, 00:00:18, FastEthernet0/0
C 10.1.34.0/24 is directly connected, FastEthernet0/0
R 10.2.3.0/24 [120/1] via 10.1.34.3, 00:00:29, FastEthernet0/0
C 10.4.0.0/24 is directly connected, FastEthernet0/1
```

Router1

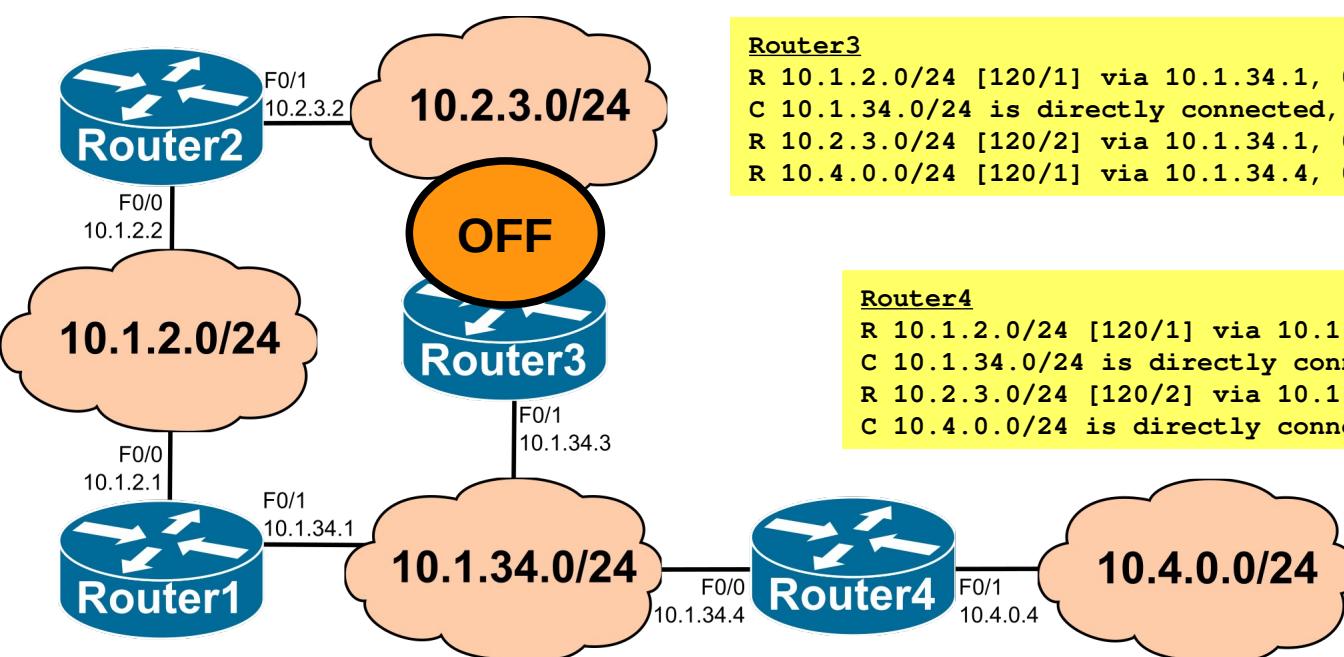
```
C 10.1.2.0/24 is directly connected, FastEthernet0/0
C 10.1.34.0/24 is directly connected, FastEthernet0/1
R 10.2.3.0/24 [120/1] via 10.1.34.3, 00:00:11, FastEthernet0/1
          [120/1] via 10.1.2.2, 00:00:01, FastEthernet0/0
R 10.4.0.0/24 [120/1] via 10.1.34.4, 00:00:24, FastEthernet0/1
```



Routing Tables with RIP

Router2 (AFTER 3 minutes TIMEOUT)

```
C 10.1.2.0/24 is directly connected, FastEthernet0/0
R 10.1.34.0/24 [120/1] via 10.1.2.1, 00:00:25, FastEthernet0/0
C 10.2.3.0/24 is directly connected, FastEthernet0/1
R 10.4.0.0/24 [120/2] via 10.1.2.1, 00:00:25, FastEthernet0/0
```



Router3

```
R 10.1.2.0/24 [120/1] via 10.1.34.1, 00:00:22, FastEthernet0/1
C 10.1.34.0/24 is directly connected, FastEthernet0/0
R 10.2.3.0/24 [120/2] via 10.1.34.1, 00:00:22, FastEthernet0/1
R 10.4.0.0/24 [120/1] via 10.1.34.4, 00:00:19, FastEthernet0/11
```

Router4

```
R 10.1.2.0/24 [120/1] via 10.1.34.1, 00:00:18, FastEthernet0/0
C 10.1.34.0/24 is directly connected, FastEthernet0/0
R 10.2.3.0/24 [120/2] via 10.1.34.1, 00:00:29, FastEthernet0/0
C 10.4.0.0/24 is directly connected, FastEthernet0/11
```

Router1

```
C 10.1.2.0/24 is directly connected, FastEthernet0/0
C 10.1.34.0/24 is directly connected, FastEthernet0/1
R 10.2.3.0/24 [120/1] via 10.1.2.2, 00:00:01, FastEthernet0/0
R 10.4.0.0/24 [120/1] via 10.1.34.4, 00:00:24, FastEthernet0/1
```



RIP Message Types

- RIP Response

- ◆ Distance vector announcement/update message.
 - ➡ Contains the distance vector.
- ◆ It is sent:
 - ➡ 1 – Periodically (~30 seconds by default, there is a random component).
 - ➡ 2 – Optionally, when some information changes (triggered updates).
 - ➡ 3 – In response to a RIP Request.
- ➡ In cases 1 and 2:
 - In RIPv1, is sent to the broadcast address.
 - In RIPv2, is sent to the multicast address 224.0.0.9 (Routers com RIP).
- ➡ In case 3, it is sent only (unicast) to the router that sent the RIP Request.

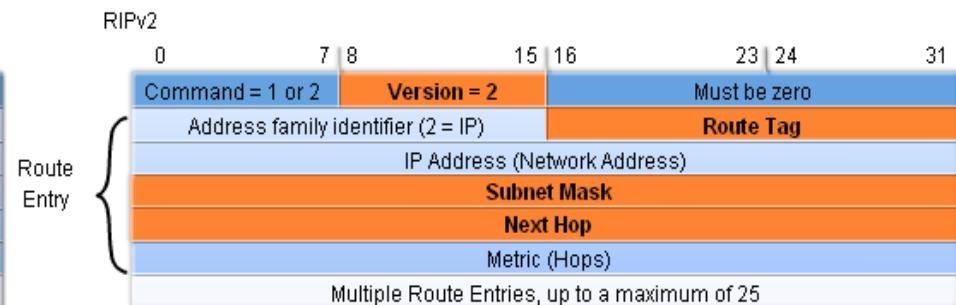
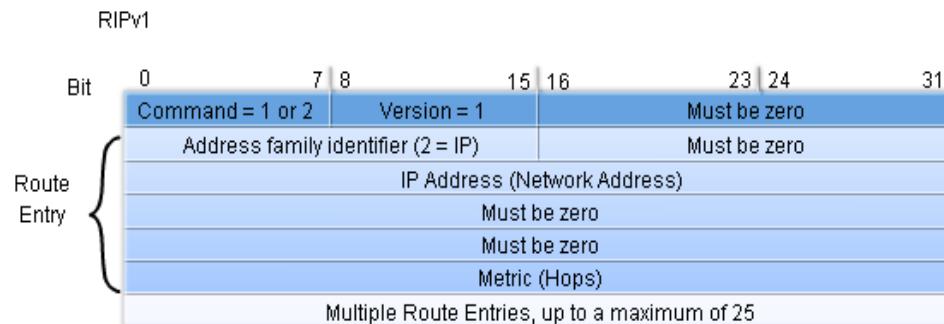
- RIP Request (Optional)

- ◆ Sent by a router that was recently started (bootstrap) or, when the validity of some of the distance vector information has expired (default timeout = 180 seconds)
- ◆ It may request specific information (a specific network) or, the complete neighbor distance vector.



RIPv1 vs. RIPv2 Responses (1)

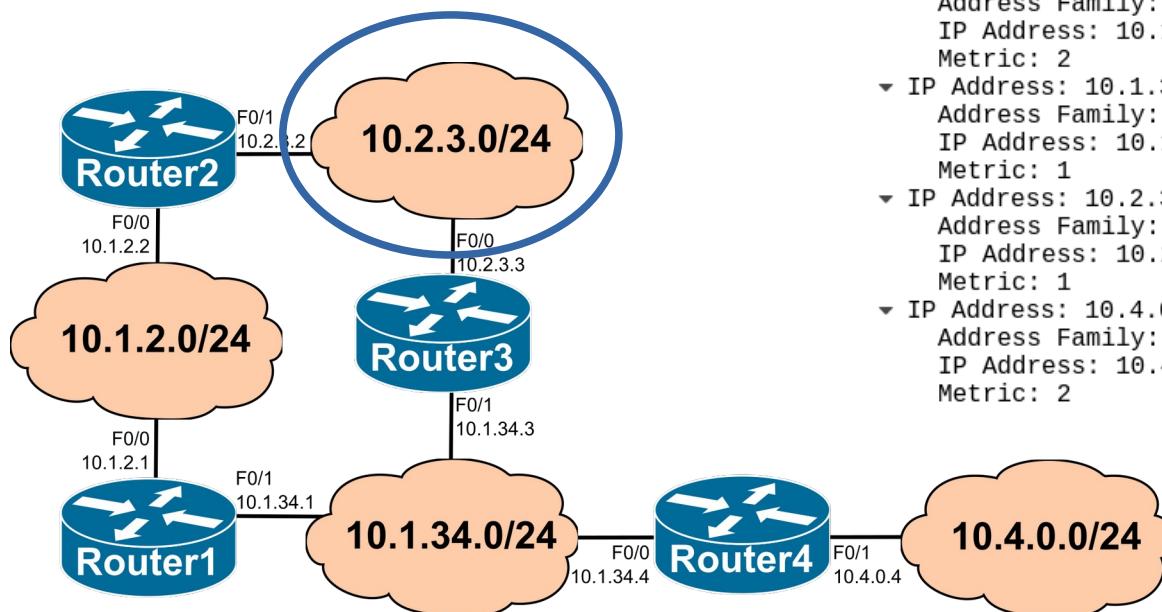
- New RIPv2 message fields in Response packets:
 - ◆ Subnet mask
 - Supports variable length masks.
 - Makes RIPv2 *classless* protocol.
 - ◆ Route tag
 - Attribute assigned to a specific network that must be reserved a re-announced.
 - Provides a method to separate internal (to the RIP domain) and external networks.
 - ◆ Next hop
 - Address to which the packets must be routed.
 - 0.0.0.0 indicates that the packets must be routed to the router that sent the RIP message.



RIPv1 Messages (Example)

Sent by Router3 with Split-Horizon

- ▶ Internet Protocol Version 4, Src: 10.2.3.3, Dst: 255.255.255.255
- ▶ User Datagram Protocol, Src Port: 520, Dst Port: 520
- ▼ Routing Information Protocol
 - Command: Response (2)
 - Version: RIPv1 (1)
 - ▼ IP Address: 10.1.34.0, Metric: 1
 - Address Family: IP (2)
 - IP Address: 10.1.34.0
 - Metric: 1
 - ▼ IP Address: 10.4.0.0, Metric: 2
 - Address Family: IP (2)
 - IP Address: 10.4.0.0
 - Metric: 2



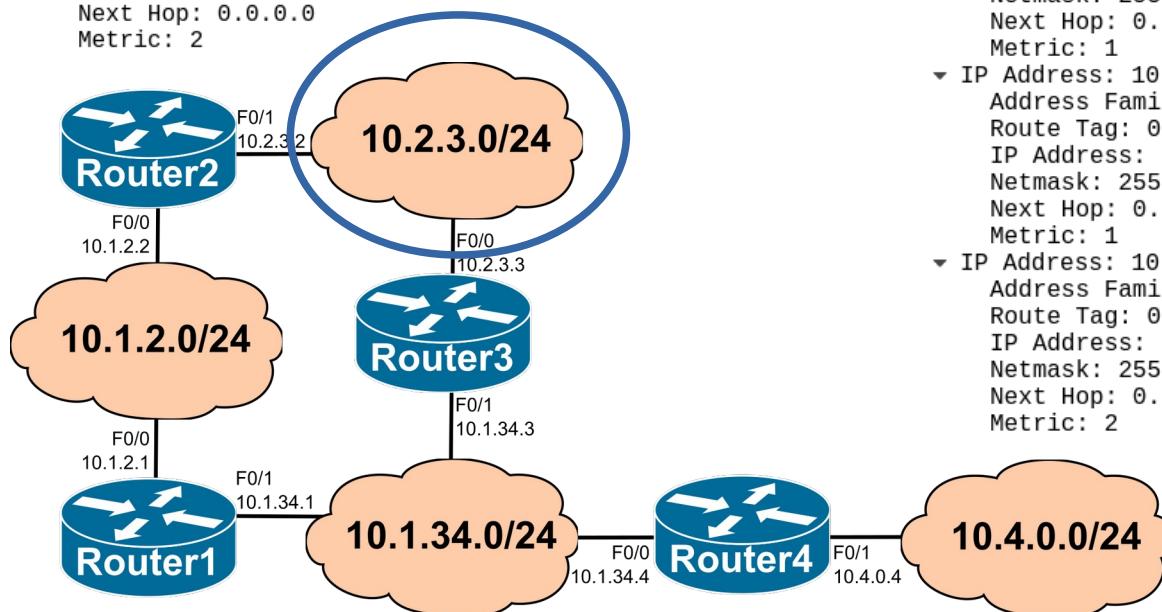
Sent by Router3 without Split-Horizon

- ▶ Internet Protocol Version 4, Src: 10.2.3.3, Dst: 255.255.255.255
- ▶ User Datagram Protocol, Src Port: 520, Dst Port: 520
- ▼ Routing Information Protocol
 - Command: Response (2)
 - Version: RIPv1 (1)
 - ▼ IP Address: 10.1.2.0, Metric: 2
 - Address Family: IP (2)
 - IP Address: 10.1.2.0
 - Metric: 2
 - ▼ IP Address: 10.1.34.0, Metric: 1
 - Address Family: IP (2)
 - IP Address: 10.1.34.0
 - Metric: 1
 - ▼ IP Address: 10.2.3.0, Metric: 1
 - Address Family: IP (2)
 - IP Address: 10.2.3.0
 - Metric: 1
 - ▼ IP Address: 10.4.0.0, Metric: 2
 - Address Family: IP (2)
 - IP Address: 10.4.0.0
 - Metric: 2

RIPv2 Messages (Example)

Sent by Router3 with Split-Horizon

```
► Internet Protocol Version 4, Src: 10.2.3.3, Dst: 224.0.0.9
► User Datagram Protocol, Src Port: 520, Dst Port: 520
▼ Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  ▼ IP Address: 10.1.34.0, Metric: 1
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 10.1.34.0
    Netmask: 255.255.255.0
    Next Hop: 0.0.0.0
    Metric: 1
  ▼ IP Address: 10.4.0.0, Metric: 2
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 10.4.0.0
    Netmask: 255.255.255.0
    Next Hop: 0.0.0.0
    Metric: 2
```



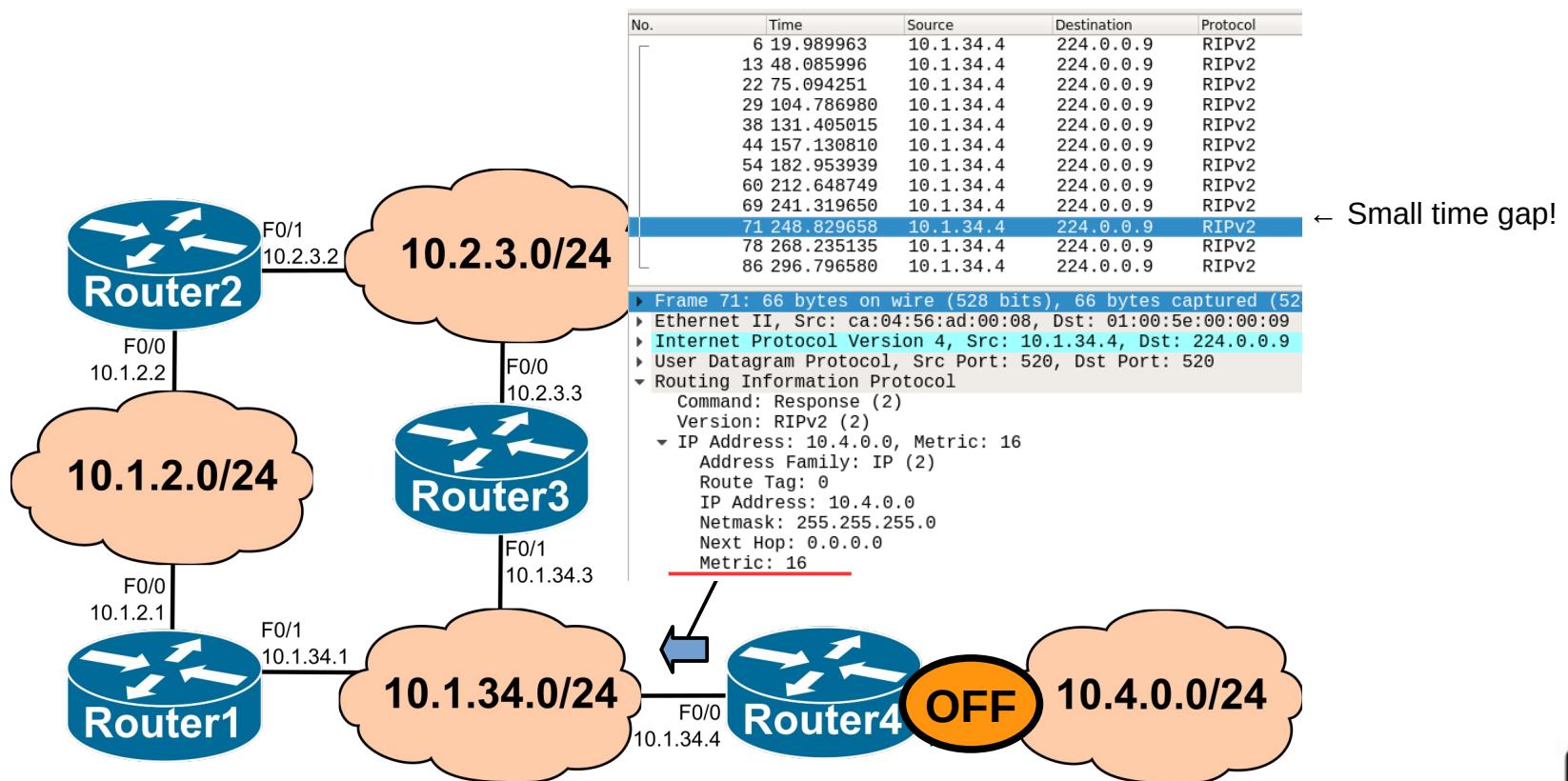
Sent by Router3 without Split-Horizon

```
► Internet Protocol Version 4, Src: 10.2.3.3, Dst: 224.0.0.9
► User Datagram Protocol, Src Port: 520, Dst Port: 520
▼ Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  ▼ IP Address: 10.1.2.0, Metric: 2
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 10.1.2.0
    Netmask: 255.255.255.0
    Next Hop: 10.2.3.2
    Metric: 2
  ▼ IP Address: 10.1.34.0, Metric: 1
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 10.1.34.0
    Netmask: 255.255.255.0
    Next Hop: 0.0.0.0
    Metric: 1
  ▼ IP Address: 10.2.3.0, Metric: 1
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 10.2.3.0
    Netmask: 255.255.255.0
    Next Hop: 0.0.0.0
    Metric: 1
  ▼ IP Address: 10.4.0.0, Metric: 2
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 10.4.0.0
    Netmask: 255.255.255.0
    Next Hop: 0.0.0.0
    Metric: 2
```



Triggered Updates

- Prevents any routing loops that involve more than two routers.
- Whenever a router changes the metric for a route, it is required to send update messages almost immediately, even if it is not yet time for one of the regular update message.
- Neighboring routers update routing tables faster and overall convergence is faster.
 - Including entries that were removed by timeout!



RIPv2 Message Authentication

With Keyed Message Digest (MD5)

```
► Internet Protocol Version 4, Src: 10.2.3.3, Dst: 224.0.0.9
► User Datagram Protocol, Src Port: 520, Dst Port: 520
▼ Routing Information Protocol
    Command: Response (2)
    Version: RIPv2 (2)
    ▼ Authentication: Keyed Message Digest
        Authentication type: Keyed Message Digest (3)
        Digest Offset: 64
        Key ID: 1
        Auth Data Len: 20
        Seq num: 0
        Zero adding:
    ▼ Authentication Data Trailer
        Authentication Data: 7f7d4fc23f02a76b9986f517f3b6a8c1
    ▼ IP Address: 10.1.34.0, Metric: 1
        Address Family: IP (2)
        Route Tag: 0
        IP Address: 10.1.34.0
        Netmask: 255.255.255.0
        Next Hop: 0.0.0.0
        Metric: 1
    ▼ IP Address: 10.4.0.0, Metric: 2
        Address Family: IP (2)
        Route Tag: 0
        IP Address: 10.4.0.0
        Netmask: 255.255.255.0
        Next Hop: 0.0.0.0
        Metric: 2
```

With Clear Text Authentication Useless!

```
► Internet Protocol Version 4, Src: 10.2.3.3, Dst: 224.0.0.9
► User Datagram Protocol, Src Port: 520, Dst Port: 520
▼ Routing Information Protocol
    Command: Response (2)
    Version: RIPv2 (2)
    ▼ Authentication: Simple Password
        Authentication type: Simple Password (2)
        Password: labcom
    ▼ IP Address: 10.1.2.0, Metric: 2
        Address Family: IP (2)
        Route Tag: 0
        IP Address: 10.1.2.0
        Netmask: 255.255.255.0
        Next Hop: 0.0.0.0
        Metric: 2
    ▼ IP Address: 10.1.34.0, Metric: 1
        Address Family: IP (2)
        Route Tag: 0
        IP Address: 10.1.34.0
        Netmask: 255.255.255.0
        Next Hop: 0.0.0.0
        Metric: 1
    ▼ IP Address: 10.4.0.0, Metric: 2
        Address Family: IP (2)
        Route Tag: 0
        IP Address: 10.4.0.0
        Netmask: 255.255.255.0
        Next Hop: 0.0.0.0
        Metric: 2
```



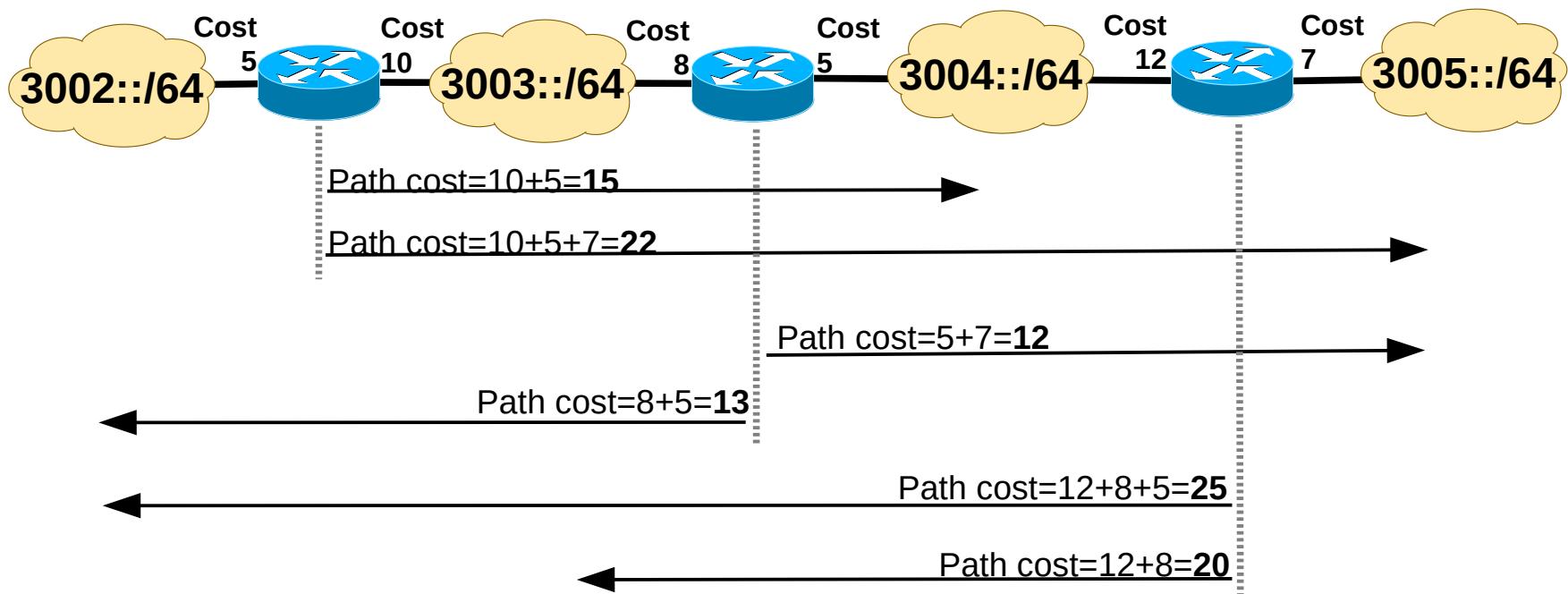
RIPng for IPv6 Routing

- Similar to IPv4 RIPv2:
 - ◆ Distance-vector concept, radius of 15 hops, infinity metric is 16, split-horizon, triggered update.
- Differences between RIPv2 and RIPng
 - ◆ Uses IPv6 for transport.
 - ◆ Uses link-local addresses (not the global ones).
 - ◆ IPv6 prefix, next-hop IPv6 link-local address.
 - ◆ Uses multicast group address FF02::9 (all-RIP-routers) as the destination address for RIP updates.
 - ◆ Routers always add the cost of the interface to the metric received.
 - ◆ Metric is sum of “output interfaces” costs to destination and not number of hops.
 - ◆ If all costs are 1, metric is number of “output interfaces” to destination.
 - ◆ Allows for node/interface costs other than 1.
 - ◆ Cisco calls it “cost offset” per interface (out or in direction).
 - ◆ Cost to network is given by the sum of all output interfaces costs along the path.
 - ◆ With the infinity metric value at 16, this require careful configurations.
 - ◆ Routers always announce directed connected networks.
 - ◆ in IOS Cisco
 - ◆ Activation per interface, named process, more than one active process.



RIPng Path Costs

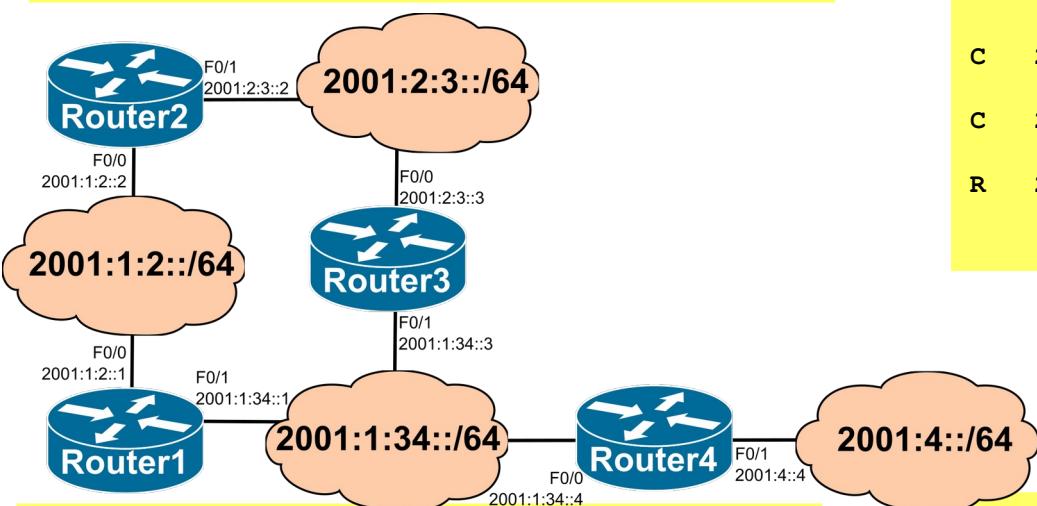
- Each router link/interface has an associated RIPng cost.
 - The total cost between a router and a network is given by the sum of all RIPng costs of the (routers) output interfaces along the path.
 - ◆ Routers to access directly connect networks never use RIPng paths.



IPv6 Routing Tables with RIPng

Router2

```
C  2001:1:2::/64 [0/0]
    via FastEthernet0/0, directly connected
R  2001:1:34::/64 [120/2]
    via FE80::C801:54FF:FE41:8, FastEthernet0/0
    via FE80::C803:56FF:FE0A:8, FastEthernet0/1
C  2001:2:3::/64 [0/0]
    via FastEthernet0/1, directly connected
R  2001:4::/64 [120/3]
    via FE80::C801:54FF:FE41:8, FastEthernet0/0
    via FE80::C803:56FF:FE0A:8, FastEthernet0/1
```



Router1

```
C  2001:1:2::/64 [0/0]
    via FastEthernet0/0, directly connected
C  2001:1:34::/64 [0/0]
    via FastEthernet0/1, directly connected
R  2001:2:3::/64 [120/2]
    via FE80::C802:54FF:FEF5:8, FastEthernet0/0
    via FE80::C803:56FF:FE0A:6, FastEthernet0/1
R  2001:4::/64 [120/2]
    via FE80::C804:56FF:FEAD:8, FastEthernet0/1
```

Assuming all interfaces with cost 1.

Router3

```
R  2001:1:2::/64 [120/2]
    via FE80::C802:54FF:FEF5:6, FastEthernet0/0
    via FE80::C801:54FF:FE41:6, FastEthernet0/1
C  2001:1:34::/64 [0/0]
    via FastEthernet0/1, directly connected
C  2001:2:3::/64 [0/0]
    via FastEthernet0/0, directly connected
R  2001:4::/64 [120/2]
    via FE80::C804:56FF:FEAD:8, FastEthernet0/1
```

Router4

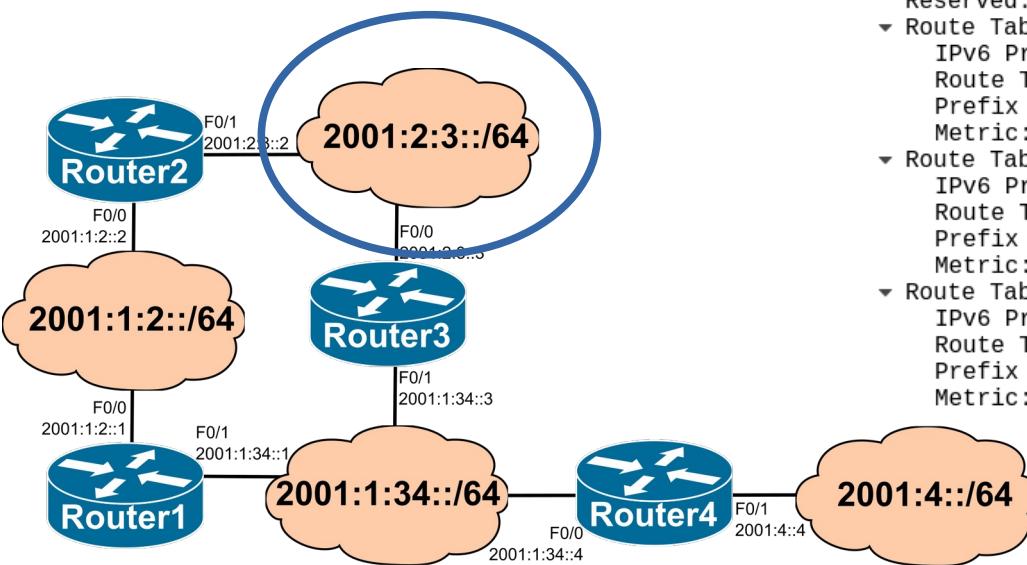
```
R  2001:1:2::/64 [120/2]
    via FE80::C801:54FF:FE41:6, FastEthernet0/0
C  2001:1:34::/64 [0/0]
    via FastEthernet0/0, directly connected
R  2001:2:3::/64 [120/2]
    via FE80::C803:56FF:FE0A:6, FastEthernet0/0
C  2001:4::/64 [0/0]
    via FastEthernet0/1, directly connected
```



RIPng Messages (Example)

Sent by Router2 with Split-Horizon

- ▶ Internet Protocol Version 6, Src: fe80::c802:54ff:fe5:6, Dst: ff02::9
- ▶ User Datagram Protocol, Src Port: 521, Dst Port: 521
- ▶ RIPng
 - Command: Response (2)
 - Version: 1
 - Reserved: 0000
 - Route Table Entry: IPv6 Prefix: 2001:1:2::/64 Metric: 1
 - IPv6 Prefix: 2001:1:2::
 - Route Tag: 0x0000
 - Prefix Length: 64
 - Metric: 1
 - Route Table Entry: IPv6 Prefix: 2001:2:3::/64 Metric: 1
 - IPv6 Prefix: 2001:2:3::
 - Route Tag: 0x0000
 - Prefix Length: 64
 - Metric: 1



Sent by Router3 with Split-Horizon

- ▶ Internet Protocol Version 6, Src: fe80::c803:56ff:fe0a:8, Dst: ff02::9
- ▶ User Datagram Protocol, Src Port: 521, Dst Port: 521
- ▶ RIPng
 - Command: Response (2)
 - Version: 1
 - Reserved: 0000
 - Route Table Entry: IPv6 Prefix: 2001:2:3::/64 Metric: 1
 - IPv6 Prefix: 2001:2:3::
 - Route Tag: 0x0000
 - Prefix Length: 64
 - Metric: 1
 - Route Table Entry: IPv6 Prefix: 2001:1:34::/64 Metric: 1
 - IPv6 Prefix: 2001:1:34::
 - Route Tag: 0x0000
 - Prefix Length: 64
 - Metric: 1
 - Route Table Entry: IPv6 Prefix: 2001:4::/64 Metric: 2
 - IPv6 Prefix: 2001:4::
 - Route Tag: 0x0000
 - Prefix Length: 64
 - Metric: 2



Open Shortest Path First (OSPF) Protocol

- OSPF is an open-standard protocol based primarily on RFC 2328.
- OSPF is a link-state routing protocol
 - ◆ Respond quickly to network changes,
 - ◆ Send triggered updates when a network change occurs,
 - ◆ Send periodic updates, known as link-state refresh, at long time intervals, such as every 30 minutes.
- Routers running OSPF collect routing information from all other routers in the network (or from within a defined area of the network)
- And then each router independently calculates its best paths to all destinations in the network, using Dijkstra's (SPF) algorithm.



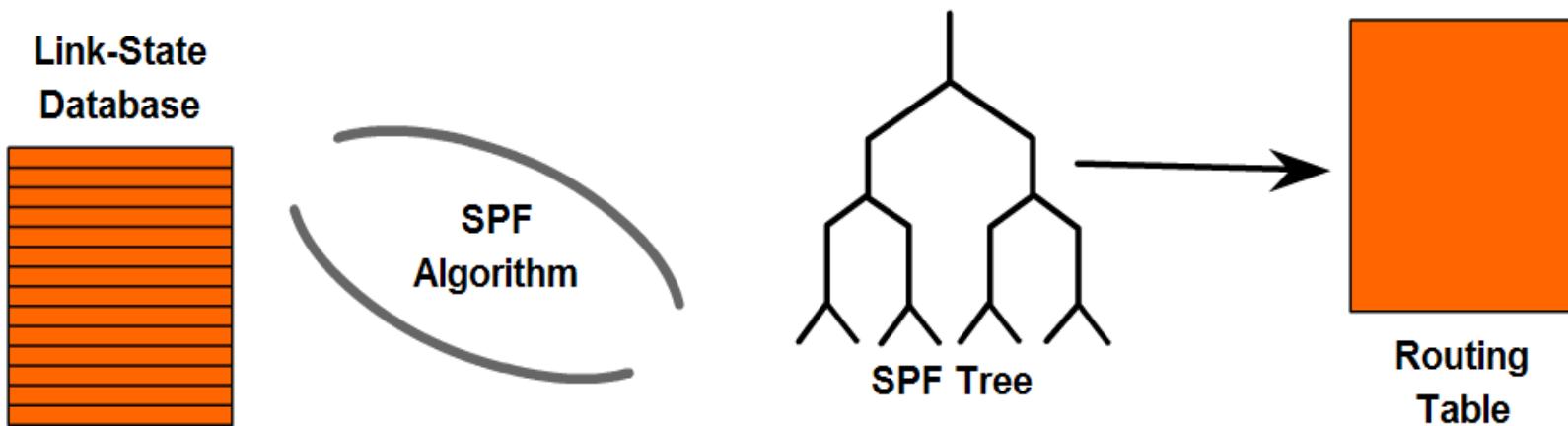
OSPF Necessary Routing Information

- For all the routers in the network to make consistent routing decisions, each link-state router must keep a record of the following information:
 - Its immediate neighbor routers
 - If the router loses contact with a neighbor router, within a few seconds it invalidates all paths through that router and recalculates its paths through the network.
 - For OSPF, adjacency information about neighbors is stored in the OSPF neighbor table, also known as an adjacency database.
 - All the other routers in the network, or in its area of the network, and their attached networks
 - The router recognizes other routers and networks through LSAs, which are flooded through the network.
 - LSAs are stored in a topology table or database (which is also called an LSDB).
 - The best paths to each destination
 - Each router independently calculates the best paths to each destination in the network using Dijkstra's (SPF) algorithm.
 - All paths are kept in the LSDB.
 - The best paths are then offered to the routing table (also called the forwarding database).
 - Packets arriving at the router are forwarded based on the information held in the routing table.



Link-State Protocol Operation

- Link-state routing protocols generate routing updates only when a change occurs in the network topology.
- When a link changes state, the device that detected the change creates a Link-State Advertisement (LSA) concerning that link.
 - ◆ LSA propagates to neighbor devices using a special multicast address.
- Each router stores the LSA, forwards the LSA to neighboring devices and updates its Link-State DataBase (LSDB).
- Link-state routers find the best paths to a destination by applying Dijkstra's algorithm, also known as SPF, against the LSDB to build the SPF tree.
- Each router selects the best paths from their SPF tree and places them in their routing table.



Link-State Advertisement (LSA)

- LSAs report the state of routers and the links between routers.
- Link-state information must be synchronized between routers.
- LSAs have the following characteristics:
 - ◆ LSAs are reliable. There is a method for acknowledging their delivery.
 - ◆ LSAs are flooded throughout the area (or throughout the domain if there is only one area).
 - ◆ LSAs have a sequence number and a set lifetime, so each router recognizes that it has the most current version of the LSA.
 - ◆ LSAs are periodically refreshed to confirm topology information before they age out of the LSDB.



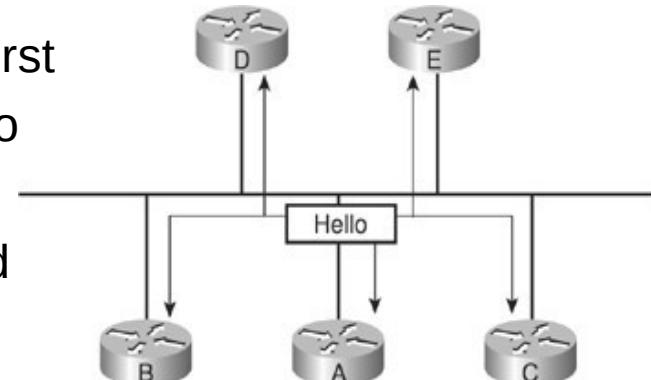
OSPF Router ID (RID)

- The Router ID identifies the router and is:
 - ◆ The highest IPv4 address of all router interfaces at the moment of the OSPF process activation.
 - ◆ A value administratively defined.
- If a physical interface address is being used as the router ID, and that physical interface fails, and the router (or OSPF process) is restarted, the router ID will change.
 - ◆ This change in router ID makes it more difficult for network administrators to troubleshoot and manage OSPF.
- Administratively defining the RID or using loopback interfaces for the router ID forces the router ID to stay the same, regardless of the state of the physical interfaces.



OSPF Adjacencies

- A router running a link-state routing protocol must first establish neighbor adjacencies, by exchanging hello packets with the neighboring routers
- The router sends and receives Hello packets to and from its neighboring routers.
 - ◆ The destination address is typically a multicast address.
 - ◆ It is possible to define unicast OSPF relations.
- The routers exchange hello packets subject to protocol-specific parameters, such as checking whether the neighbor is in the same area, using the same hello interval, and so on.
 - ◆ Routers declare the neighbor up when the exchange is complete.
- Two OSPF routers on a point-to-point serial link, usually encapsulated in High-Level Data Link Control (HDLC) or Point-to-Point Protocol (PPP), form a full adjacency with each other.
- However, OSPF routers on broadcast networks, such as LAN links, elect one router as the designated router (DR) and another as the backup designated router (BDR).
 - ◆ All other routers on the LAN form full adjacencies with these two routers and pass LSAs only to them.



DR and BDR Election

- The first OSPF router to boot becomes the Designated Router (DR).
- The second router to boot becomes the Backup Designated Router (BDR).
- If multiple routers boot simultaneously,
 - ◆ The DR will be the router with the highest priority. The BDR the second.
 - ◆ The OSPF priority is a administratively defined parameter.
 - ◆ In case of tie, it will be chosen the router with the highest Router ID (RID).
- When the DR fails, the BDR assumes the role of DR.
 - ◆ The BDR does not perform any DR functions when the DR is operating.
 - ◆ The choice of the new BDR is done according to some criteria of the initial election.
- After the election, the DR and BDR maintain that role, independently of which routers join the OSPF process.
- The ID of an OSPF Network is the IP address of the network's Designated Router (DR) interface.



OSPF LS Database

- The OSPF database (LSDB) is organized in two tables.
 - Router Link States – Routers related information table.
 - The routers are identified by theirs RID.
 - Net Link States – Networks/Links related information table.
 - Networks are identified by their ID.

OSPF Router with ID (20.20.20.1) (Process ID 1)

Router Link States (Area 0)

| Link ID | ADV Router | Age | Seq# | Checksum | Link count |
|------------|------------|-----|------------|----------|------------|
| 20.20.20.1 | 20.20.20.1 | 40 | 0x8000000A | 0x00E7FB | 2 |
| 30.30.30.2 | 30.30.30.2 | 69 | 0x80000006 | 0x002906 | 2 |
| 30.30.30.3 | 30.30.30.3 | 41 | 0x80000007 | 0x00283D | 2 |

Net Link States (Area 0)

| Link ID | ADV Router | Age | Seq# | Checksum |
|------------|------------|-----|------------|----------|
| 10.10.10.3 | 30.30.30.3 | 41 | 0x80000001 | 0x00051C |
| 20.20.20.2 | 30.30.30.2 | 70 | 0x80000001 | 0x00A164 |
| 30.30.30.3 | 30.30.30.3 | 154 | 0x80000001 | 0x00A91C |



OSPF LS Database Tables (1)

- Router Link States

- For each router, it contains the information about the networks directly connected to that router.

```
LS age: 321
Options: (No TOS-capability, DC)
LS Type: Router Links
Link State ID: 20.20.20.1 ← Router ID
Advertising Router: 20.20.20.1
LS Seq Number: 8000000A
Checksum: 0xE7FB
Length: 48
Number of Links: 2 ← Number of Links
```

```
Link connected to: a Transit Network ← Network Type
(Link ID) Designated Router address: 20.20.20.2 ← Network ID
(Link Data) Router Interface address: 20.20.20.1 ← Interface IP Address
Number of TOS metrics: 0
TOS 0 Metrics: 1 ← Interface Cost
```

```
Link connected to: a Transit Network
(Link ID) Designated Router address: 10.10.10.3
(Link Data) Router Interface address: 10.10.10.1
Number of TOS metrics: 0
TOS 0 Metrics: 1
```



OSPF LS Database Tables (2)

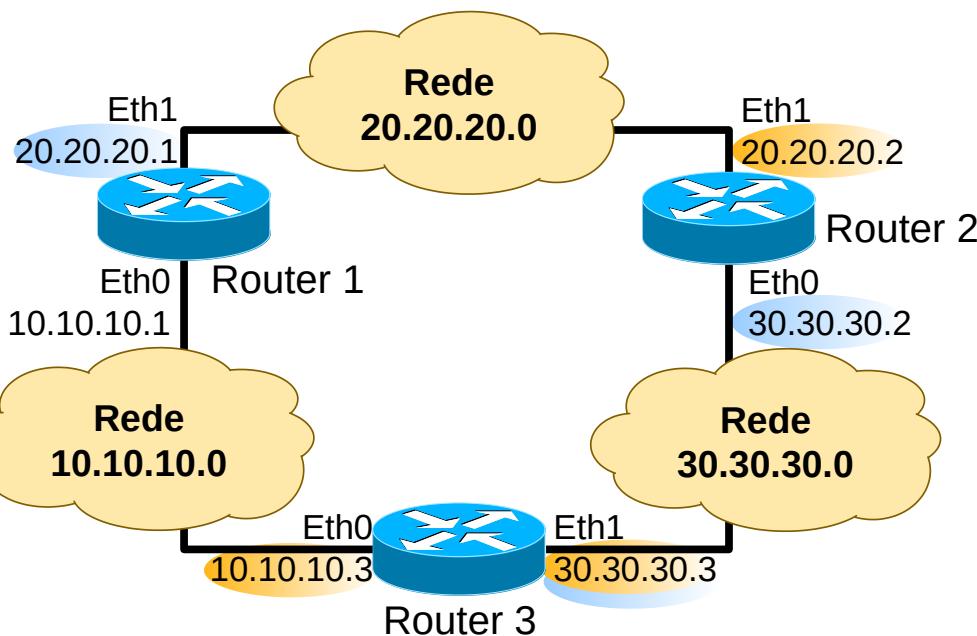
- Network Link States

- For each network, it contains the information about the routers directly attached to that network.

```
Routing Bit Set on this LSA
LS age: 483
Options: (No TOS-capability, DC)
LS Type: Network Links
Link State ID: 10.10.10.3 (address of Designated Router) ← Network ID
Advertising Router: 30.30.30.3
LS Seq Number: 80000001
Checksum: 0x51C
Length: 32
Network Mask: /24
Attached Router: 30.30.30.3 } ← Attached routers (RID)
Attached Router: 20.20.20.1 }
```



OSPF LSDatabase Example



Routing Bit Set on this LSA

LS age: 208
Options: (No TOS-capability, DC)
LS Type: Network Links

Link State ID: 20.20.20.2 (address of Designated Router)
Advertising Router: 30.30.30.2
LS Seq Number: 80000001

Checksum: 0xA164
Length: 32

Network Mask: /24

Attached Router: 30.30.30.2
Attached Router: 20.20.20.1

Network 20.20.20.0's Network Link State

LS age: 321
Options: (No TOS-capability, DC)
LS Type: Router Links
Link State ID: 20.20.20.1
Advertising Router: 20.20.20.1
LS Seq Number: 8000000A
Checksum: 0xE7FB
Length: 48
Number of Links: 2

Link connected to: a Transit Network
(Link ID) Designated Router address: 20.20.20.2
(Link Data) Router Interface address: 20.20.20.1
Number of TOS metrics: 0
TOS 0 Metrics: 1

Link connected to: a Transit Network
(Link ID) Designated Router address: 10.10.10.3
(Link Data) Router Interface address: 10.10.10.1
Number of TOS metrics: 0
TOS 0 Metrics: 1

Router 1's Router Link State



OSPF Packets

- Hello - Discovers neighbors and builds adjacencies between them.
- Database Description (DBD) - Checks for database synchronization between routers.
- Link-State Request (LSR) - Requests specific link-state records from another router.
- Link-State Update (LSU) - Sends specifically requested link-state records.
- LSAck - Acknowledges the other packet types.



OSPF Packet Format

Version Number

- Set to 2 for OSPF Version 2, the IPv4 version of OSPF.
- Set to 3 for OSPF Version 3, the IPv6 version of OSPF.

Type

- Differentiates the five OSPF packet types.

Packet Length

- The length of the OSPF packet in bytes.

Router ID

- Defines which router is the packet's source.

Area ID

- Defines the area in which the packet originated.

Checksum

- Used for packet header error detection to ensure that the OSPF packet was not corrupted during transmission.

Authentication Type

- An option in OSPF that describes either no authentication, clear-text passwords, or encrypted message digest 5 (MD5) for router authentication.

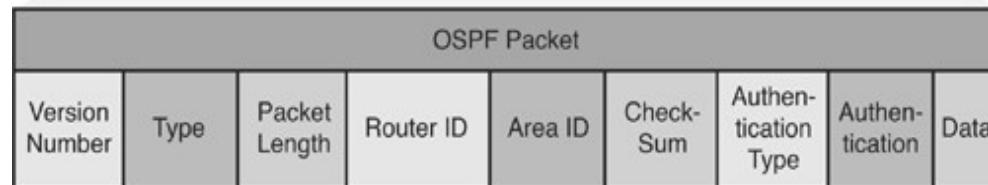
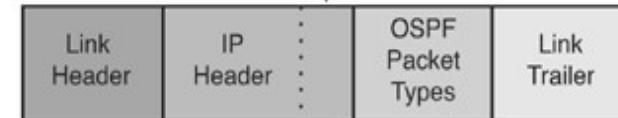
Authentication

- Used with authentication type.

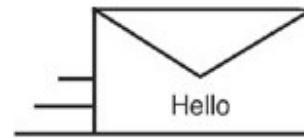
Data, contains different information, depending on the OSPF packet type:

- For the Hello packet - Contains a list of known neighbors.
- For the DBD packet - Contains a summary of the LSDB, which includes all known router IDs and their last sequence number, among several other fields.
- For the LSR packet - Contains the type of LSU needed and the router ID of the router that has the needed LSU.
- For the LSU packet - Contains the full LSA entries. Multiple LSA entries can fit in one OSPF update packet.
- For the LSAck packet - This data field is empty.

Protocol
ID No.
89 = OSPF



OSPF Hello Packets



| |
|--------------------------|
| Router ID |
| Hello/Dead Intervals* |
| Neighbors |
| Area ID* |
| Router Priority |
| DR IP Address |
| BDR IP Address |
| Authentication Password* |
| Stub Area Flag* |

*Entry Must Match on Neighboring Routers

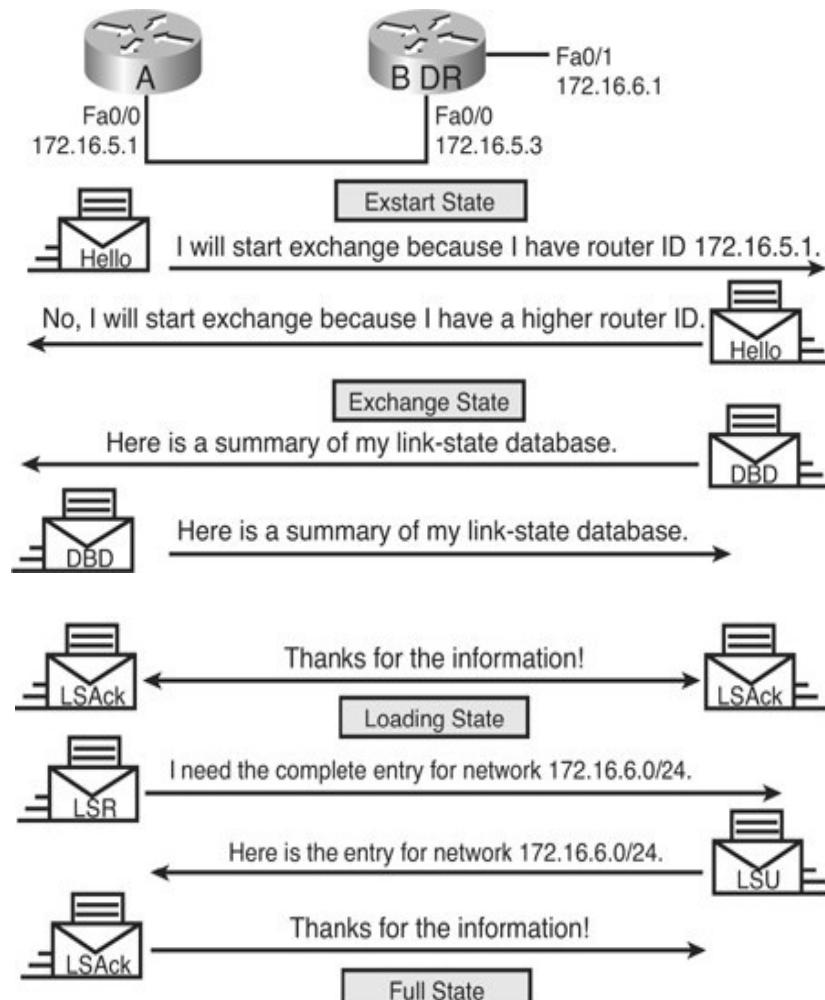
- An hello packet contains the following information:

- ◆ Router ID
 - A 32-bit number that uniquely identifies the router.
- ◆ Hello and dead intervals
 - The hello interval specifies how often, in seconds, a router sends hello packets (10 seconds is the default on multiaccess networks).
 - The dead interval is the amount of time in seconds that a router waits to hear from a neighbor before declaring the neighbor router out of service (the dead interval is four times the hello interval by default).
 - These timers must be the same on neighboring routers; otherwise an adjacency will not be established.
- ◆ Neighbors
 - The Neighbors field lists the adjacent routers with which this router has established bidirectional communication.
 - Bidirectional communication is indicated when the router sees itself listed in the Neighbors field of the hello packet from the neighbor.
- ◆ Area ID
 - To communicate, two routers must share a common segment, and their interfaces must belong to the same OSPF area on that segment.
 - These routers will all have the same link-state information for that area.
- ◆ Router priority
 - An 8-bit number that indicates a router's priority. Priority is used when electing a DR and BDR.
- ◆ DR and BDR IP addresses
 - If known, the IP addresses of the DR and BDR for the specific multiaccess network.
- ◆ Authentication password
 - If router authentication is enabled, two routers must exchange the same password.
 - Authentication is not required, but if it is enabled, all peer routers must have the same password.
- ◆ Stub area flag
 - A stub area is a special area.
 - The stub area technique reduces routing updates by replacing them with a default route.
 - Two neighboring routers must agree on the stub area flag in the hello packets.

- Hello Interval, Dead Interval, Area ID, Authentication Password and Stub Area Flag fields must match on neighboring routers for them to establish an adjacency.



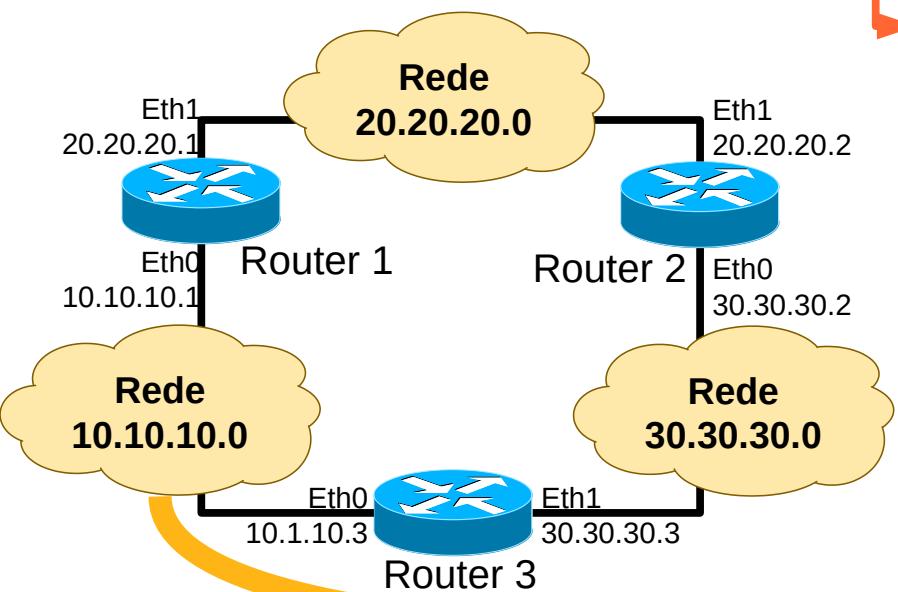
Discovering the Network Routes



- A master and slave relationship is created between each router and its adjacent DR and BDR.
 - ◆ Only the DR exchanges and synchronizes link-state information with the routers to which it has established adjacencies.
- The master and slave routers exchange one or more DBD packets.
 - ◆ A DBD includes information about the LSA entry header that appears in the router's LSDB.
 - ◆ The entries can be about a link or about a network.
 - ◆ Each LSA entry header includes information about the link-state type, the address of the advertising router, the link's cost, and the sequence number.
 - ◆ The router uses the sequence number to determine the "newness" of the received link-state information.
- It acknowledges the receipt of the DBD using the LSAck packet.
 - ◆ It compares the information it received with the information it has in its own LSDB.
- If the DBD has a more current link-state entry, the router sends an LSR to the other router.
- The other router responds with the complete information about the requested entry in an LSU packet.
- Again, when the router receives an LSU, it sends an LSAck.
- The router adds the new link-state entries to its LSDB.



OSPF Example



OSPF activated on Router 1

OSPF activated on Router 3

OSPF activated on Router 2

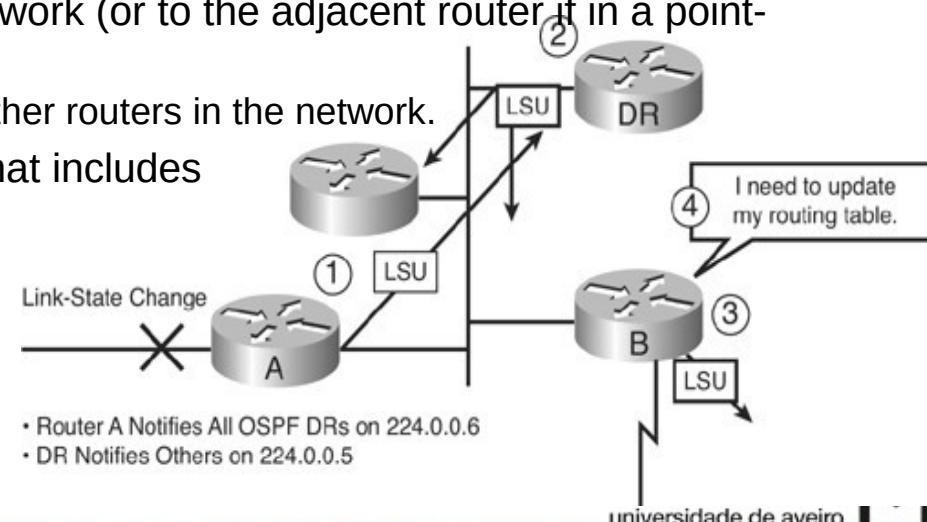
| Time | Source | Destination | Protocol | Info |
|------------|------------|-------------|----------|----------------|
| 0.000000 | 10.10.10.1 | 224.0.0.5 | OSPF | Hello Packet |
| 10.002318 | 10.10.10.1 | 224.0.0.5 | OSPF | Hello Packet |
| 20.003116 | 10.10.10.1 | 224.0.0.5 | OSPF | Hello Packet |
| ----- | | | | |
| 80.000000 | 10.10.10.3 | 224.0.0.5 | OSPF | Hello Packet |
| 83.683033 | 10.10.10.3 | 224.0.0.5 | OSPF | LS Update |
| 83.715683 | 10.10.10.3 | 224.0.0.5 | OSPF | Hello Packet |
| 83.717864 | 10.10.10.1 | 10.10.10.3 | OSPF | Hello Packet |
| 83.726166 | 10.10.10.3 | 10.10.10.1 | OSPF | DB Descr. |
| 83.726258 | 10.10.10.3 | 10.10.10.1 | OSPF | Hello Packet |
| 83.728433 | 10.10.10.1 | 10.10.10.3 | OSPF | DB Descr. |
| 83.732590 | 10.10.10.3 | 10.10.10.1 | OSPF | DB Descr. |
| 83.734733 | 10.10.10.1 | 10.10.10.3 | OSPF | DB Descr. |
| 83.738942 | 10.10.10.3 | 10.10.10.1 | OSPF | LS Request |
| 83.741083 | 10.10.10.1 | 10.10.10.3 | OSPF | LS Update |
| 84.240362 | 10.10.10.3 | 224.0.0.5 | OSPF | LS Update |
| 86.245792 | 10.10.10.3 | 224.0.0.5 | OSPF | LS Acknowledge |
| 86.380876 | 10.10.10.1 | 224.0.0.5 | OSPF | Hello Packet |
| 86.741036 | 10.10.10.1 | 224.0.0.5 | OSPF | LS Acknowledge |
| 93.721376 | 10.10.10.3 | 224.0.0.5 | OSPF | Hello Packet |
| 96.380005 | 10.10.10.1 | 224.0.0.5 | OSPF | Hello Packet |
| ----- | | | | |
| 213.780338 | 10.10.10.3 | 224.0.0.5 | OSPF | Hello Packet |
| 216.542473 | 10.10.10.1 | 224.0.0.5 | OSPF | Hello Packet |
| 216.568852 | 10.10.10.1 | 224.0.0.5 | OSPF | LS Update |
| 217.048427 | 10.10.10.1 | 224.0.0.5 | OSPF | LS Update |
| 217.084909 | 10.10.10.1 | 224.0.0.5 | OSPF | LS Update |
| 219.067748 | 10.10.10.3 | 224.0.0.5 | OSPF | LS Acknowledge |
| 219.650308 | 10.10.10.1 | 224.0.0.5 | OSPF | LS Update |
| 222.150349 | 10.10.10.3 | 224.0.0.5 | OSPF | LS Acknowledge |
| 223.779492 | 10.10.10.3 | 224.0.0.5 | OSPF | Hello Packet |
| 224.284149 | 10.10.10.3 | 224.0.0.5 | OSPF | LS Update |
| 224.789598 | 10.10.10.1 | 224.0.0.5 | OSPF | LS Update |
| 224.789775 | 10.10.10.3 | 224.0.0.5 | OSPF | LS Update |
| 226.545718 | 10.10.10.1 | 224.0.0.5 | OSPF | Hello Packet |
| 226.785254 | 10.10.10.1 | 224.0.0.5 | OSPF | LS Acknowledge |
| 227.294756 | 10.10.10.3 | 224.0.0.5 | OSPF | LS Acknowledge |
| 233.779863 | 10.10.10.3 | 224.0.0.5 | OSPF | Hello Packet |
| 236.544658 | 10.10.10.1 | 224.0.0.5 | OSPF | Hello Packet |



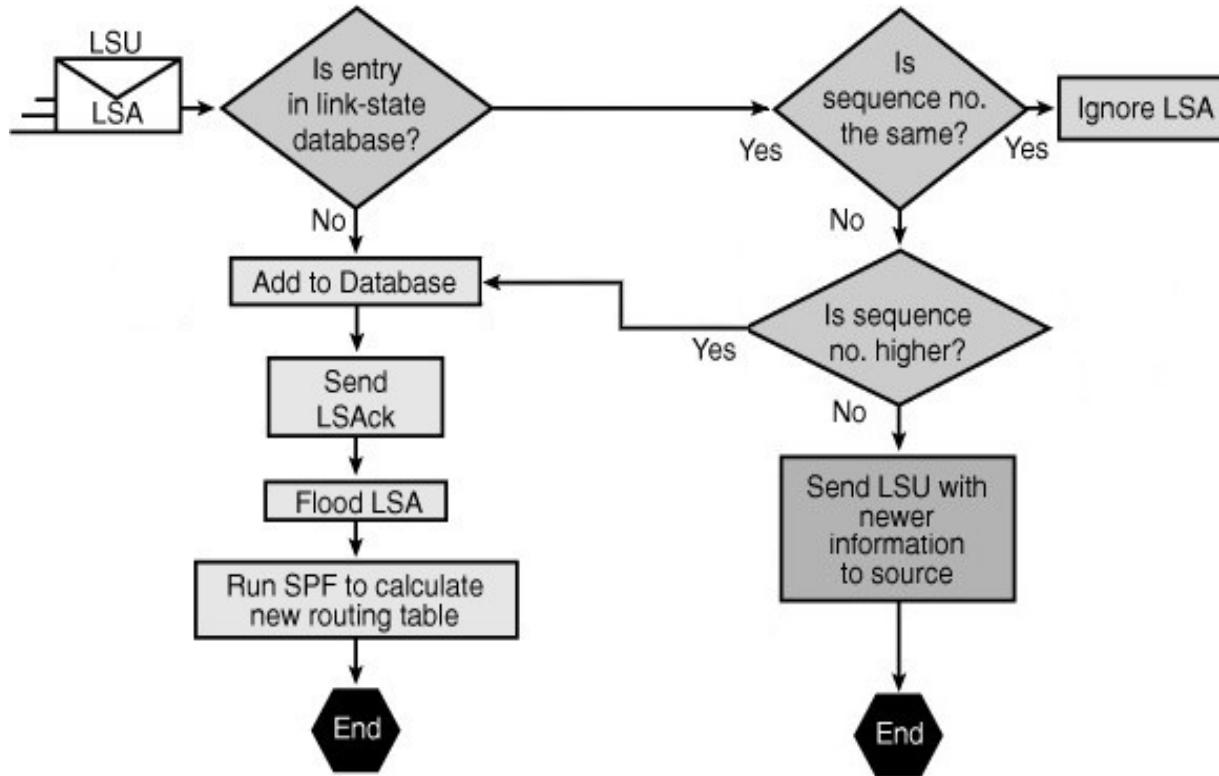
Maintaining Routing Information

- Flooding process:

- ◆ A router notices a change in a link state and multicasts an LSU packet, which includes the updated LSA entry with the sequence number incremented, to 224.0.0.6.
 - ◆ This address goes to all OSPF DRs and BDRs.
 - ◆ On point-to-point links, the LSU is multicast to 224.0.0.5.)
 - ◆ An LSU packet might contain several distinct LSAs.
 - ◆ The DR receives the LSU, processes it, acknowledges the receipt of the change and floods the LSU to other routers on the network using the OSPF multicast address 224.0.0.5.
 - ◆ After receiving the LSU, each router responds to the DR with an LSAck.
 - ◆ To make the flooding procedure reliable, each LSA must be acknowledged separately.
 - ◆ If a router is connected to other networks, it floods the LSU to those other networks by forwarding the LSU to the DR of the other network (or to the adjacent router if in a point-to-point network).
 - ◆ That DR, in turn, multicasts the LSU to the other routers in the network.
 - ◆ The router updates its LSDB using the LSU that includes the changed LSA.
 - ◆ It then recomputes the SPF algorithm against the updated database after a short delay and updates the routing table as necessary.



LSA Operation



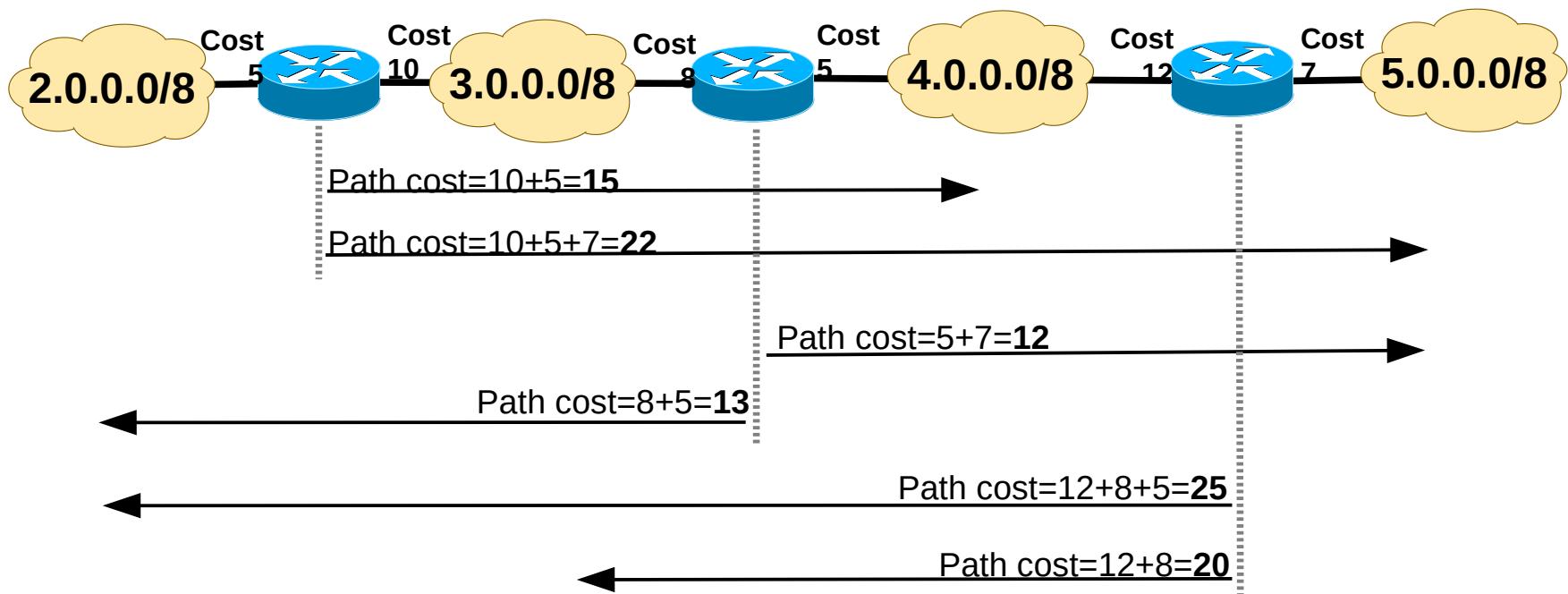
- When each router receives the LSU:

- If the LSA entry does not already exist, the router adds the entry to its LSDB, sends back a link-state acknowledgment (LSAck), floods the information to other routers, runs SPF, and updates its routing table.
- If the entry already exists and the received LSA has the same sequence number, the router ignores the LSA entry.
- If the entry already exists but the LSA includes newer information (it has a higher sequence number), the router adds the entry to its LSDB, sends back an LSAck, floods the information to other routers, runs SPF, and updates its routing table.
- If the entry already exists but the LSA includes older information, it sends an LSU to the sender with its newer information.

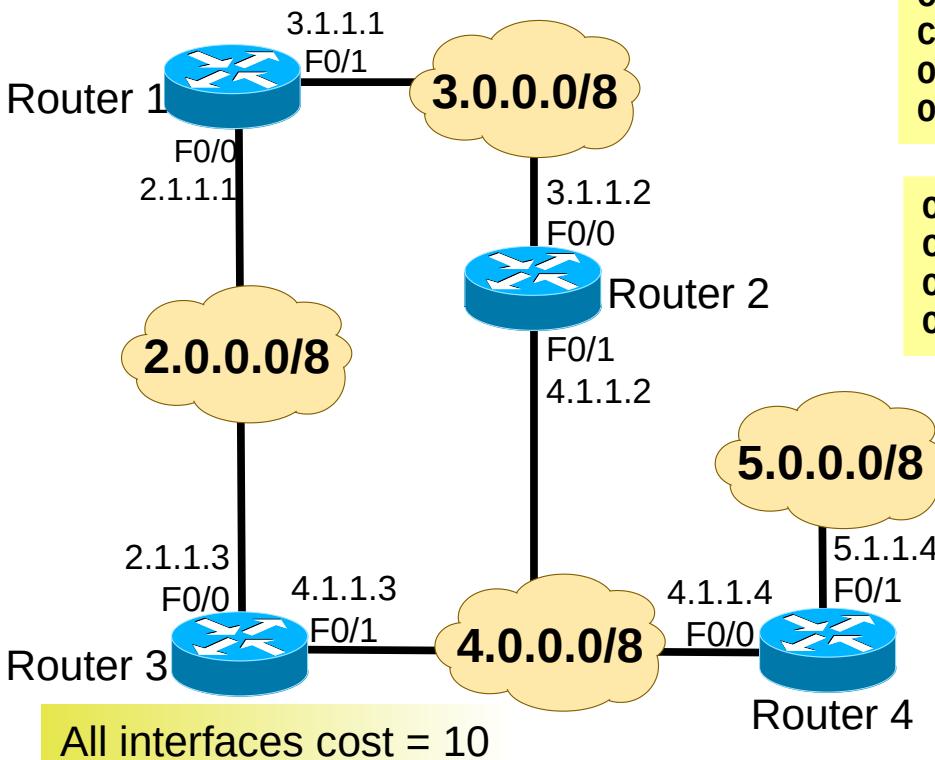


OSPF Path Costs

- Each router link/interface has an associated OSPF cost.
- The total cost between a router and a network is given by the sum of all OSPF costs of the (routers) output interfaces along the path.
 - ◆ Routers to access directly connect networks never use OSPF paths.



OSPF Example



```
C 2.0.0.0/8 is directly connected, F0/0
C 3.0.0.0/8 is directly connected, F0/1
O 4.0.0.0/8 [110/20] via 2.1.1.3, 00:01:18, F0/0
O 5.0.0.0/8 [110/30] via 2.1.1.3, 00:01:00, F0/0
```

```
O 2.0.0.0/8 [110/20] via 3.1.1.1, 00:01:13, F0/0
C 3.0.0.0/8 is directly connected, F0/0
O 4.0.0.0/8 [110/30] via 3.1.1.1, 00:01:13, F0/0
O 5.0.0.0/8 [110/40] via 3.1.1.1, 00:01:10, F0/0
```

Router 1 and Router 2 after disconnecting the F0/1 at Router2

```
C 2.0.0.0/8 is directly connected, F0/0
C 3.0.0.0/8 is directly connected, F0/1
O 4.0.0.0/8 [110/15] via 3.1.1.2, 00:01:13, F0/1
O 5.0.0.0/8 [110/25] via 3.1.1.2, 00:01:10, F0/1
```

Router1, now with the cost of Router2 F0/1 interface equal to 5

```
C 2.0.0.0/8 is directly connected, F0/0
C 3.0.0.0/8 is directly connected, F0/1
O 4.0.0.0/8 [110/20] via 3.1.1.2, 00:01:13, F0/1
[110/20] via 2.1.1.3, 00:01:31, F0/0
O 5.0.0.0/8 [110/30] via 3.1.1.2, 00:01:10, F0/1
[110/30] via 2.1.1.3, 00:01:10, F0/0
```



IPv6 Routing - OSPFv3

- Based on OSPFv2, with enhancements:
 - ◆ Uses IPv6 for transport
 - ◆ Distributes IPv6 prefixes
 - ◆ Uses multicast group addresses FF02::5 (OSPF IGP) and FF02::6 (OSPF IGP Designated Routers)
 - ◆ Runs over a link rather than a subnet
 - ◆ Multiple instances per link
 - ◆ Topology not IPv6-specific
 - Router ID, Area ID, Link ID remain a 4 bytes number
 - Neighbors are always identified by Router ID (4 bytes)
 - With an additional table with mapping between IPv6 prefixes and Link IDs
 - ◆ Uses link-local addresses as IPv6 source addresses

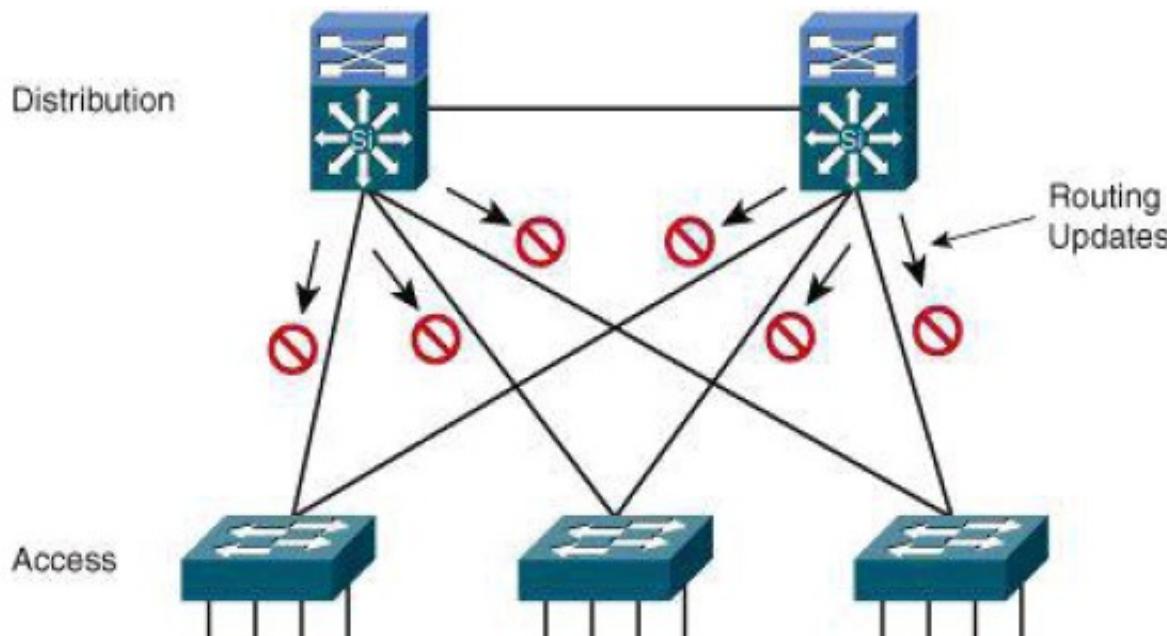


OSPFv3 - LSA Types

- Link LSA (Type 8)
 - ◆ Informs neighbors of link local address
 - ◆ Informs neighbors of IPv6 prefixes on link
- Intra-Area Prefix LSA (Type 9)
 - ◆ Associates IPv6 prefixes with a network or router
- Flooding scope for LSAs has been generalized
 - ◆ Three flooding scopes for LSAs
 - Link-local
 - Area
 - AS
- LSA Type encoding expanded to 16 bits
 - ◆ Includes flooding scope



Passive Interfaces on Access Layer

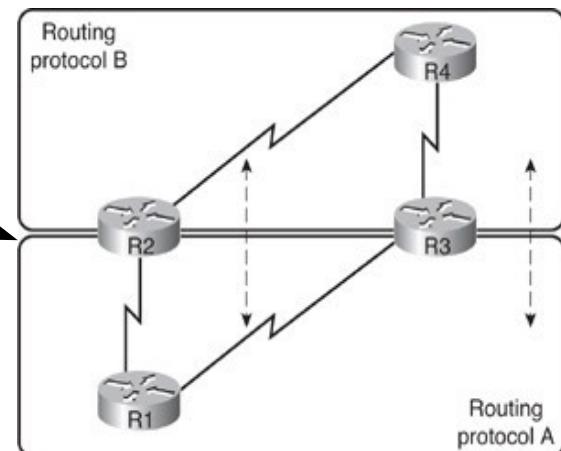
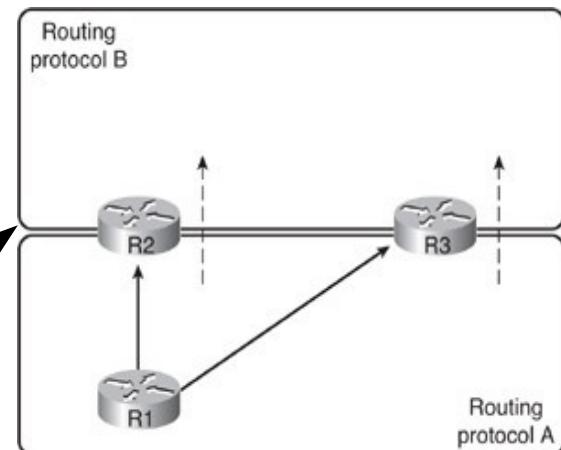


- As a recommended practice, limit unnecessary L3 routing peer adjacencies by configuring the ports toward Layer 2 access switches as passive.
 - ◆ Suppress the advertising of routing updates.
 - ◆ If a distribution switch does not receive L3 routing updates from a potential peer on a specific interface, it does not form a neighbor adjacency with the potential peer across that interface.



Route Redistribution

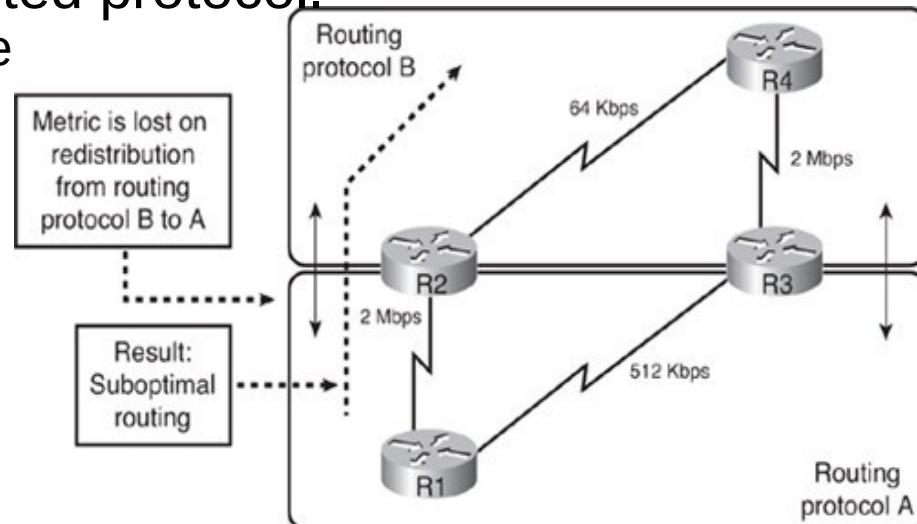
- Domains with different routing protocols can exchange routes.
 - ◆ This is called route redistribution.
 - ◆ One-way redistribution - Redistributes only the networks learned from one routing protocol into the other routing protocol.
 - Uses a default or static route so that devices in that other part of the network can reach the first part of the network
 - ◆ Two-way redistribution - Redistributes routes between the two routing processes in both directions
 - ◆ Static routes can also be redistributed.



Redistribution Issues

- Lost metric from redistributed protocol.

- ◆ It is not possible to achieve an optimal overall routing.



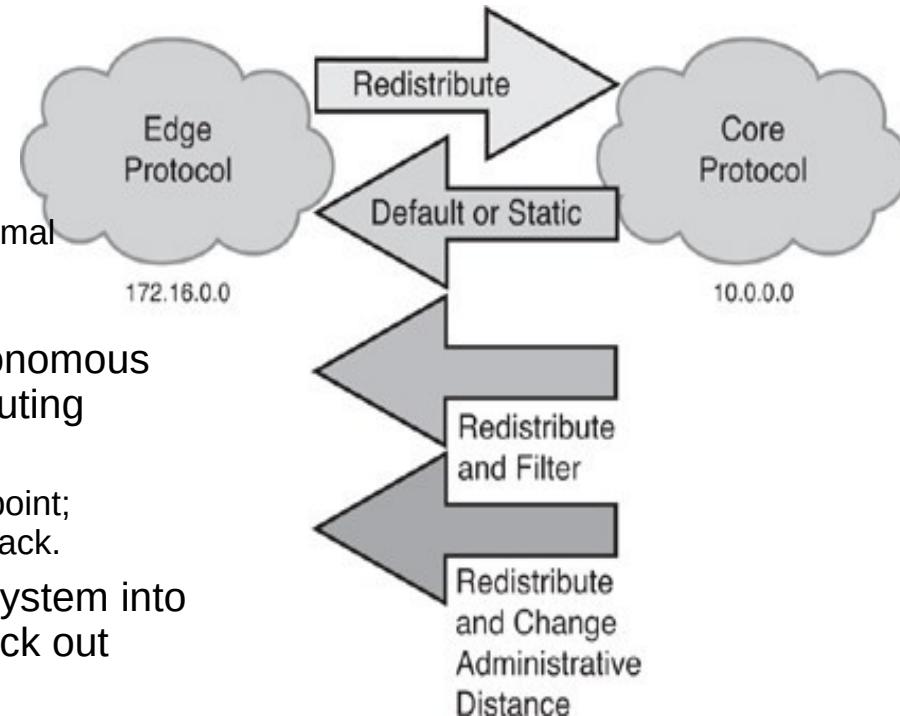
- Preventing Routing Loops in a Redistribution Environment.

- ◆ Safest way to perform redistribution is to redistribute routes in only one direction, on only one boundary router within the network.
 - ◆ However, that this results in a single point of failure in the network.
 - ◆ If redistribution must be done in both directions or on multiple boundary routers, the redistribution should be tuned to avoid problems such as suboptimal routing and routing loops.

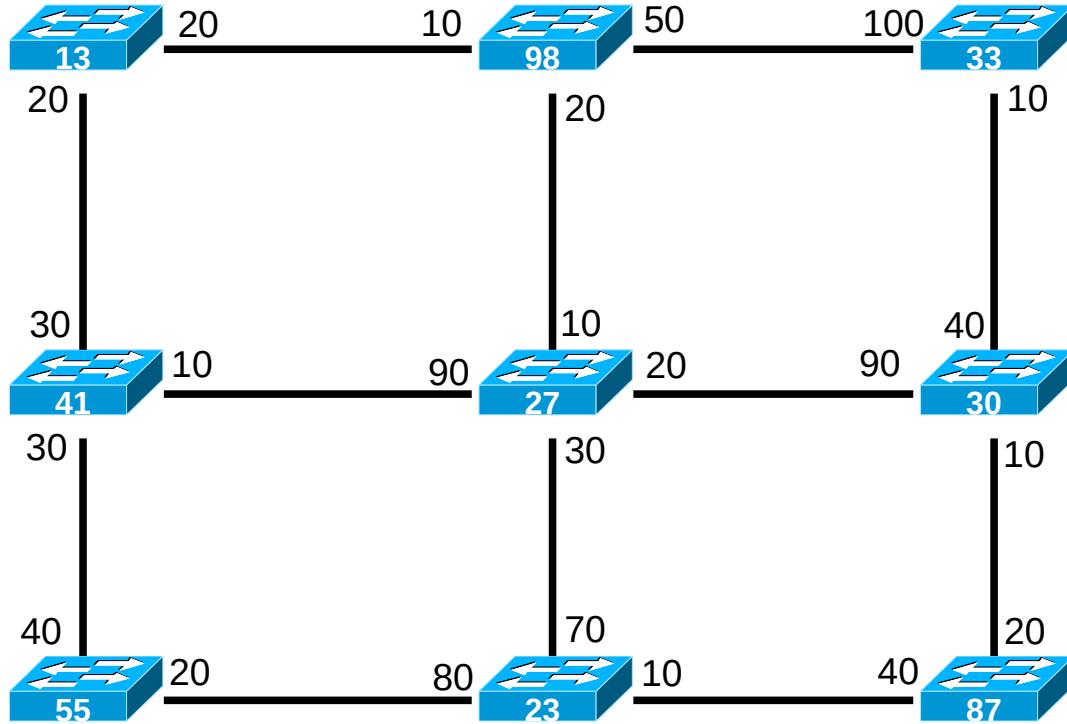


Redistribution Techniques

- Redistribute a default route from the core autonomous system into the edge autonomous system, and redistribute routes from the edge routing protocols into the core routing protocol.
 - This technique helps prevent route feedback, suboptimal routing, and routing loops.
- Redistribute multiple static routes about the core autonomous system networks into the edge autonomous system, and redistribute routes from the edge routing protocols into the core routing protocol.
 - This method works if there is only one redistribution point; multiple redistribution points might cause route feedback.
- Redistribute routes from the core autonomous system into the edge autonomous system with filtering to block out inappropriate routes.
 - For example, when there are multiple boundary routers, routes redistributed from the edge autonomous system at one boundary router should not be redistributed back into the edge autonomous system from the core at another redistribution point.
- Redistribute all routes from the core autonomous system into the edge autonomous system, and from the edge autonomous system into the core autonomous system, and then modify the administrative distance associated with redistributed routes so that they are not the selected routes when multiple routes exist for the same destination.

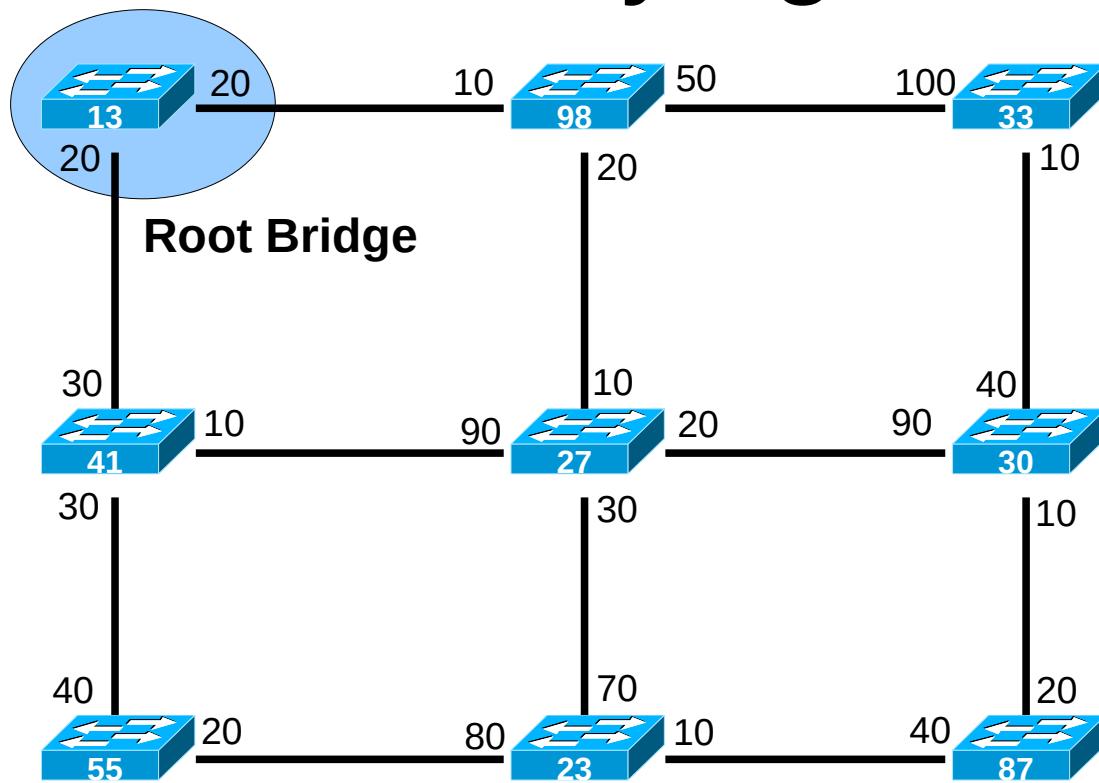


How to determine the Spanning-tree



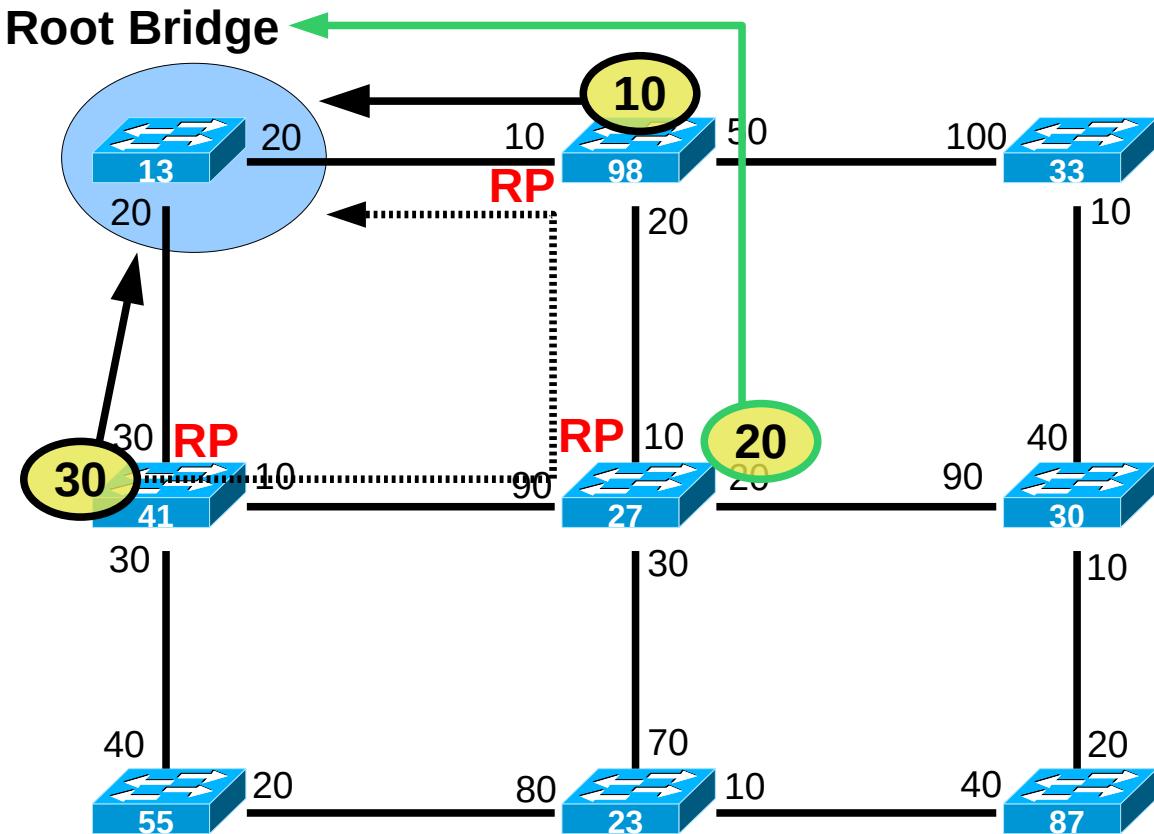
1. Identify the **root bridge**
2. Identify “**root path costs**” and **root ports**
3. Identify **designated bridges** and **designated ports**

Identifying the root bridge



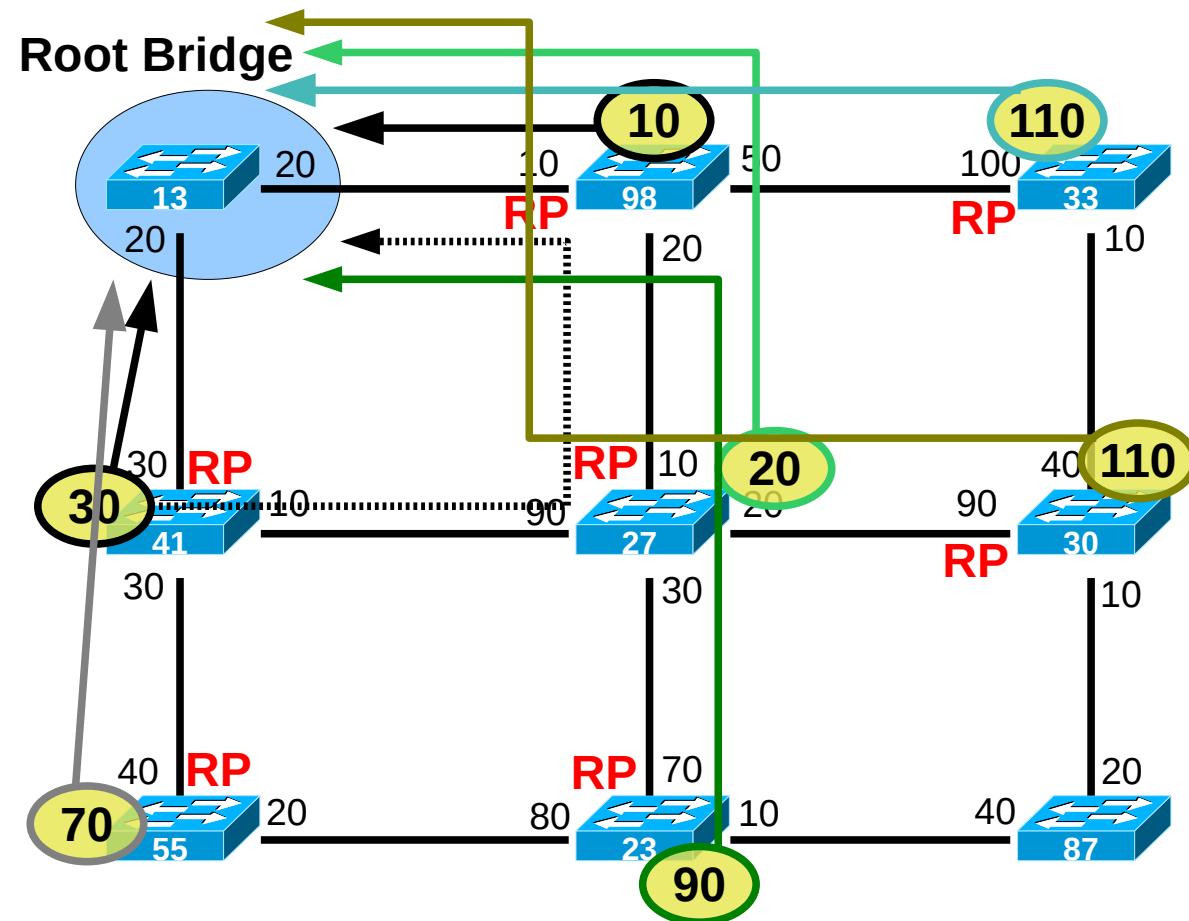
- The root bridge is the one with the lowest ID
 - ID = priority + MAC
 - The bridge with the lowest priority will be the root
 - For equal priorities it's necessary to analyze the bridge's MAC address

“Root Path Costs” and root ports



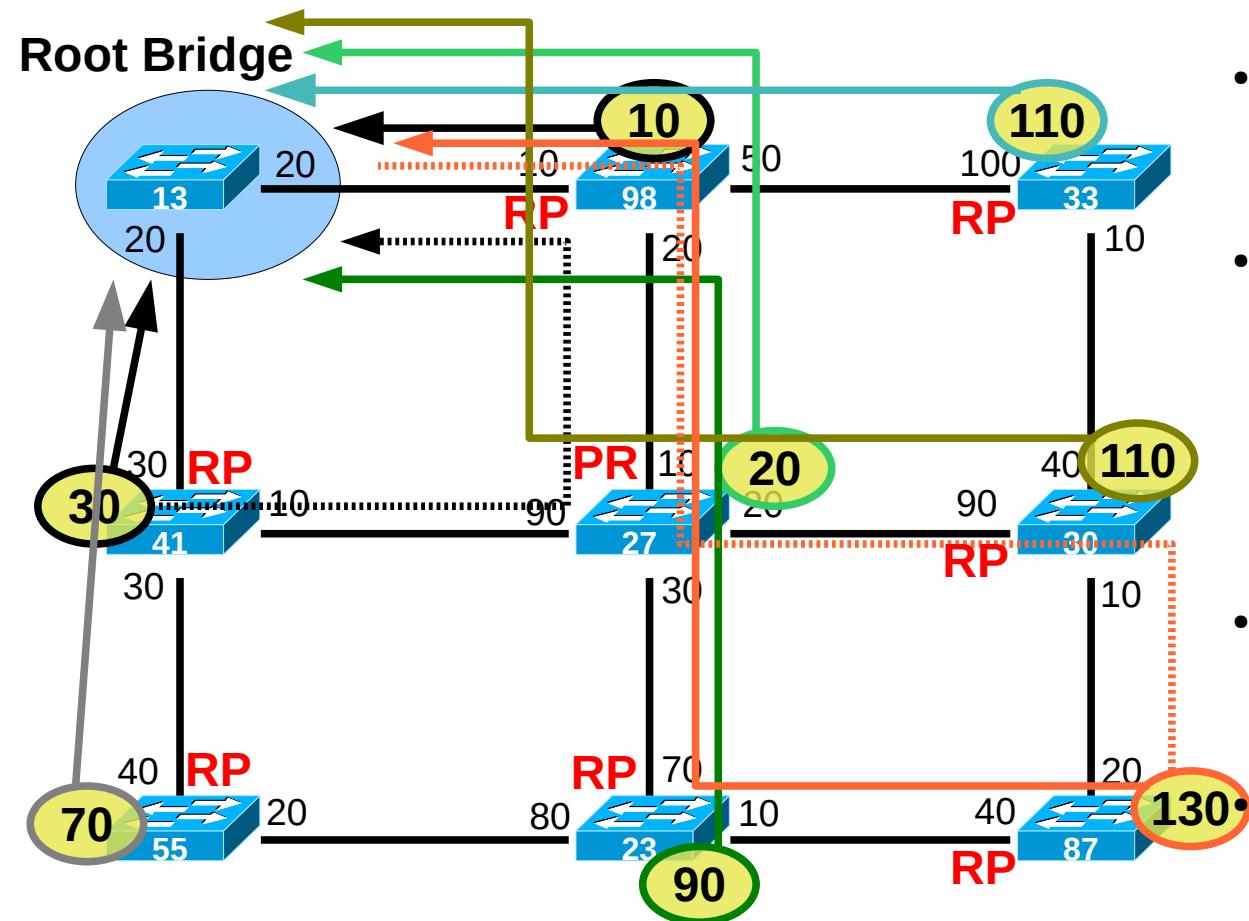
- “Root Path Cost” (RPC) is the cost of the path between a bridge and the root.
- The cost is given by the sum of all “output” ports' costs in the path to the root.
 - In each bridge, it's given by the sum of the RPC of the neighbor bridge plus the cost of the port that connects to that neighbor bridge.
- For paths with the same cost, it's chosen the one announced by the bridge with the lowest ID.
- Tip: start the RPC calculations from the bridges “closer” to the root.

“Root Path Costs” and root ports



- “Root Path Cost” (RPC) is the cost of the path between a bridge and the root.
- The cost is given by the sum of all “output” ports' costs in the path to the root.
 - In each bridge, it's given by the sum of the RPC of the neighbor bridge plus the cost of the port that connects to that neighbor bridge.
- For paths with the same cost, it's chosen the one announced by the bridge with the lowest ID.
- Tip: start the RPC calculations from the bridges “closer” to the root.

“Root Path Costs” and root ports

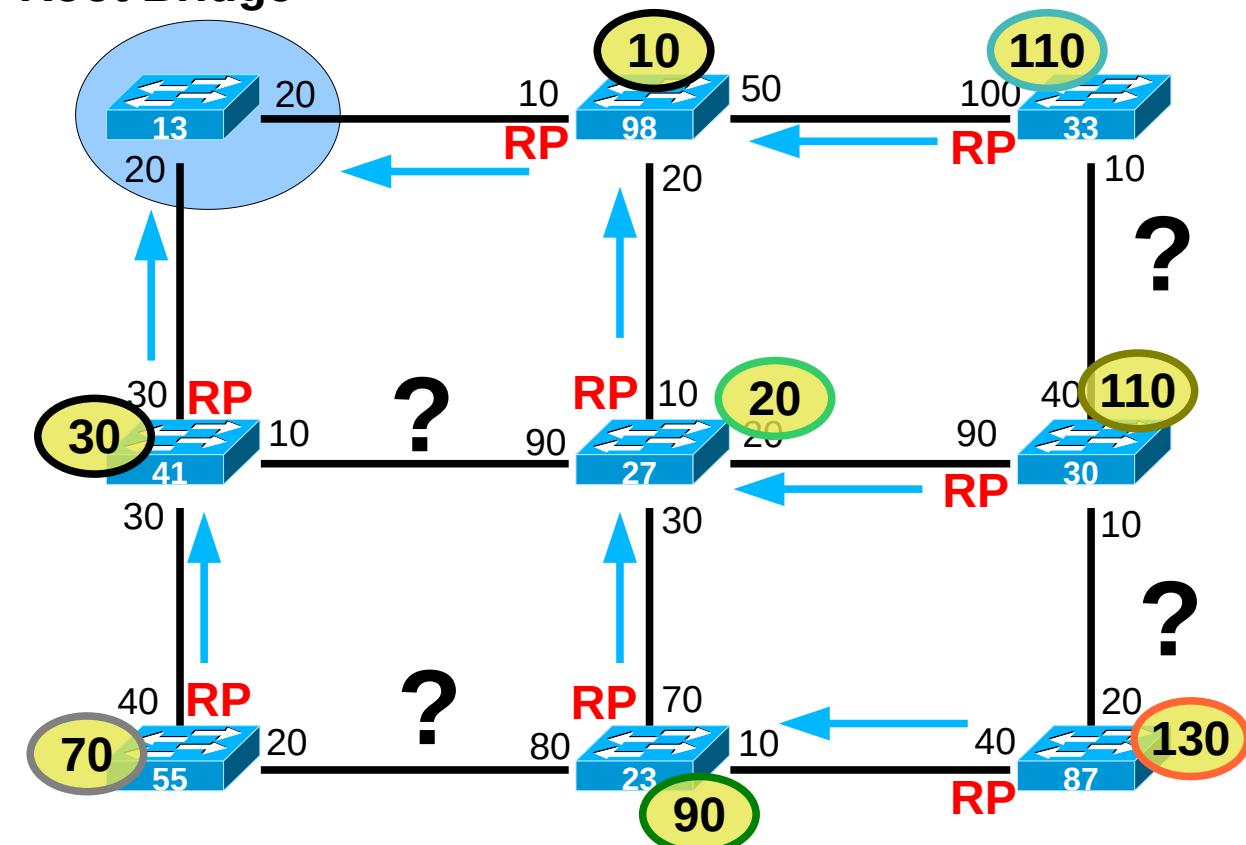


- “Root Path Cost” (RPC) is the cost of the path between a bridge and the root.
- The cost is given by the sum of all “output” ports' costs in the path to the root.
 - In each bridge, it's given by the sum of the RPC of the neighbor bridge plus the cost of the port that connects to that neighbor bridge.
- For paths with the same cost, it's chosen the one announced by the bridge with the lowest ID.

Tip: start the RPC calculations from the bridges “closer” to the root.

Designated bridges and ports

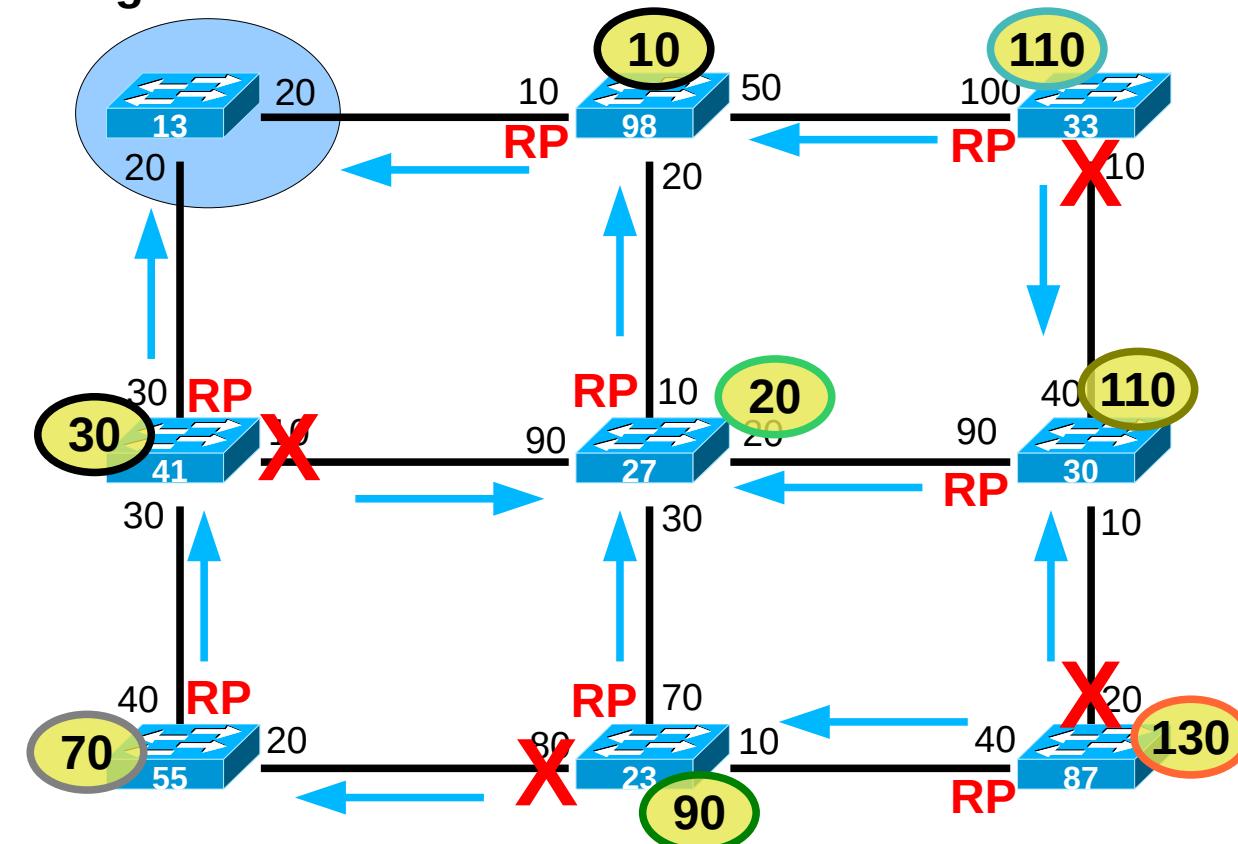
Root Bridge



- A Ethernet segment's designated bridge is the one with:
 - The lowest RPC
 - For equal costs, the one with the lowest ID
- The root bridge is always the designated bridge of all Ethernet segments connected to it.
- In a Ethernet segment that belongs to the minimum cost path, the designated bridge is always the one that provides that path to the root.

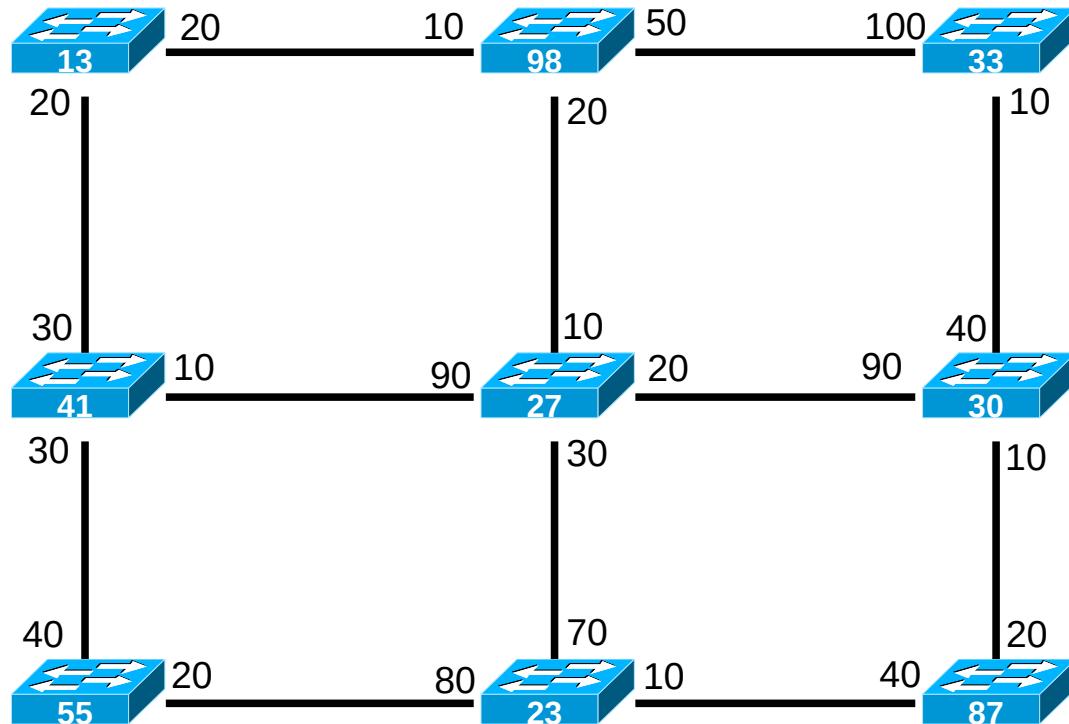
Designated bridges and ports

Bridge raíz



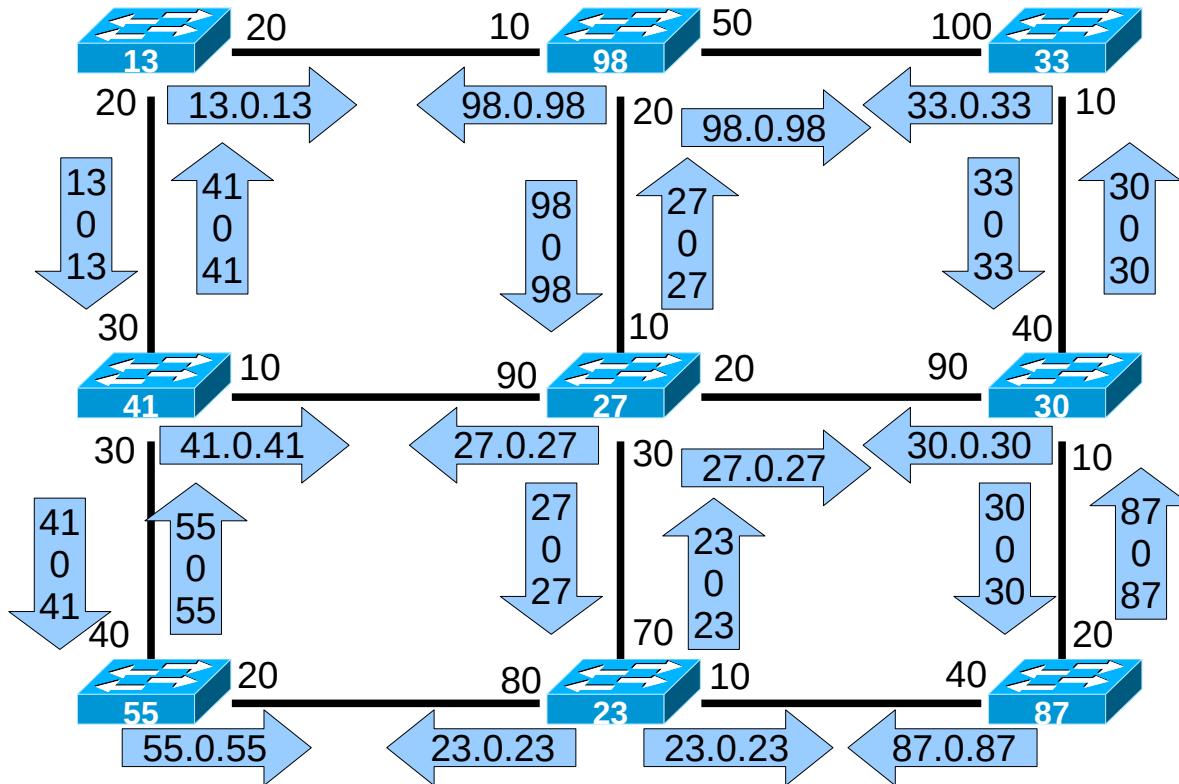
- A Ethernet segment's designated bridge is the one that has:
 - The lowest Root Path Cost
 - For equal costs, the lowest ID
 - Ethernet segment 41-27:
Designated bridge 27
 - Lowest cost
 - Ethernet segment 30-33:
Designated bridge 30
 - Same cost, lowest ID
 - Ethernet segment 23-55:
Designated bridge 55
 - Lowest cost
 - Ethernet segment 30-87:
Designated bridge 30
 - Lowest cost

ST Construction - Messages Exchange



- At start, all bridges assume to be the root bridge.
- Send Conf-BPDUs to all connected Ethernet segments.

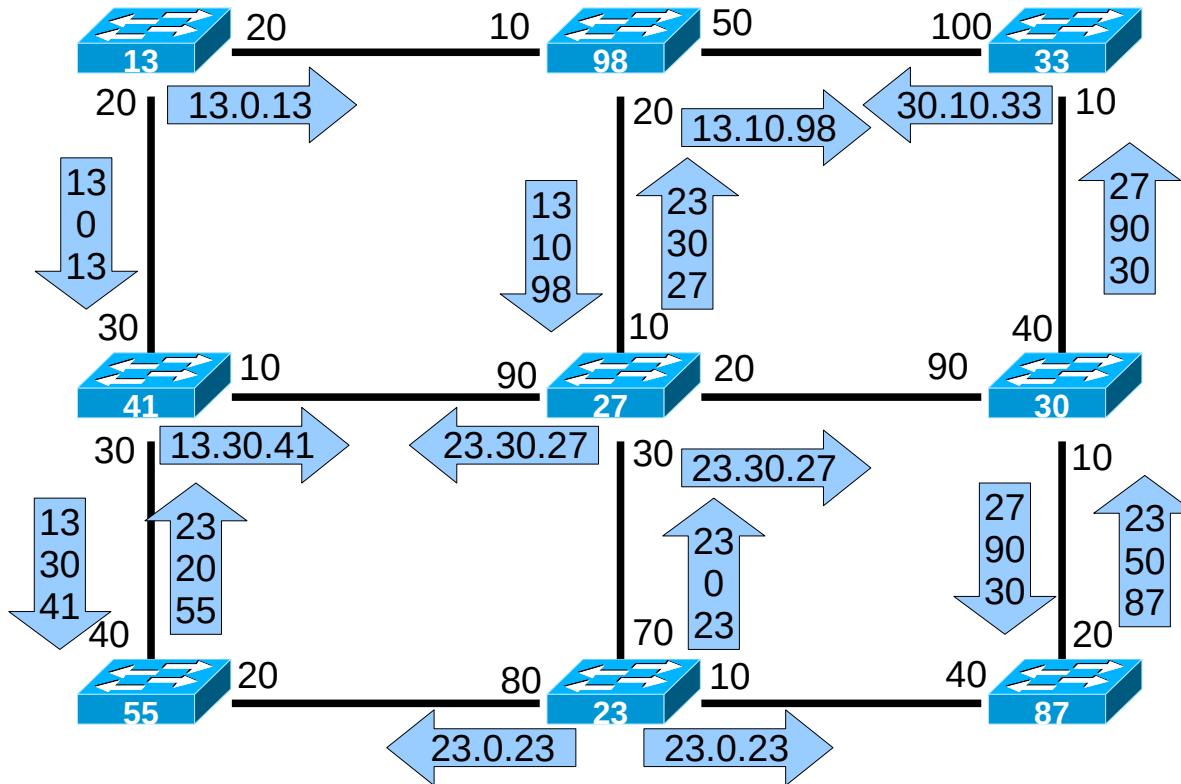
ST Construction - Messages Exchange



- At start, all bridges assume to be the root bridge.
- Send Conf-BPDUs to all connected Ethernet segments.
 - 13 remains root
 - 98 accepts 13 as root (cost 10)
 - 33 accepts 30 as root (cost 10)
 - 41 accepts 13 as root (cost 30)
 - 27 accepts 23 as root (cost 30)
 - 30 accepts 27 as root (cost 90)
 - 55 accepts 23 as root (cost 20)
 - 23 remains root
 - 87 accepts 23 as root (cost 50)

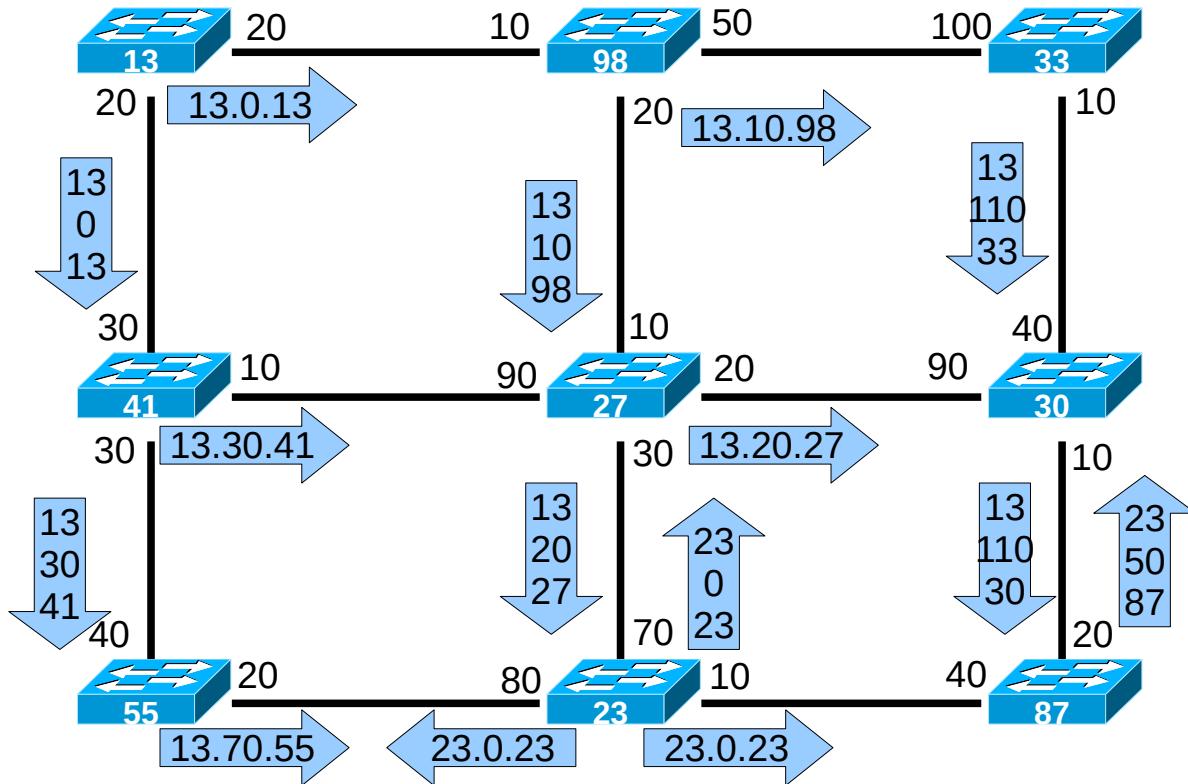
Raíz.Custo.ID

ST Construction - Messages Exchange



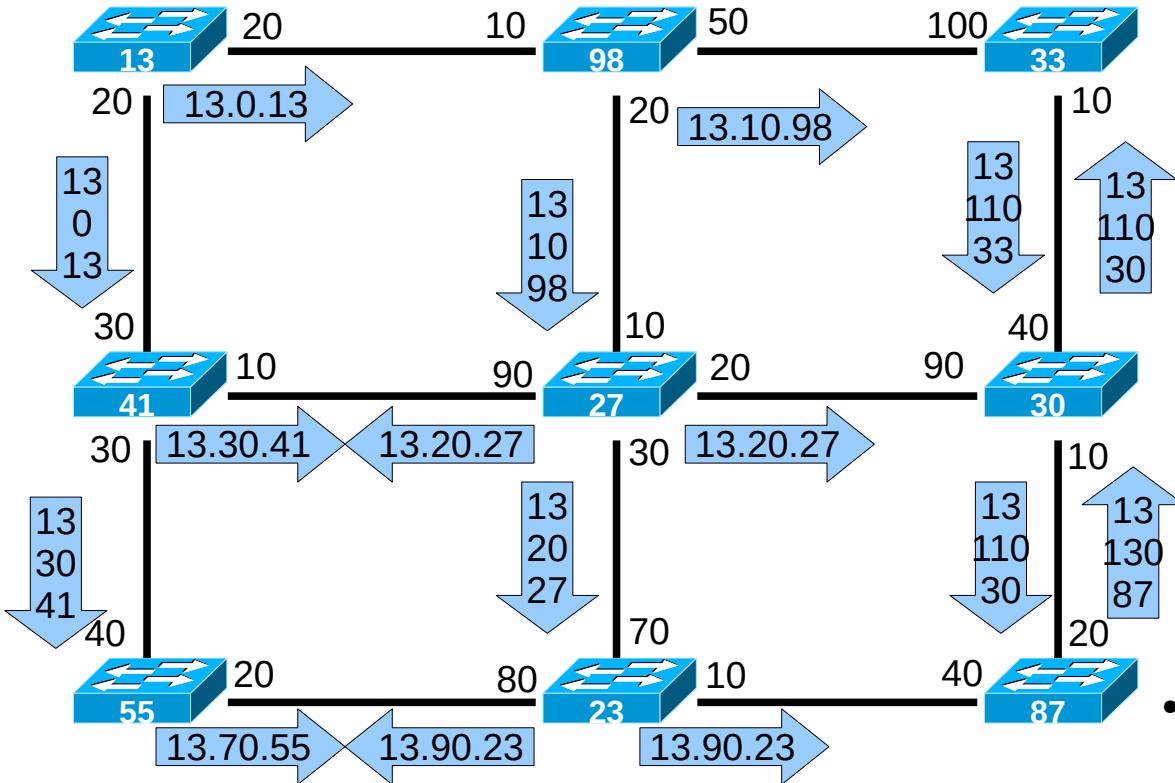
- Bridges only send Conf-BPDUs to the Ethernet segments where they are designated.
 - 13 remains root
 - 98 accepts 13 as root (cost 10)
 - 33 accepts 13 as root (cost 110 – via 98)
 - 41 accepts 13 as root (cost 30)
 - 27 accepts 13 as root (cost 20 – via 98)
 - 30 accepts 23 as root (cost 120 – via 27)
 - 55 accepts 13 as root (cost 70 – via 41)
 - 23 remains root
 - 87 accepts 23 as root (cost 40)

ST Construction - Messages Exchange



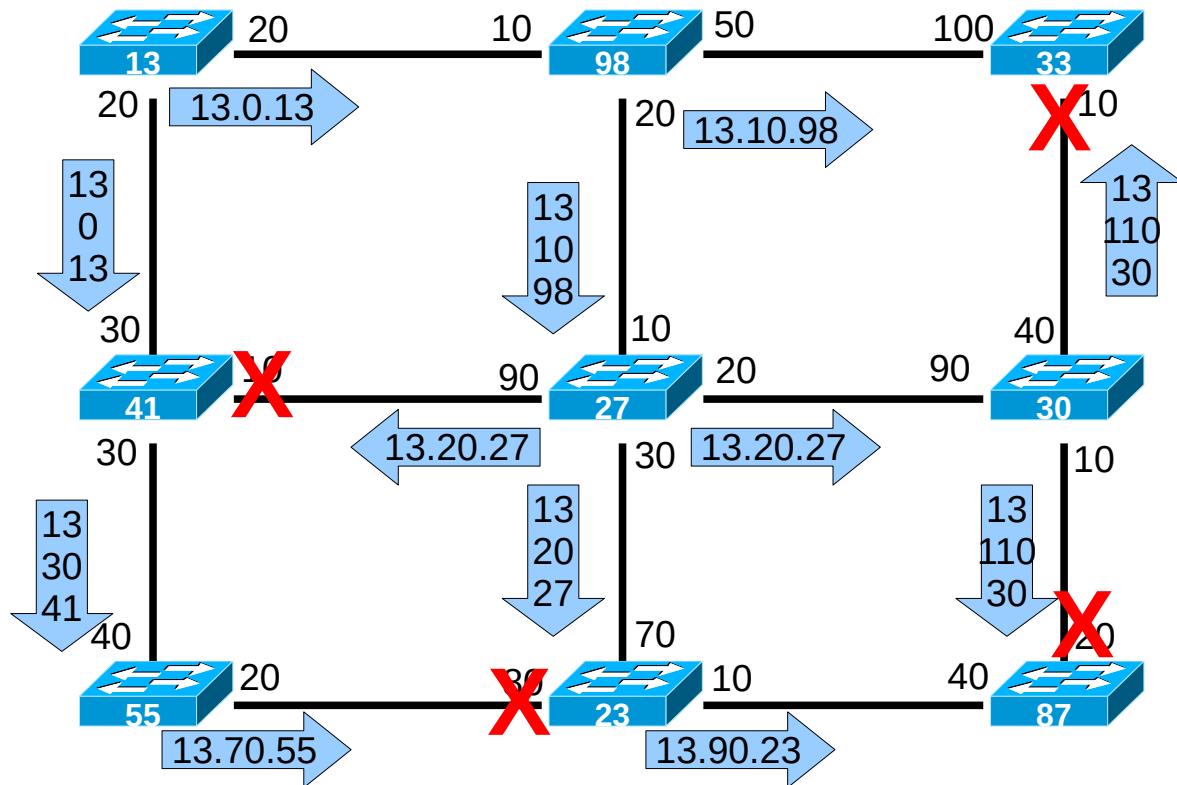
- Bridges only send Conf-BPDUs to the Ethernet segments where they are designated.
 - 13 remains root
 - 98 accepts 13 as root (cost 10)
 - 33 accepts 13 as root (cost 110 – via 98)
 - 41 accepts 13 as root (cost 30)
 - 27 accepts 13 as root (cost 20 – via 98)
 - 30 accepts 13 as root (cost 110 – via 27)
 - 55 accepts 13 as root (cost 70 – via 41)
 - 23 accepts 13 as root (cost 90 – via 27)
 - 87 accepts 13 as root (cost 130 – via 30)

ST Construction - Messages Exchange



- Bridges only send Conf-BPDUs to the Ethernet segments where they are designated.
 - 13 remains root
 - 98 accepts 13 as root (cost 10)
 - 33 accepts 13 as root (cost 110 – via 98)
 - 41 accepts 13 as root (cost 30)
 - 27 accepts 13 as root (cost 20 – via 98)
 - 30 accepts 13 as root (cost 110 – via 27)
 - 55 accepts 13 as root (cost 70 – via 41)
 - 23 accepts 13 as root (cost 90 – via 27)
 - 87 accepts 13 as root (cost 130 – via 23)
 - Cost 130 – via 23 is preferred because the bridge ID is lower (23<30)
- The designated bridge of a Ethernet segment is chosen according with the best messages sent.
 - Ethernet segment 41-27: designated bridge 27 (lowest cost)
 - Ethernet segment 55-23: designated bridge 55 (lowest cost)
 - Ethernet segment 30-33: designated bridge 30 (Lowest bridge ID)
 - Ethernet segment 30-87: designated bridge 30 (lowest cost)

ST Construction - Messages Exchange



Error Control and Detection

Fundamentos de Redes

**Mestrado Integrado em
Engenharia de Computadores e Telemática
DETI-UA**



universidade de aveiro

deti.ua.pt

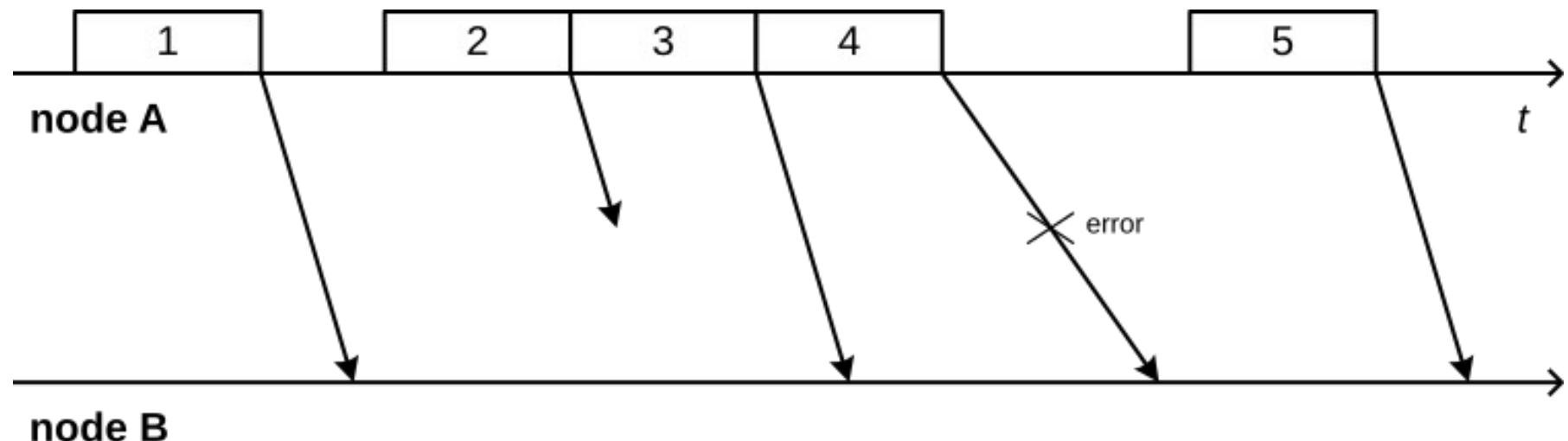
Error Control in a Communication Channel

- Causes:
 - Corrupted packets (data received with errors);
 - Lost packets;
 - Packets received out of order.
- Solutions:
 - Sender and receiver must be able to coordinate between them to retransmit a lost or corrupted packet.
 - Retransmit protocols: ARQ (Automatic Repeat reQuest).



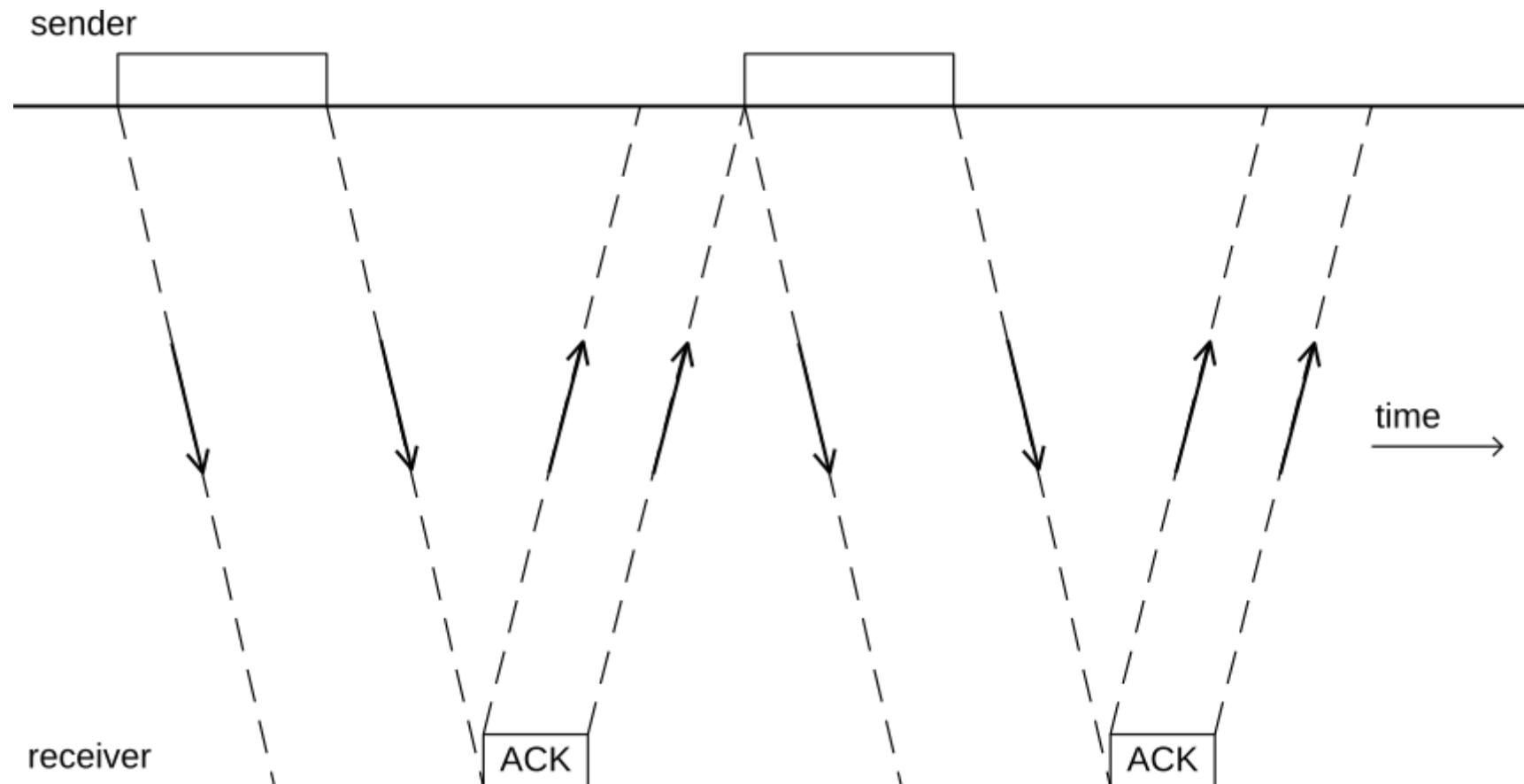
Error Control Assumptions

- Errors are always detected;
- Frames/packets may have variable (limited) delays;
- Some frames/packets may be lost.



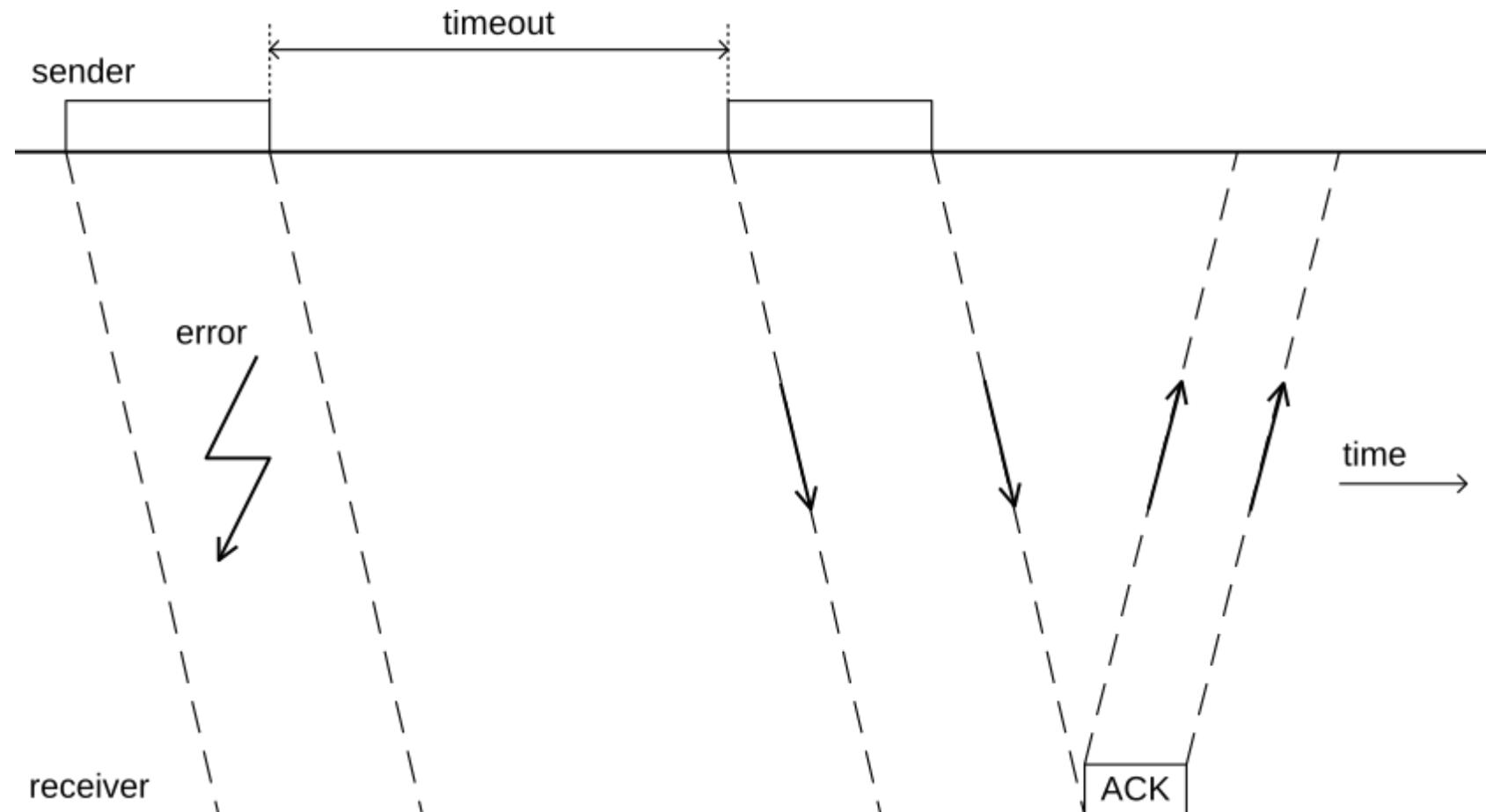
Stop-and-Wait (SW)

- Operation without errors:



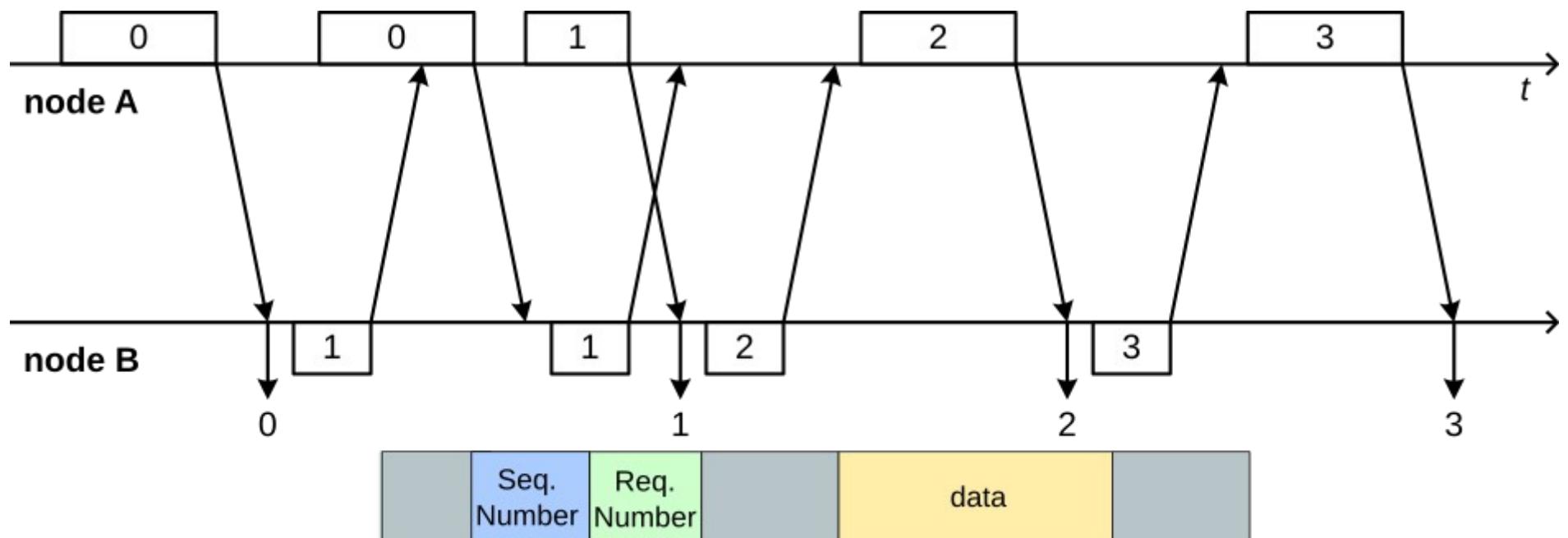
Stop-and-Wait (SW)

- Operation with error and recover:



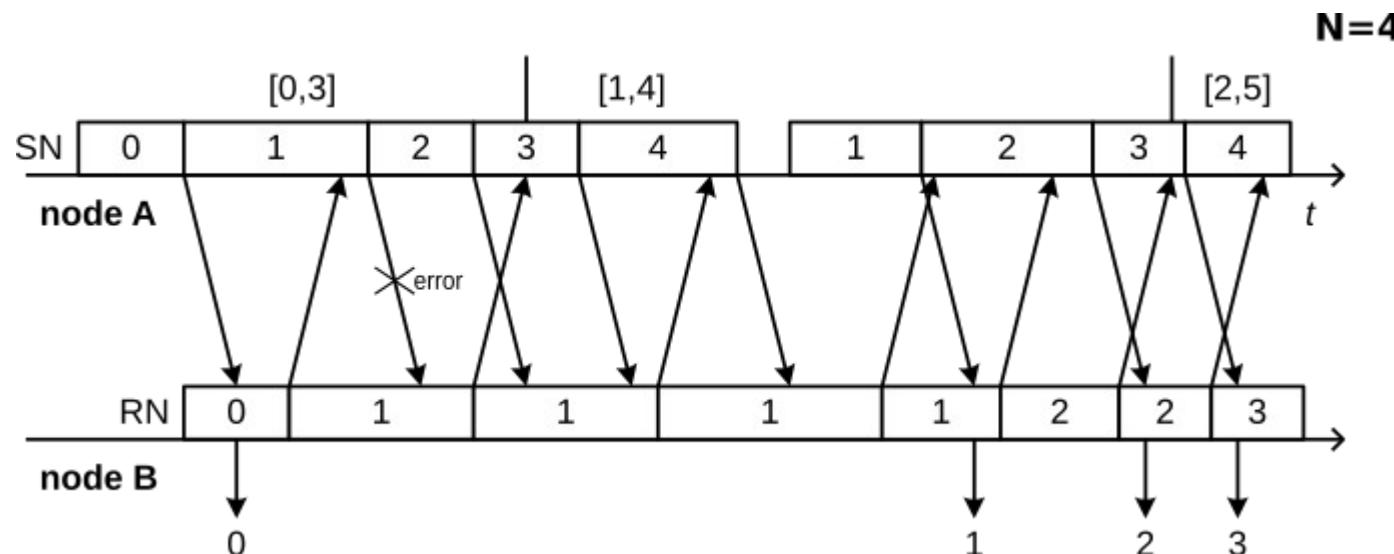
Stop-and-Wait (SW)

- Messages have a sequence number and a request number.
 - The sequence number identifies the messages sent.
 - The request number allow the receiver to notify the sender about the received message and next message expected.



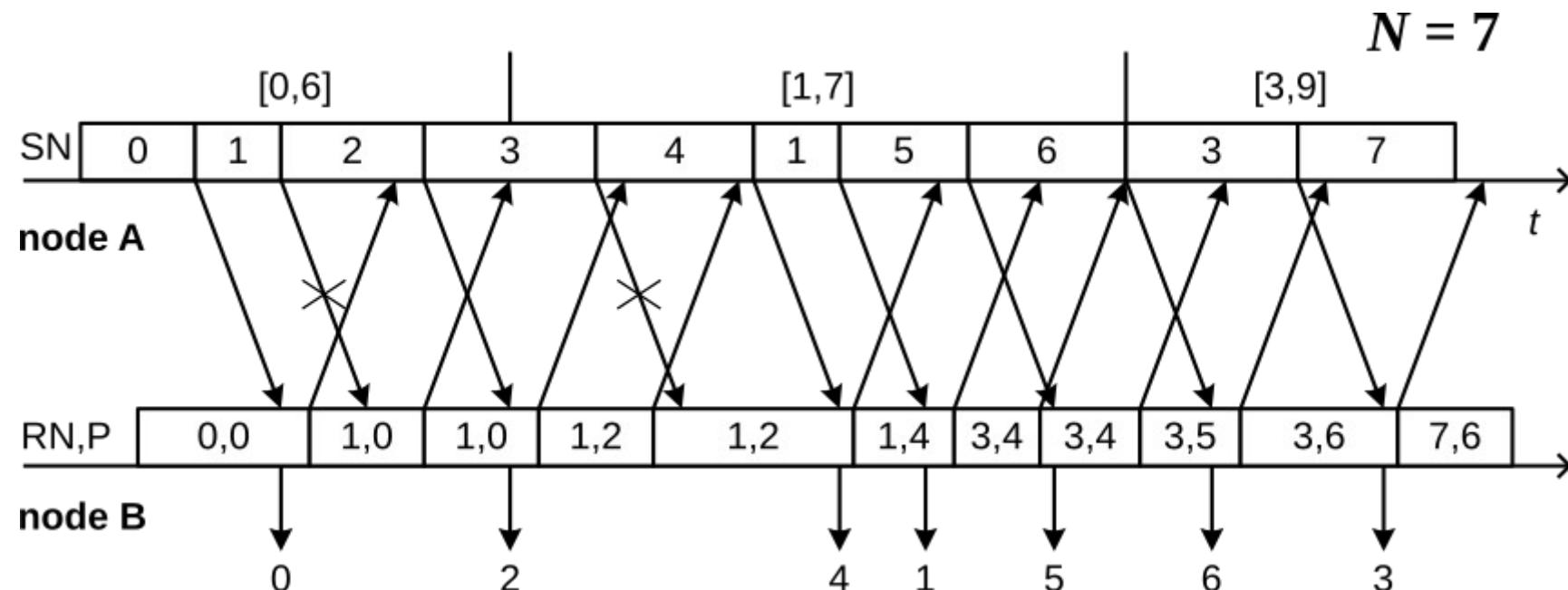
Go-Back-N (GB-N)

- The sender is allowed to send more than one message before receiving the confirmation of reception of the previous messages.
 - Window with size $N - N$ defines the number of messages that may be sent without confirmation of reception of the first.
- The sender, after a timeout, resends the first message without confirmation of reception and all of the following.
- The receiver, after receiving all messages until a sequence number equal to n ($SN=n$), accepts only the message with $SN=n+1$.
 - Drops all the others.
- The receiver, responds with a request number (RN) to identify the next message not received.
 - Implicitly, indicates that all messages up to RN were correctly received.



Selective Repeat (SR)

- The sender is allowed to send more than one message before receiving the confirmation of reception of the previous messages.
 - Window with size $N - N$ defines the number of messages that may be sent without confirmation of reception of the first.
- The sender, after a timeout, resends a message without confirmation of reception.
- The receiver, after receiving all messages until a sequence number equal to n ($SN=n$), accepts to receive all messages with $SN > n$ and $SN \leq N+n$.
- The receiver, responds with a request number (RN) to identify the next message not received, and the higher payload/message number correctly received (P).



Error Detection

- Performed at multiple Layers.

- Nowadays the (new protocols) tendency is to be performed only at the physical/link layer and application layer.
 - Has a performance impact.
 - E.g., IPv4 supports it, IPv6 does not.

- Methods

- CRC (Cyclic-Redundancy Check)
 - Based on the theory of cyclic error-correcting codes.
 - Adds a fixed-length check value to messages.
 - Requires of a generator polynomial.
 - The binary message with the CRC field is divided by the generator polynomial.
 - No errors imply remainder equal to zero.
- Checksums and hash/digest values
 - Generated by a predefined function that receives as input all message bits.
 - Value is appended to message.



UDP & TCP

Fundamentos de Redes

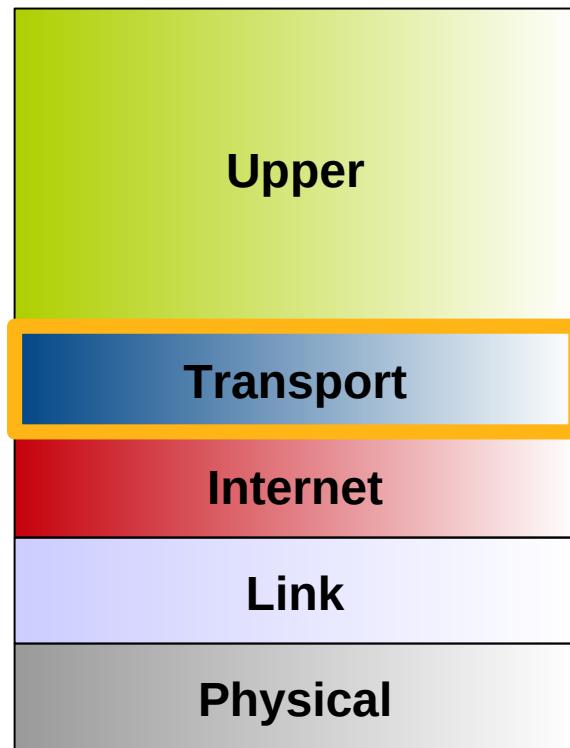
**Mestrado Integrado em
Engenharia de Computadores e Telemática
DETI-UA**



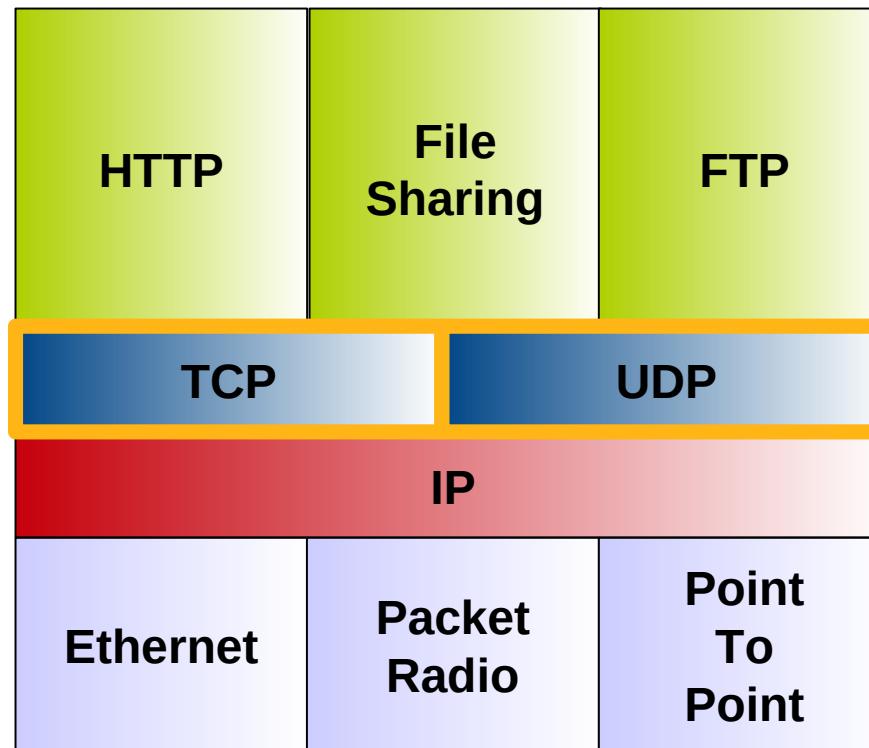
universidade de aveiro

deti.ua.pt

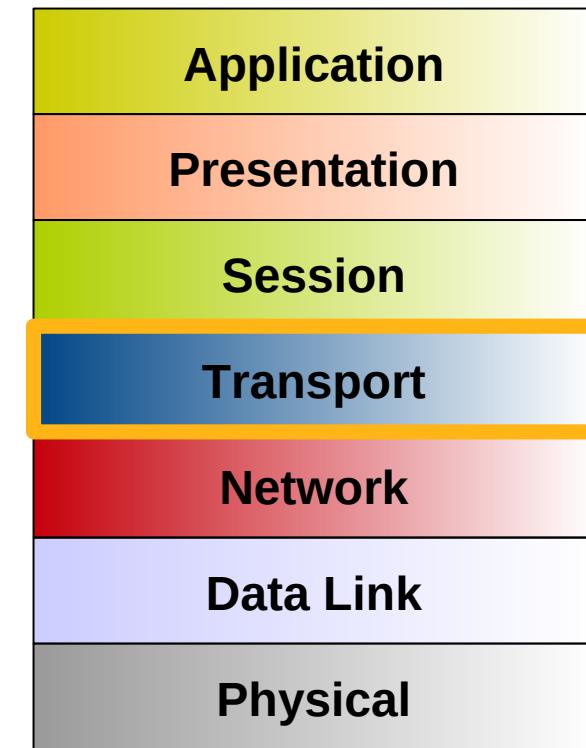
TCP/IP Reference Models



TCP/IP

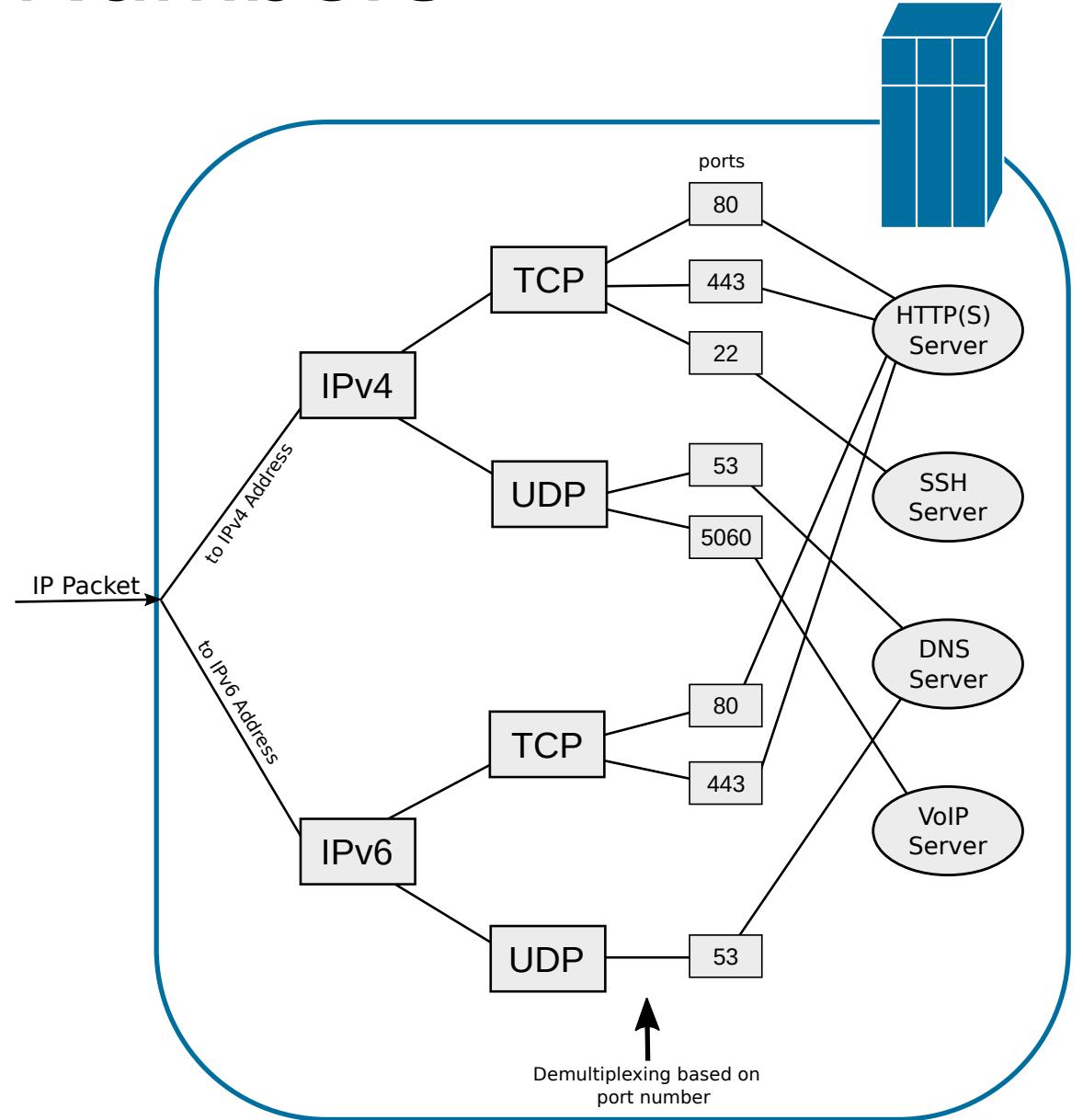


OSI



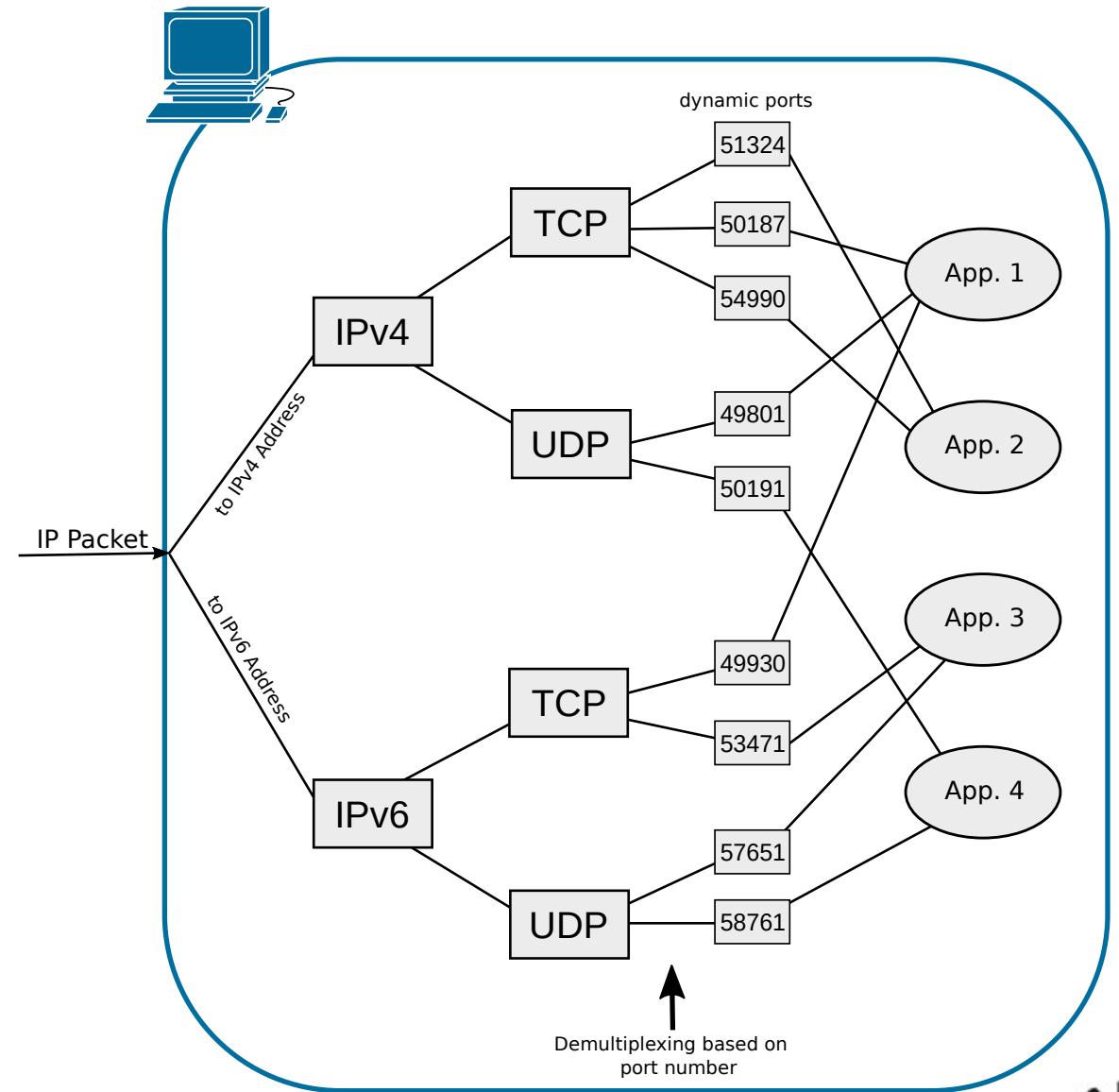
Port Numbers

- The packet destination IP address identifies a target host on the network.
- The transport protocol identifies the type of communication.
- The port number identifies the application running in the target host.
 - ◆ Chosen by the application/service.
 - ◆ Non-dynamic ports.
 - ◆ Each application/service may use more than one port.
 - ◆ The OS assures the assignment of different port numbers to each application.



Ephemeral/Dynamic Ports

- Ephemeral/Dynamic ports are used to identify a client application in a client-server communication.
- The Internet Assigned Numbers Authority (IANA) suggests the range 49152 to 65535.
- Randomly assigned by OS.



Well Known Port Numbers

| Decimal Keyword | Protocol | Description |
|---------------------|----------|-----------------------------------|
| 20 FTP-DATA | TCP | File Transfer Protocol (dados) |
| 21 FTP-CONTROL | TCP | File Transfer Protocol (controlo) |
| 22 SSH | TCP | Secure Shell (SSH) service |
| 25 SMTP | TCP | Simple Mail Transport Protocol |
| 67,68 BOOTP | UDP | Bootstrap Protocol (DHCP) |
| 53 DNS | UDP/TCP | Domain Name System |
| 69 TFTP | UDP | Trivial File Transfer Protocol |
| 80 HTTP | TCP | Hypertext Transfer Protocol |

- Many Internet IP services were the subject of study by IETF that proposed adequate support protocols.
- For these services, IETF together with the protocol specification proposed also a number (or numbers) to be used by that service at the server side.
- E.g., for protocol HTTP IETF recommends the usage of port 80. Therefore, all Web Browsers use port 80 as default for HTTP accesses.

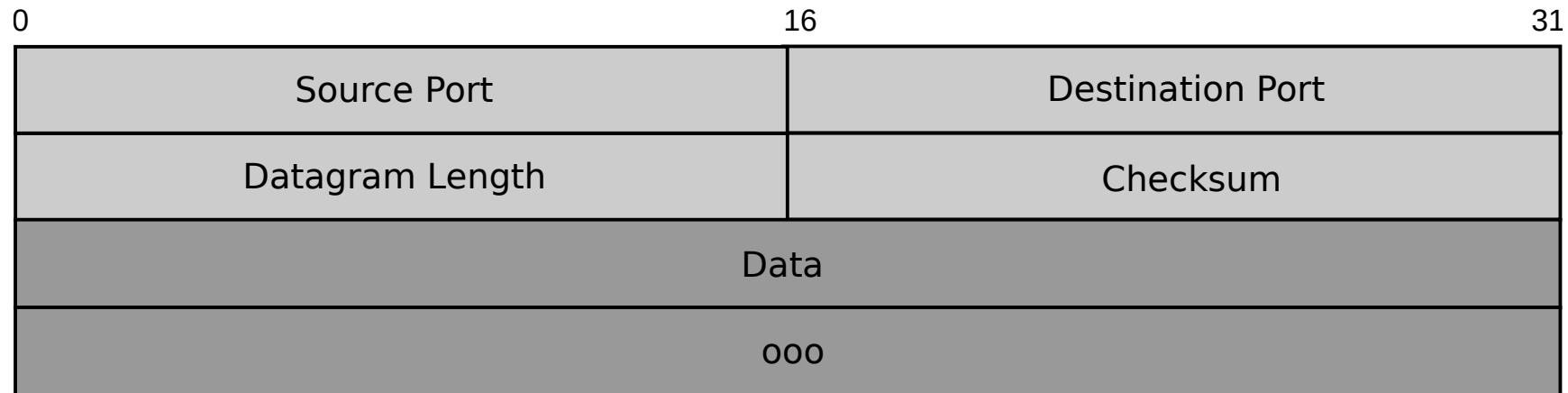


User Datagram Protocol (UDP)

- Provides a data transport service with the performance characteristics offered by the IP network.
- Provides exchange of data between individual applications, and not only between hosts, with the introduction of a port identifier field.
- Does not provide any mechanism to recover from lost messages.
- Does not provide any mechanism to order received data.
 - ◆ Must rely on information at the application level.
- Allows to send data to multiple destinations simultaneously (point-to-multipoint communications).



UDP Datagram (Header+Data)

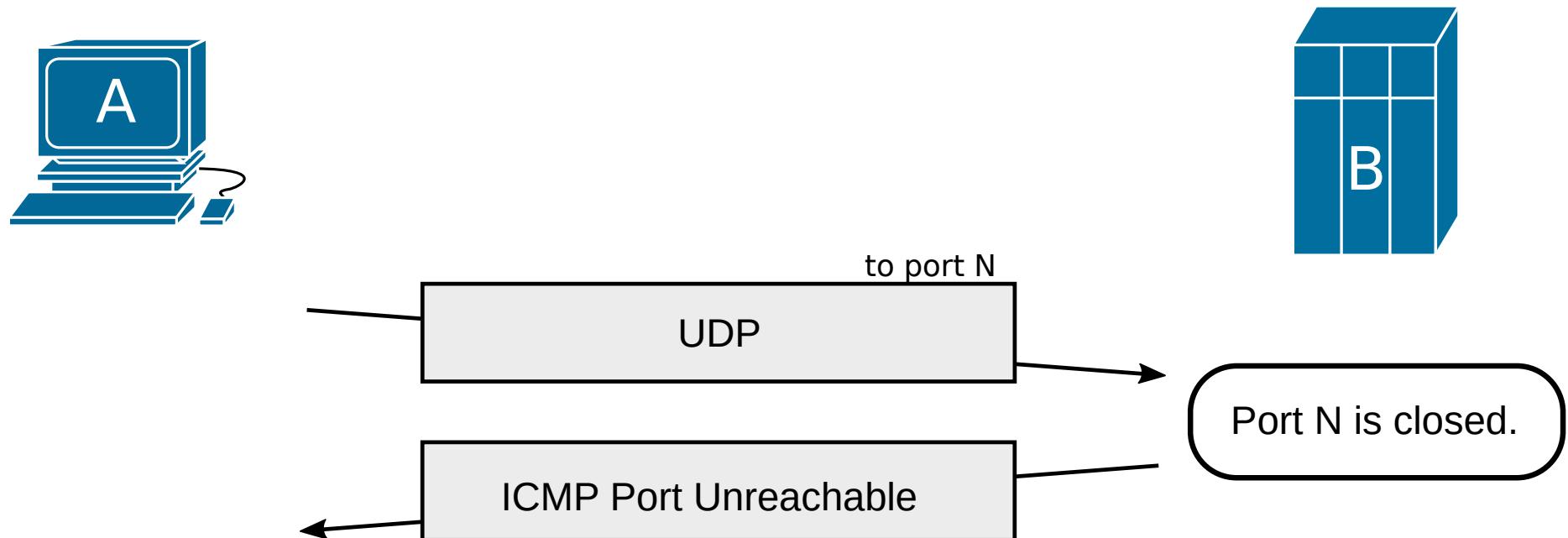


- Source Port (2 bytes): defines the port number assign/chosen by the sender application. This field is optional, if not used should be field with zeros.
- Destination Port (2 bytes): defines the port number assign/chosen by the receiver application.
- Datagram Length (2 bytes): defines the size,in bytes, of the datagram (header+data).
- Checksum (2 bytes): used for data error detection and validation at the end-points. This field is optional, if not used should be field with zeros.
 - ◆ The checksum us calculated based on the UDP datagram and a pseudoheader IP (IP protocol identifier, source and destination IP addresses, and length of the IP datagram).
 - ◆ Can be used to verify if the end-points are the correct ones.



UDP Closed Port

- When an UDP packet arrives to a host, but the UDP port is not open (no application listening):
 - The host responds with a packet ICMP *port unreachable*.

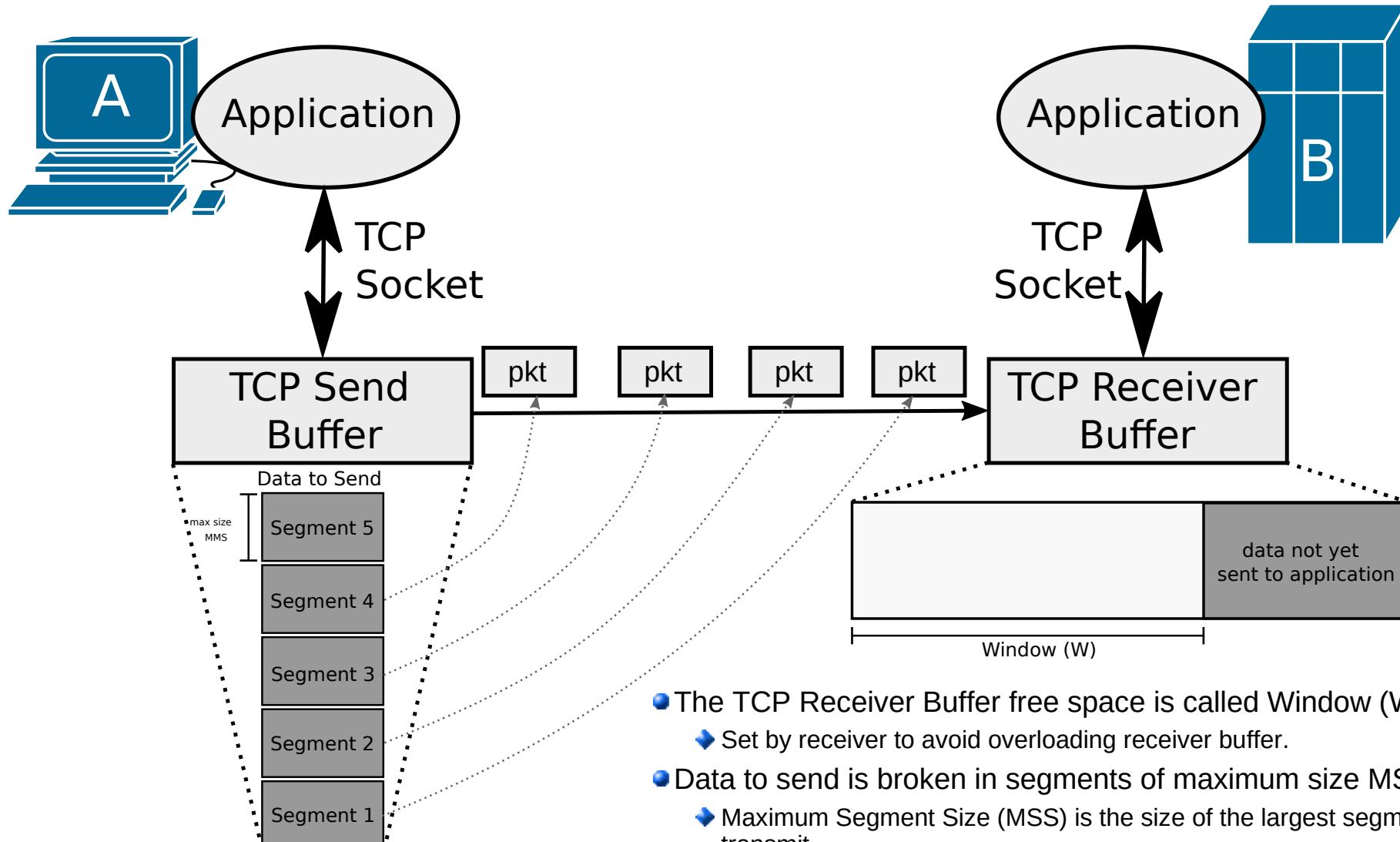


Transmission Control Protocol (TCP)

- Provides a reliable data transport service.
 - ◆ Data is received by the destination application without any losses and in order.
- Is connection-oriented protocol.
 - ◆ End-points establish a logical channel, to which are assigned applications' identifiers and the memory resources required to have a reliable transmission of data.
 - ◆ This connection is called Session.
- It is bi-directional.
 - ◆ Both end-points can send and receive data using the same logical channel.
- Traditional TCP supports only point-to-point connections.
 - ◆ See: Multipath TCP on last slide.
- Provides mechanisms to establish and terminate the connection.
- Network congestion and/or temporary lack of connectivity result in variable delays and consequent losses of packets (at transit or by timeout).
 - ◆ TCP includes algorithm that allows to efficiently react in this scenario.



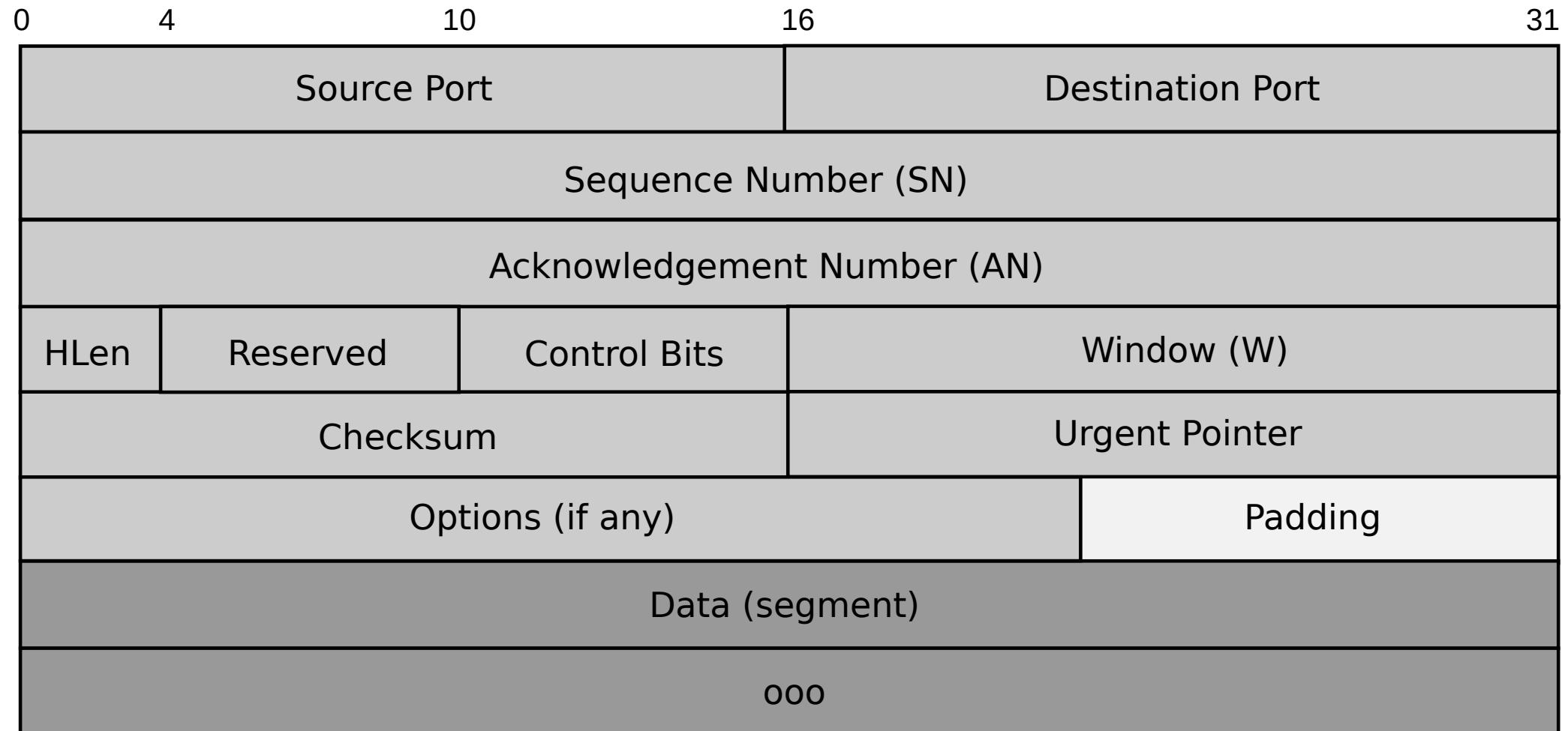
TCP Buffers and Receiver Window



- The TCP Receiver Buffer free space is called Window (W).
 - ◆ Set by receiver to avoid overloading receiver buffer.
- Data to send is broken in segments of maximum size MSS.
 - ◆ Maximum Segment Size (MSS) is the size of the largest segment that a sender can transmit.
 - ◆ Defined by local configuration.
- The sender has an estimation of the available space at the receiver buffer (RWND) given by the reported remote buffer space (W) minus the already sent bytes not yet acknowledged (NACK).



TCP Packet Header (1)



TCP Packet Header (2)

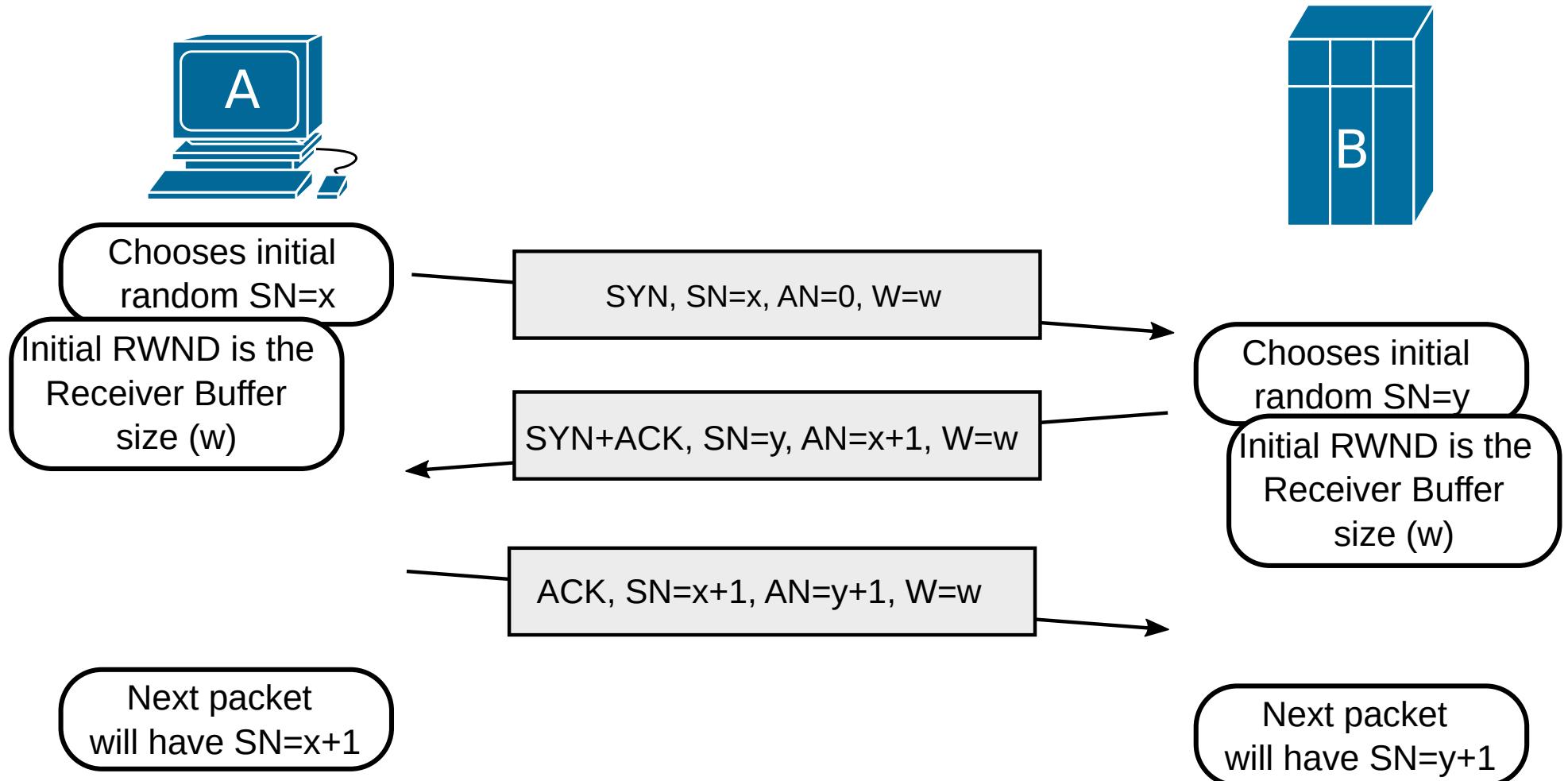
- The TCP Header has a variable length.
 - ◆ The field *Options* may contain additional fields.
- *Source Port* and *Destination Port* fields define the respective end-point ports.
- The *Sequence Number* defines the index of the first data byte in this segment.
- The *Acknowledge Number* defines the value of the next sequence number.
 - ◆ Acknowledges the good reception of data until byte with index (Acknowledge Number-1)
- *HLen* defines the size of the TCP header in bytes.
 - ◆ When *Options* are present, the field *Padding* is used to extend the header size to a multiple of 32 bytes.
- *Window (W)* field defined the number of data bytes the sender of this segment is willing to accept. Free space on the reception buffer.
 - ◆ The estimation of the remote receiver free buffer is called Receiver Window (RWND), as is given by the reported remote buffer space (W) minus the already sent bytes not yet acknowledged (NACK).
- *Control Bits* field is a binary set of flags
 - ◆ URG: Urgent Pointer is a valid field.
 - ◆ ACK: Acknowledgement is a valid field.
 - ◆ PSH: Data requires Push (receiver should push immediately data from TCP buffer to application).
 - ◆ RST: Connection Reset.
 - ◆ SYN: Sincronize Sequence Number.
 - ◆ FIN : Source closing connection.
- *Urgent Pointer* field defines the portion of data that should be considered urgent.
 - ◆ Not used by modern protocols.
- *Checksum* is used to detect errors.



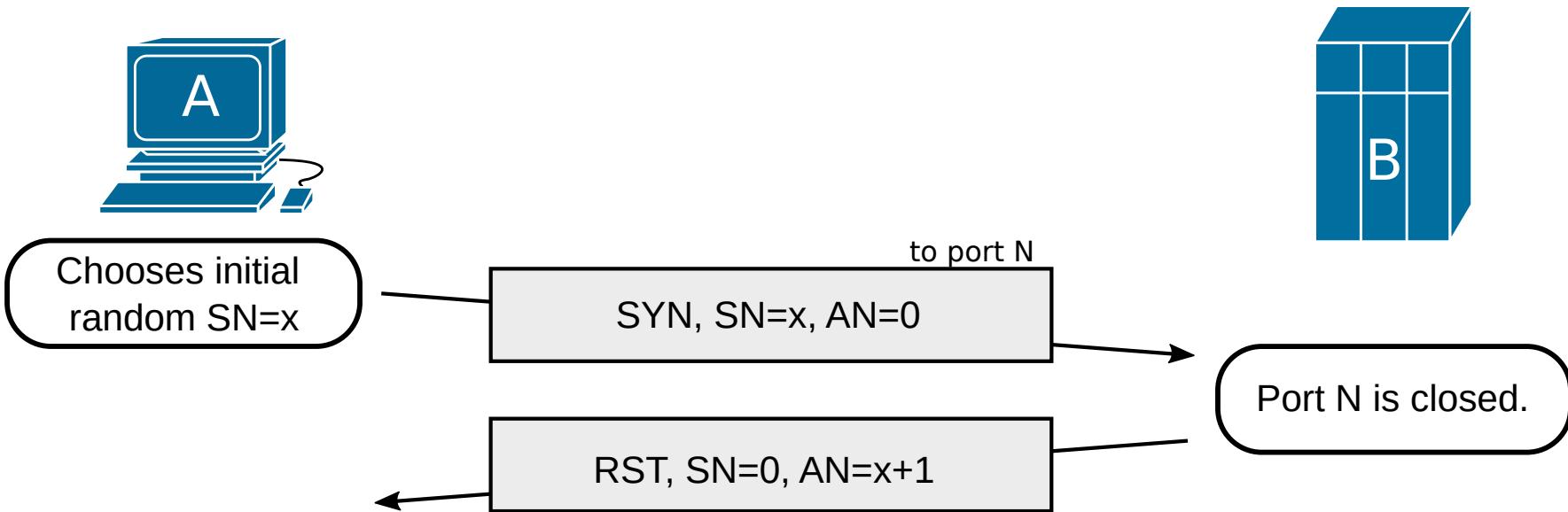
Establishement of a TCP Connection

- 3-Way Handshake.

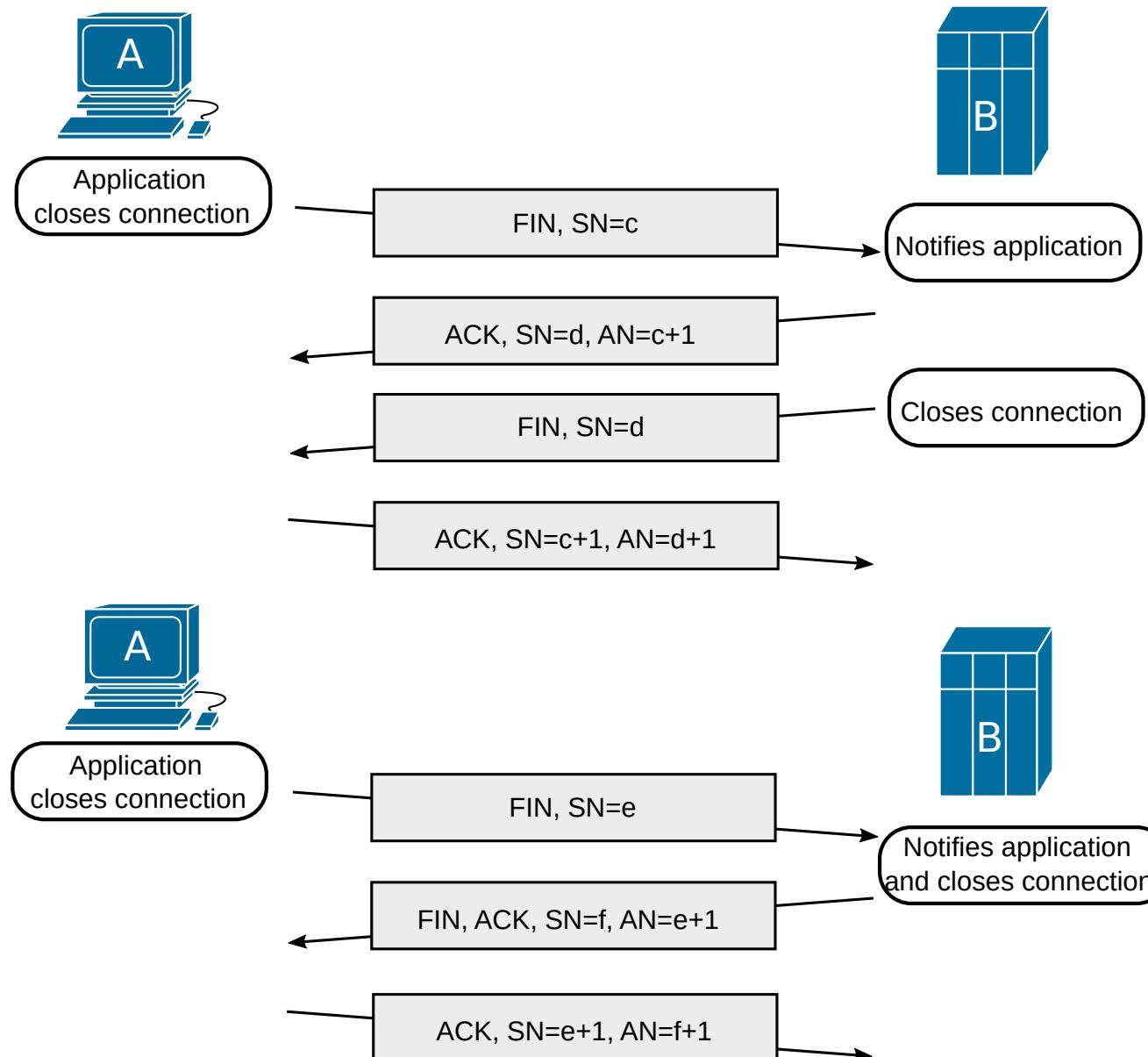
- Synchronizes the both end-points initial *Sequence Numbers* (SYN), and acknowledges it (ACK flag).



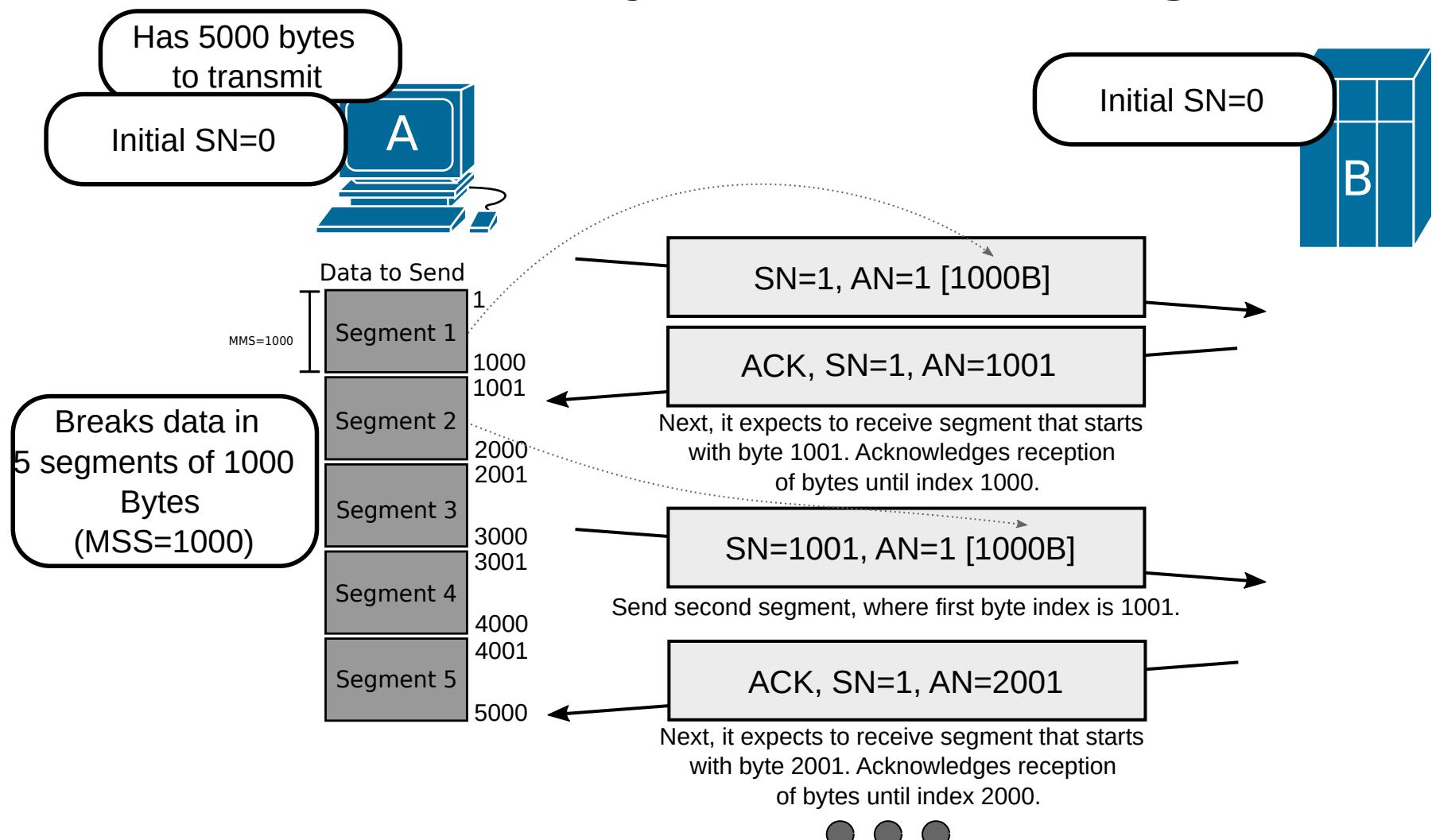
Establishement of a TCP Connection to a Closed Port



Closing a TCP Connection



Data Delivery Acknowledgment

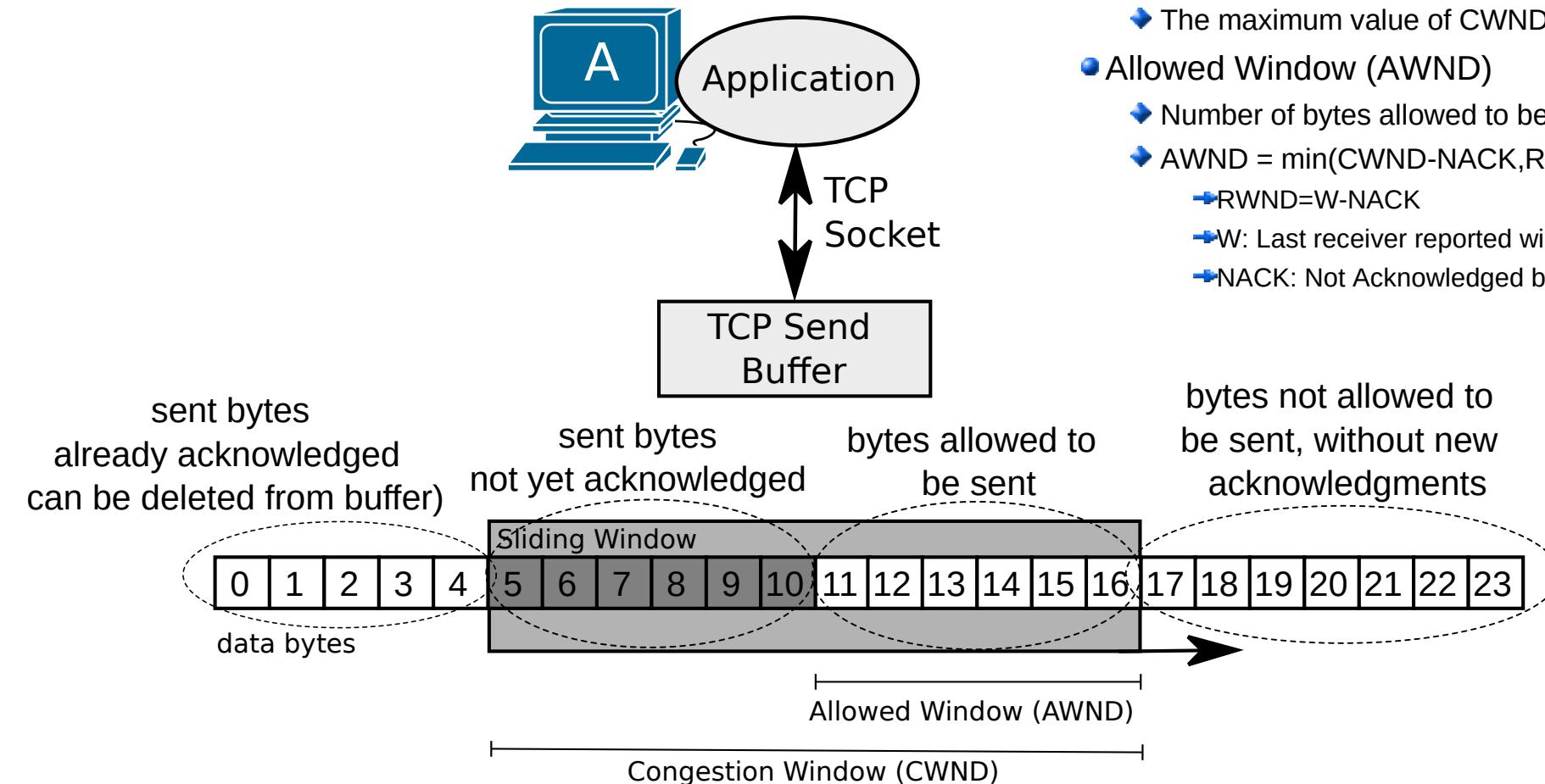


- Sender can (usually) send more than one packet before receiving the ACK.
 - ◆ Depends on the congestion window (next slide).
- Both end-points can send data.
 - ◆ A packet with a data segment, can acknowledge the reception of data a segment received from the other end-point.



TCP Congestion Control (1)

- Uses a sliding window to determine the number of packets/bytes the sender is allowed to transmit.



- Congestion Window (CWND)

- Set by sender to avoid overloading network.
- The maximum value of CWND is RWND.

- Allowed Window (AWND)

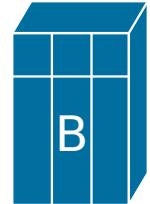
- Number of bytes allowed to be sent.
- $\text{AWND} = \min(\text{CWND}-\text{NACK}, \text{RWND})$
- $\text{RWND}=\text{W-NACK}$
- W : Last receiver reported window.
- NACK : Not Acknowledged bytes.



TCP Congestion Control (2)



Both have:
MSS=1000 bytes (by configuration)
RWND=2500 (Window value after establishment)
CWND=RWND



CWND=2500, NACK=0, AWND=2500

CWND=2500, NACK=1000, AWND=1500

CWND=2500, NACK=2000, AWND=500

CWND=2500, NACK=2500 AWND=0

CWND=2500, NACK=1500 AWND=0 (RWND=W-NACK=0)

CWND=2500, NACK=500 AWND=0 (RWND=W-NACK=0)

CWND=2500, NACK=0 AWND=0 (RWND=0)

SN=x+1, AN=y+1 [1000B]
SN=x+1001, AN=y+1 [1000B]
SN=x+2001, AN=y+1 [500B]

ACK, SN=y+1, AN=x+1001, W=1500 [0B]
ACK, SN=y+1, AN=x+2001, W=500 [0B]
ACK, SN=y+1, AN=x+2501, W=0 [0B]

Transfers 2500B to application.

CWND=2500, NACK=0 AWND=2500 (RWND=2500)

CWND=2500, NACK=1000, AWND=1500

CWND=2500, NACK=1500, AWND=1000

SN=x+2501, AN=y+1 [1000B]
SN=x+3501, AN=y+1 [500B]

ACK, SN=y+1, AN=x+2501, W=2500 [0B]
ACK, SN=y+1, AN=x+3501, W=1500 [0B]
ACK, SN=y+1, AN=x+4001, W=500 [0B]



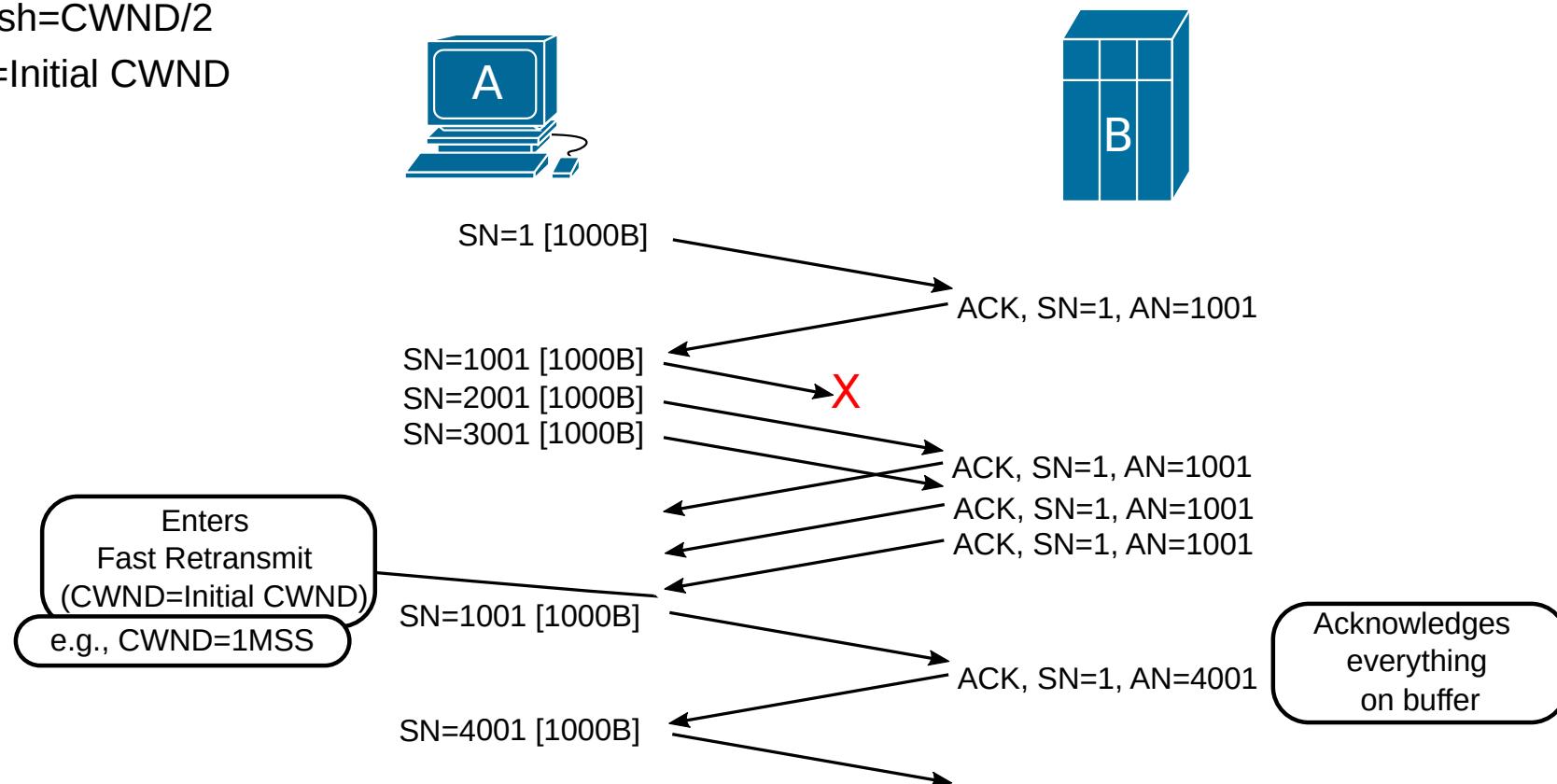
TCP Congestion Control (3)

- A packet with a data segment is considered lost when:
 - ◆ Timeout: After some time no ACK is received for that data segment
 - ◆ With Fast Retransmit/Recovery: After 3 or more duplicate ACKs are received for the previous data segment.
- When a packet is lost, TCP automatically decreases transmission rate to adapt to network conditions.
 - ◆ i.e., Decreases CWND size.
- When data delivery is acknowledged, TCP increases transmission rate.
 - ◆ i.e., Increases CWND size.
- The way the CWND size varies depend on the TCP Algorithms used:
 - ◆ Tahoe (Original TCP, 1988) uses:
 - ◆ Fast Retransmit, Slow Start, and Congestion Avoidance.
 - ◆ Reno (1990) uses:
 - ◆ Uses Fast Recovery, and Congestion Avoidance.
- At the beginning, the initial CWND value is usually 2, 3, 4, or 10 MSS and then the terminal starts the *Slow Start* with SSThresh=RWND.



TPC Fast Retransmit

- A segment is considered lost if 3 or more duplicate ACKs are received for the previous data segment.
 - ◆ Faster detection than waiting for a timeout.
 - ◆ Requires receiver to work.
- TCP retransmits immediately the lost segment.
- The TCP algorithm enters Slow-Start, with:
 - ◆ SSThresh=CWND/2
 - ◆ CWND=Initial CWND



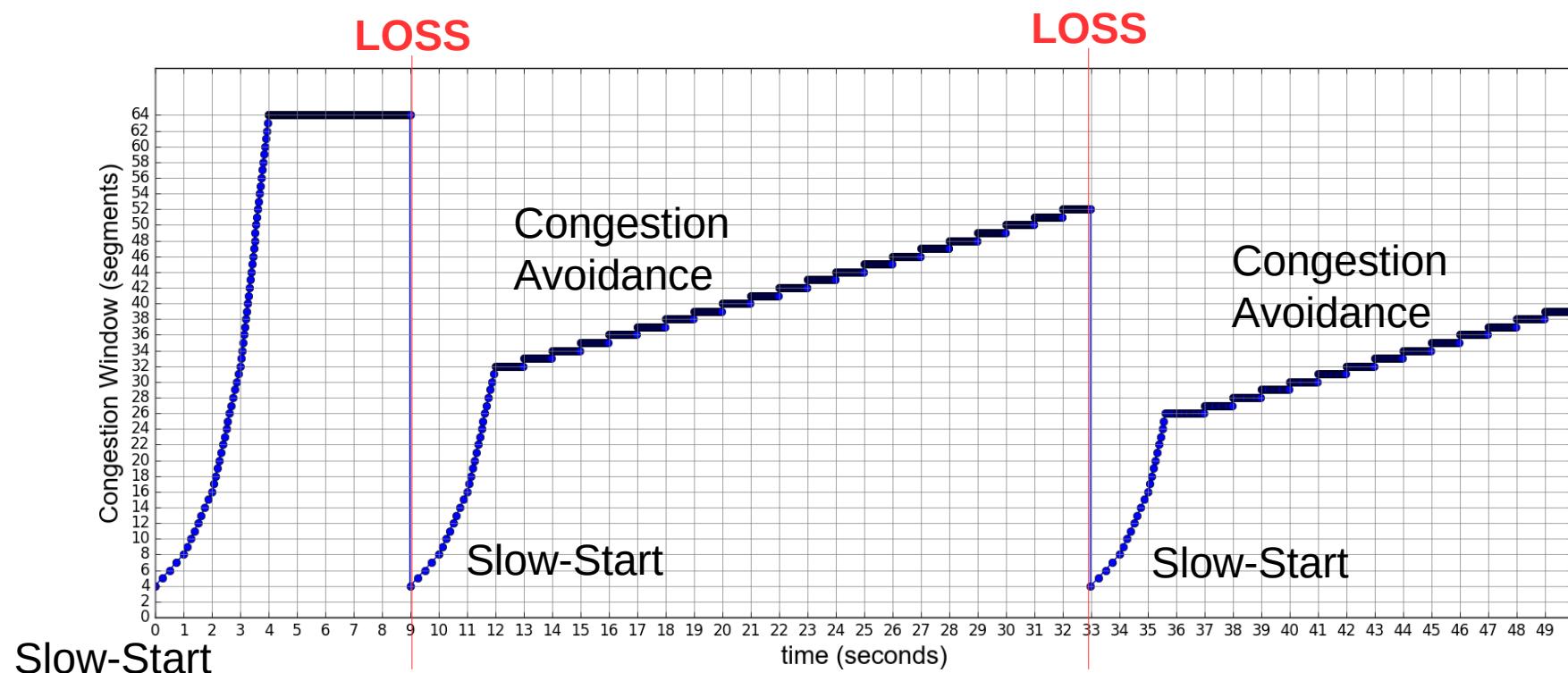
TCP Slow-Start

- At the beginning, the initial CWND value is usually 2, 3, 4, or 10 MSS and the terminal starts the *Slow Start* with SSThresh=RWND.
- CWND size grows very fast while smaller than the predefined threshold ($CWND < SSThresh$).
 - ◆ When a ACK arrives the CWND is updated: $CWND = CWND + N$,
 - ◆ N is the number of bytes acknowledged in the ACK.
 - ◆ Results that the window size (approximately) doubles each round-trip time.
 - Exponential growth.
 - ◆ Continues until a loss occurs or CWND reaches RWND.
- When a loss occurs, SSThresh is defined as $CWND/2$ and Slow-Start begins again from its initial CWND.
- Once the CWND reaches the SSThresh, it changes to congestion avoidance algorithm.
 - ◆ Linear growth.



TCP Congestion Avoidance

- When a ACK arrives the CWND is updated:
 $CWND=CWND+N/CWND$,
 - This results in a linear increase of the CWND.

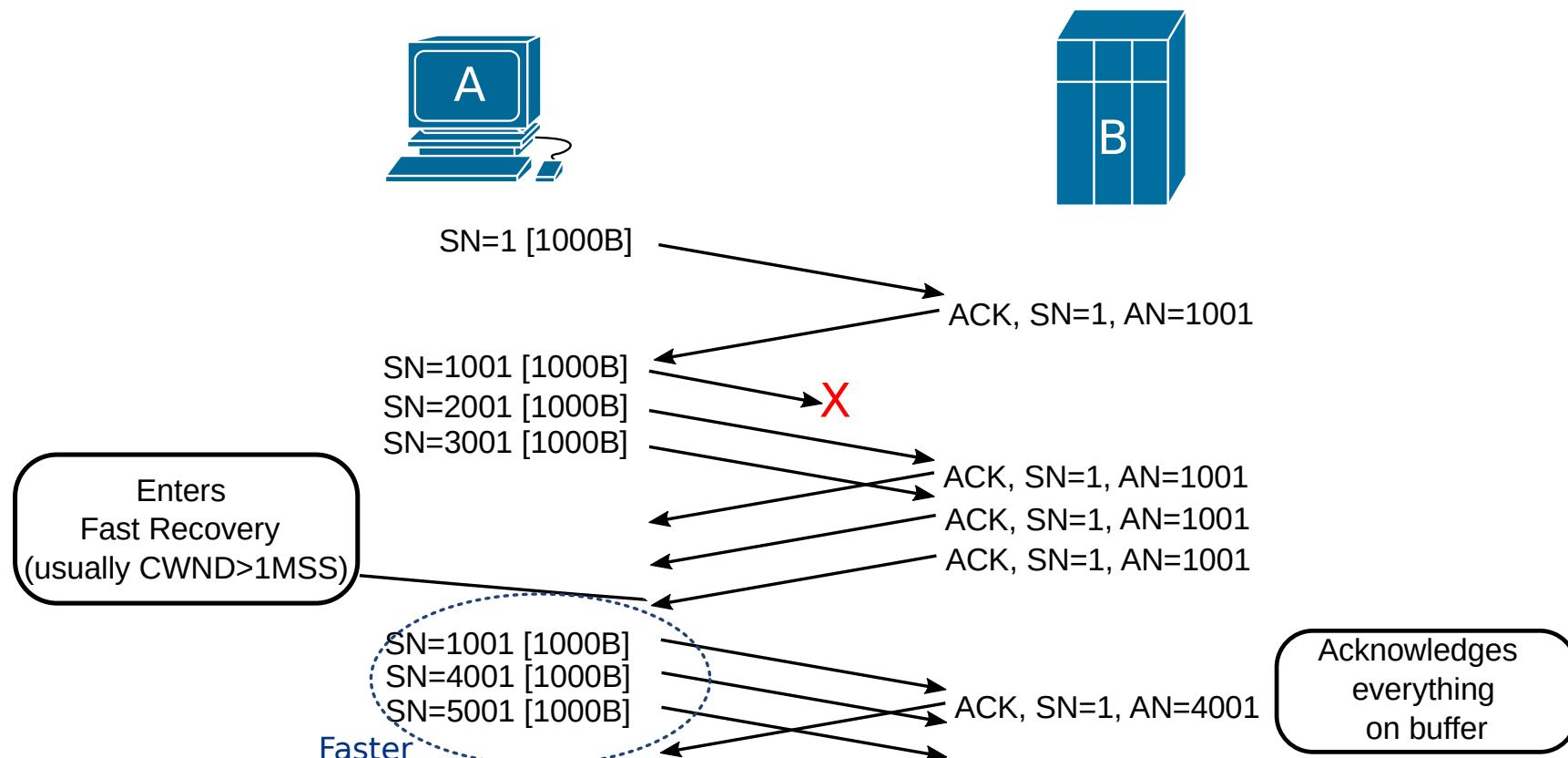


- RWND is 64, and initial CWND is 4 MSS (segments).
- Initial SSThresh=RWND=64. After first loss: SSThresh=CWND/2=32. After second loss: SSThresh=CWND/2=26.



TPC Fast Recovery

- The same as TPC Fast Retransmit.
- However when a loss occurs enters directly to Congestion Avoidance with:
 - ◆ $SSThresh = CWND/2$
 - ◆ $CWND = SSThresh$
 - ◆ Some implementation have: $CWND = SSThresh + 3 * MSS$.



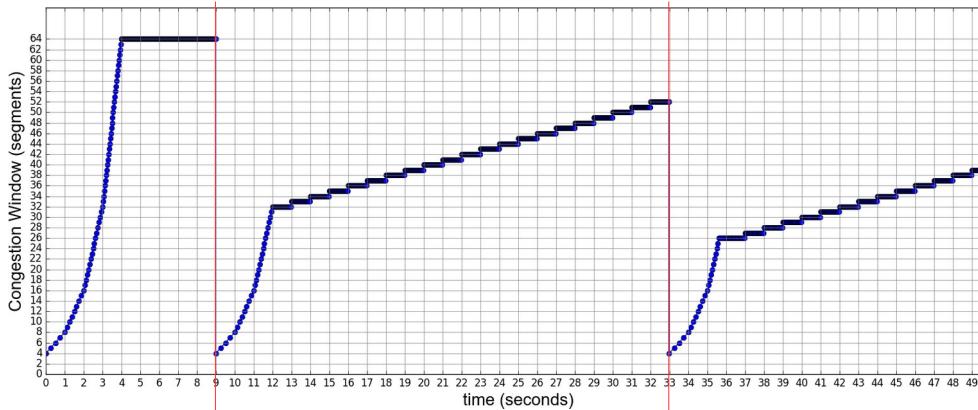
Other TCP Algorithms

- NewReno (1996)
 - ◆ Allows for partial ACK.
 - ◆ When a loss occurs, CWND is defined as $\beta \times \text{CWND}$, with $\beta=0.5$. When a ACK arrives, CWND is updated as $\text{CWND}=\text{CWND}+\alpha$, with $\alpha=1$ MMS.
 - ◆ Used by default in Windows and supported by Mac OS X.
 - Used in Windows XP and earlier.
 - After Windows Vista, Compound TCP can also be enabled.
- CUBIC (2005)
 - ◆ Uses a cubic function to control the CWND.
 - ◆ Used by Linux (kernel 2.6.19 and later) and supported by Mac OS X.
- Compound TCP (2006)
 - ◆ Adapts its behavior by use of a scalable delay-based component. T
 - Increases throughput more quickly in the congestion avoidance phase.
 - ◆ The AWND depend on the RTT measurements from successfully acknowledged packets.
 - ◆ Windows OS supports it as an option.
- Low Extra Delay Background Transport (LEDBAT)
 - ◆ Delay-based congestion control algorithm that uses all the available bandwidth while limiting the increase in delay. Measures one-way delay.
 - ◆ Supported by Windows 10 and latest versions of Mac OS X.

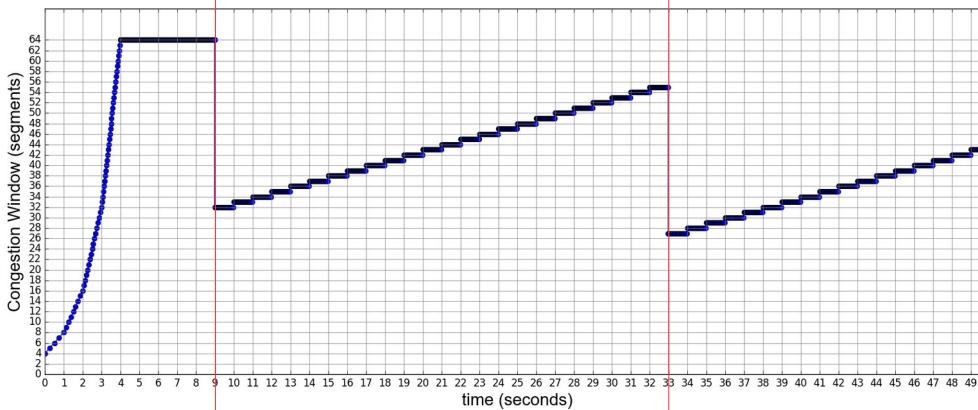


TCP Algorithms Comparison

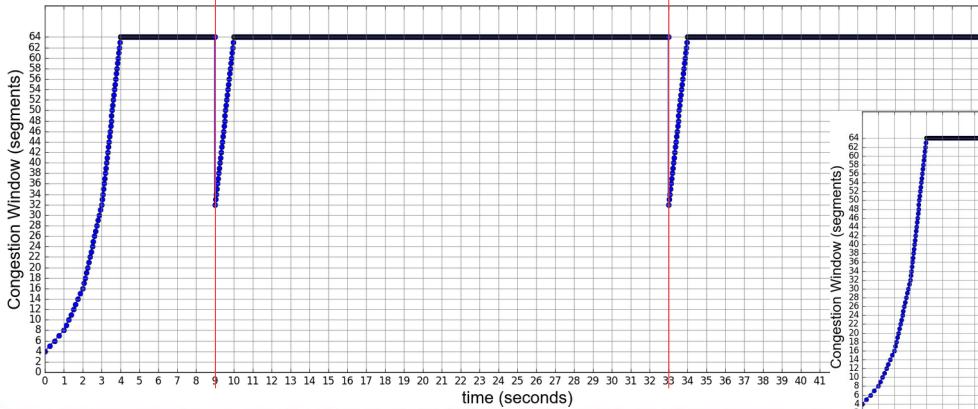
- Tahoe



- Reno



- NewReno



- Cubic



Multipath TCP (MPTCP)

- TCP is essentially a single-path protocol.
 - ◆ When a TCP connection is established, the transmission is bound to the IP addresses of the two end-points.
 - ◆ If one address changes the TCP session will fail.
 - ◆ TCP can not load balance segments using more than one TCP session.
 - This load balancing must be done at the application level.
- Multipath TCP allows multiple subflows within a single MPTCP session.
 - ◆ A MPTCP session starts with an initial subflow, using the traditional 3-Way Handshake.
 - ◆ After the first MPTCP subflow is established, additional subflows can also be established similar to the traditional TCP 3-Way Handshake. However, rather than being a separate session, all subflows are bounded to the same MPTCP session.
 - ◆ Data can then be sent over any of the active subflows, using joint *Sequence* and *Acknowledgment Numbers*.
- Apple's Siri application uses Multipath TCP.



Network Programming

Fundamentos de Redes

**Mestrado Integrado em
Engenharia de Computadores e Telemática
DETI-UA**

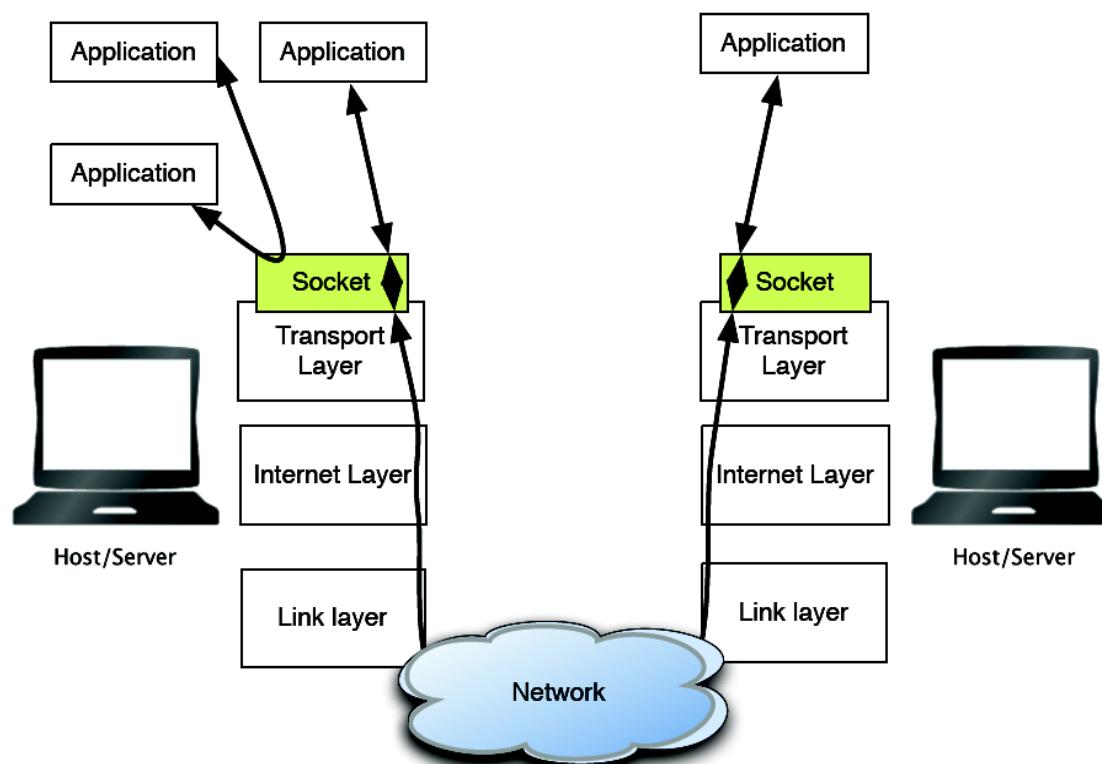


universidade de aveiro

deti.ua.pt

Sockets (1)

- Inter-process communication mechanism
 - ◆ Either local or remote processes
- Provide an abstraction for processes to exchanging information
 - ◆ Follows a client/server paradigm.



Sockets (2)

- A Socket is identified by
 - ◆ Family: AF_INET (IPv4), AF_INET6 (IPv6) and many other less common.
 - ◆ Defines the address structure.
 - ◆ Defines also the communications layer (e.g. IP version).
 - ◆ Type: Determines what transport protocol is used.
 - ◆ UDP – Connectionless (SOCK_DGRAM).
 - ◆ TCP – Connection oriented (SOCK_STREAM).
 - ◆ RAW – Direct access to a layer of the stack (SOCK_RAW).
 - Allows to send and receive crafted packets.
 - e.g. the ping command (ICMP packets).
 - ◆ Address: local address (IP or path)
 - ◆ Also remote address if connection oriented
 - ◆ Port: Local port 0-65535
 - ◆ Also remote port if connection oriented
- Restriction
 - ◆ 1 socket per Address, per Port, per Protocol, per Family, per Host



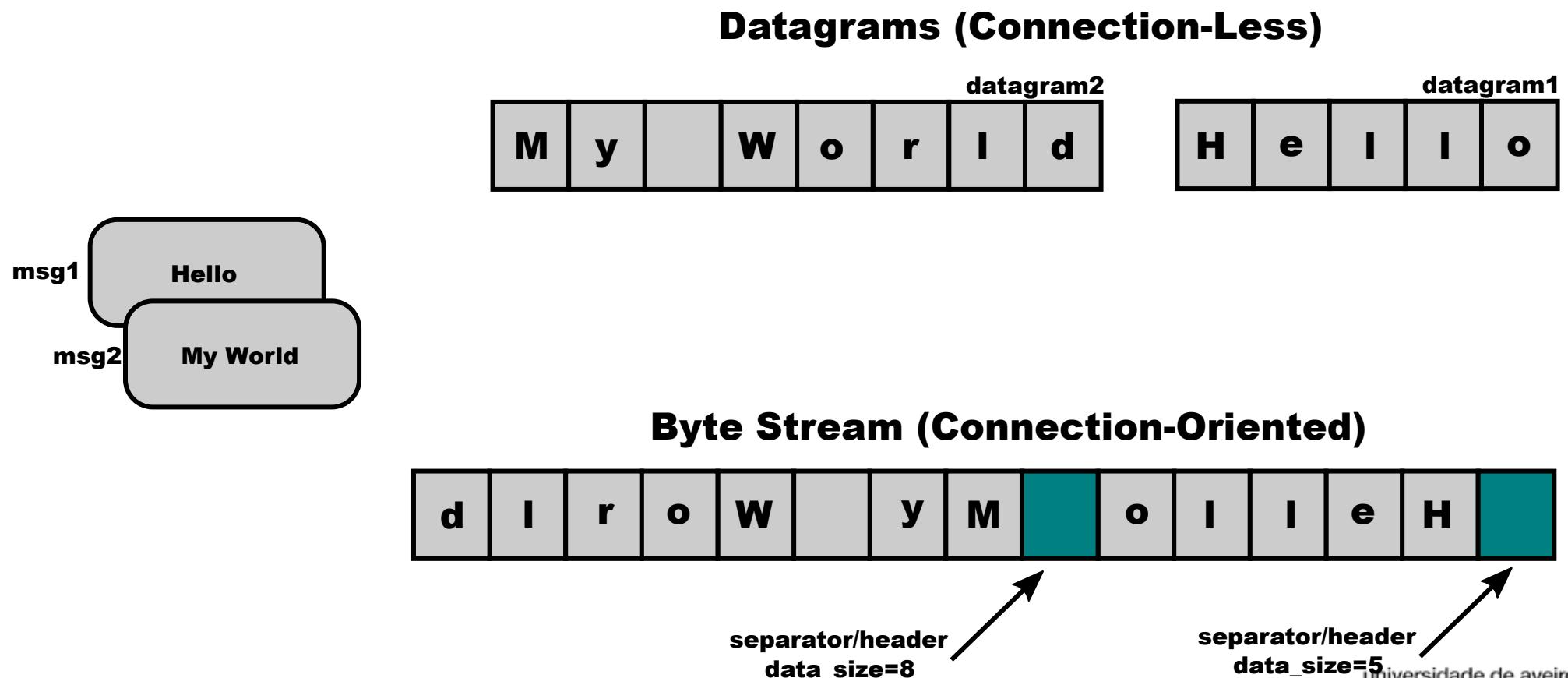
Sockets (3)

- AF_INET/AF_INET6 families
 - ◆ Allows communication between processes on any IP/IPv6 enabled machine.
 - ◆ Endpoints can be on local or remote machines
 - ◆ 127.0.0.1 or ::1 for the localhost
- A Socket must be “Bound” to a local IP/PORT
 - ◆ Sockets can be bound to a specific address or to any address
 - ◆ e.g. 192.168.0.1 (only listens in this address)
 - ◆ e.g. 0.0.0.0 (listens in all active addresses and broadcast)
 - ◆ bind() method can be used to associate a Socket to a local IP/Port.



Byte Stream vs. Datagrams

- TCP needs application-level message separators (headers).
 - ◆ Must contain size information of each “independent” data chunk in the bytestream.

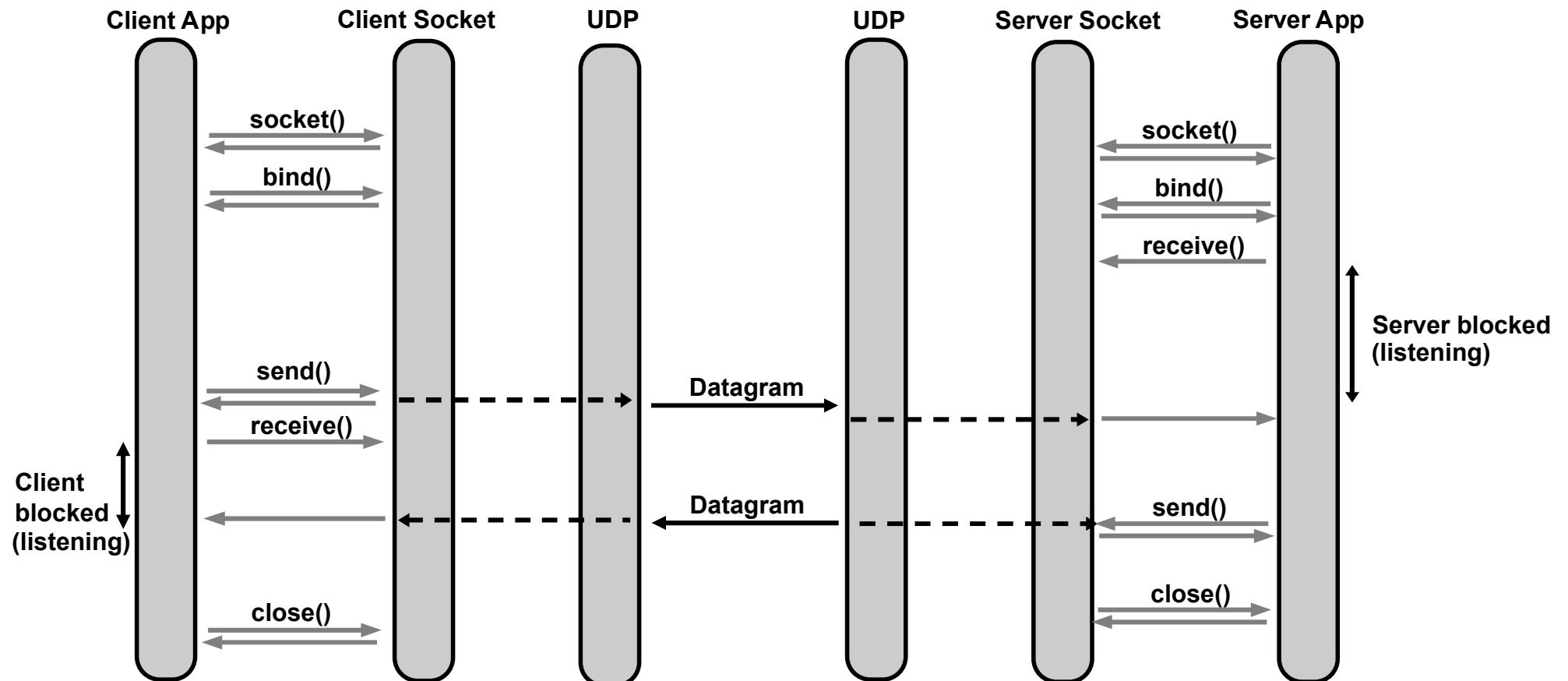


Socket IO / Blocking

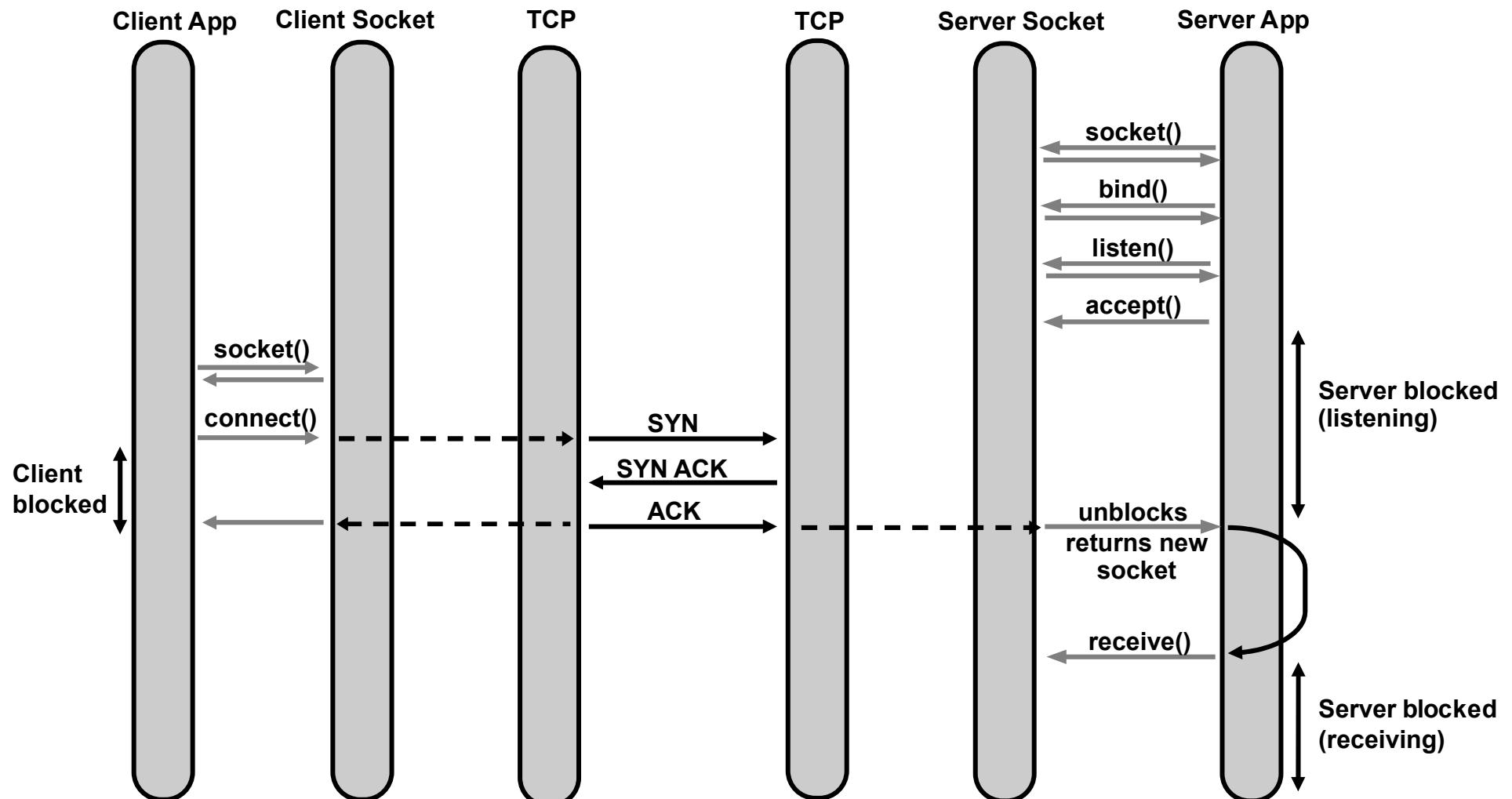
- Socket Operations are Blocking
 - ◆ They block until:
 - ◆ Packet is fully sent,
 - ◆ Client is accepted,
 - ◆ Packet is received,
 - ◆ Etc...
 - ◆ Can be set to non-blocking.
 - ◆ Program flow must take that in consideration.



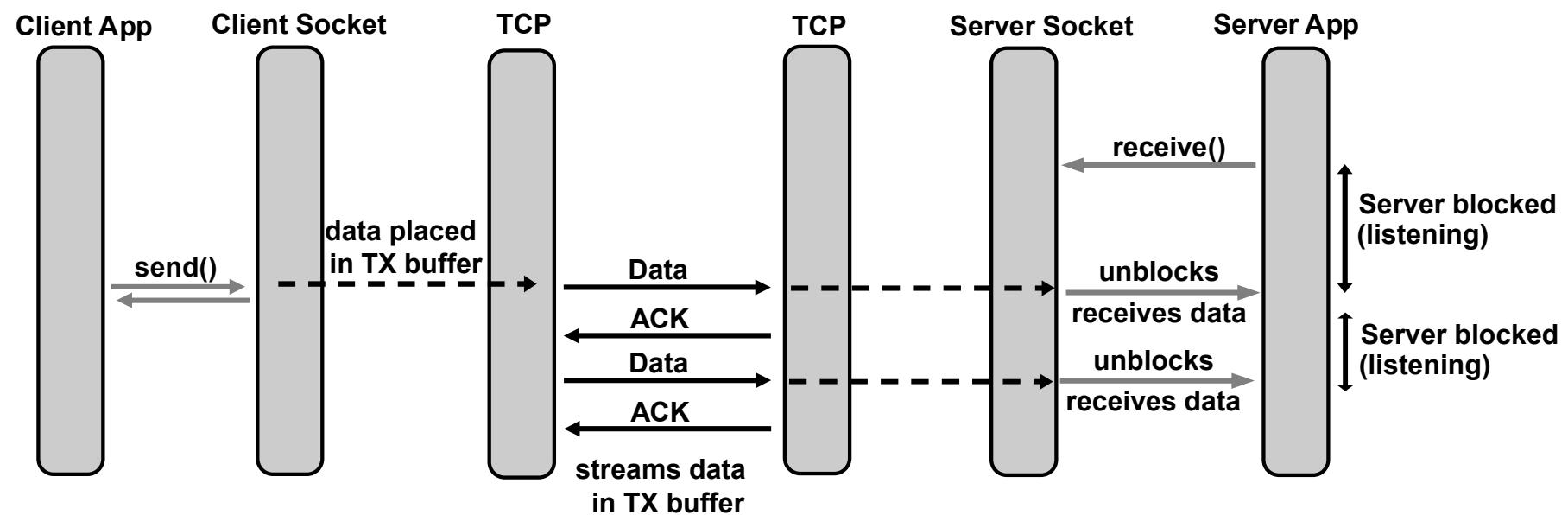
Connection-Less



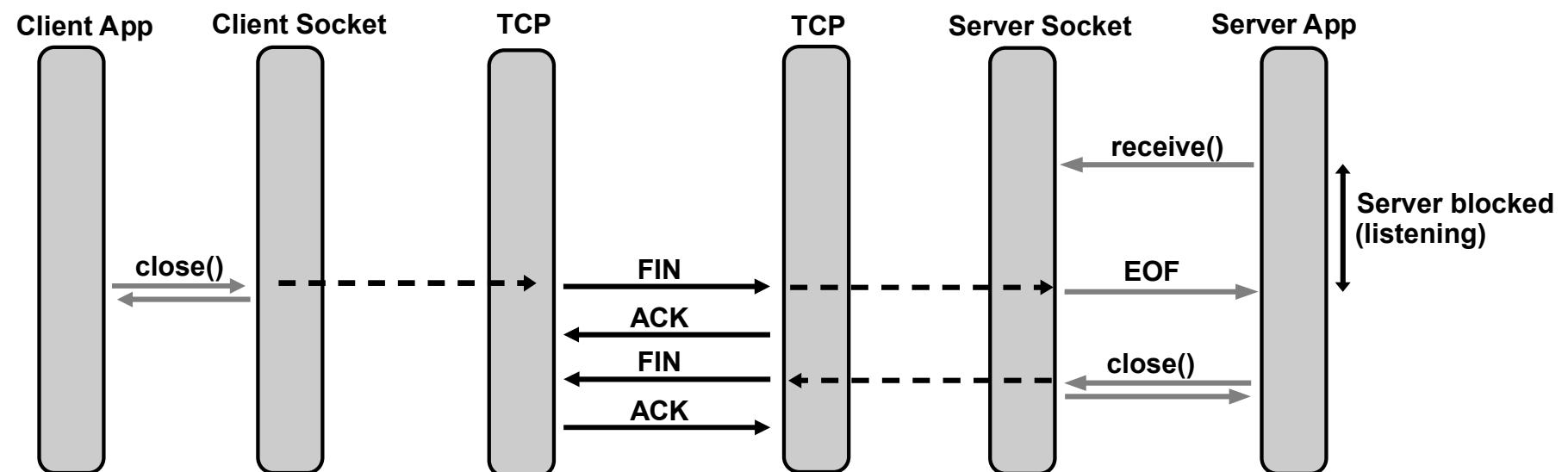
Connection-Oriented (1)



Connection-Oriented (2)



Connection-Oriented (3)



Non-Blocking IO

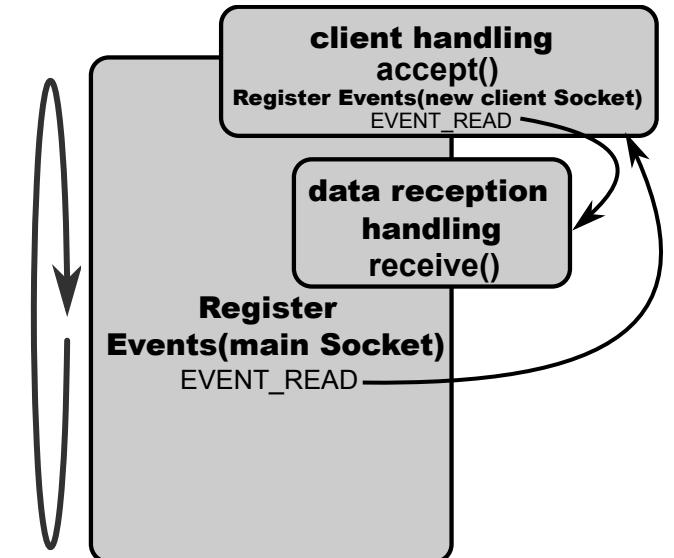
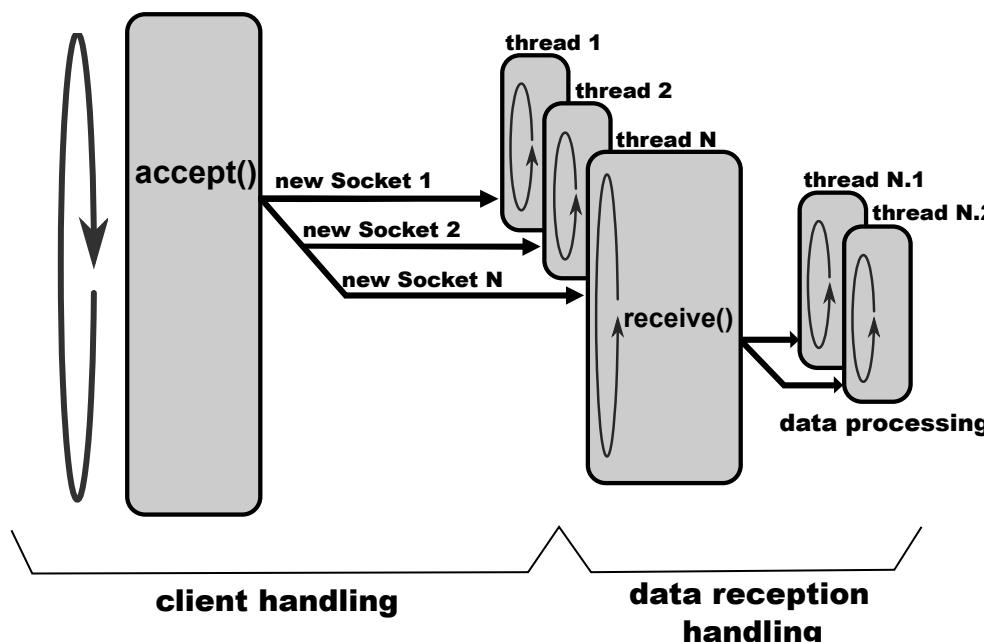
- Solutions for Socket Operations Blocking

- Threads

- Multiple parallel process can be used to process simultaneous connections.
 - Most solutions used (and still use) IO operations with multiple threads.

- Selector

- Socket is set to non-blocking.
 - Actions are performed upon the detection of predefined socket events (e.g., EVENT_READ – data available to read).



Socket Timeouts

- A socket can be in one of three modes:
 - ◆ Blocking,
 - ◆ Default state.
 - ◆ Non-blocking,
 - ◆ or Timeout.
- In blocking mode, operations block until complete or the system returns an error (such as connection timed out).
- In non-blocking mode, operations fail if they cannot be completed immediately.
 - ◆ Selects can be used to know when and whether a socket is available for reading or writing.
- In timeout mode, operations fail if they cannot be completed within the timeout specified for the socket (they raise a timeout exception) or if the system returns an error.



Data Format



Textual vs. Binary Structure

• Textual

- ◆ Pure text (format based on CSV, TSV, newline, ...), HTML, JSON, XML.
- ◆ Larger messages and higher processing times.
 - Higher Bandwidth, CPU and Memory requirements.
 - Constrains utilization in high performance applications.

• Binary Structure

- ◆ Defined by the protocol stack (definition of formats and methodologies).
- ◆ Faster at all levels.
- ◆ Little/Big Endian concerns.
 - Must depend on platform and/or be defined by the protocol stack.

```
{"msg_id":21654,  
 "values":[12, 45, 109]  
 }
```

Message data has **42 bytes**

VS.

Structure format

```
uint16 msg_id  
uint8 num_values  
uint8 values[]
```

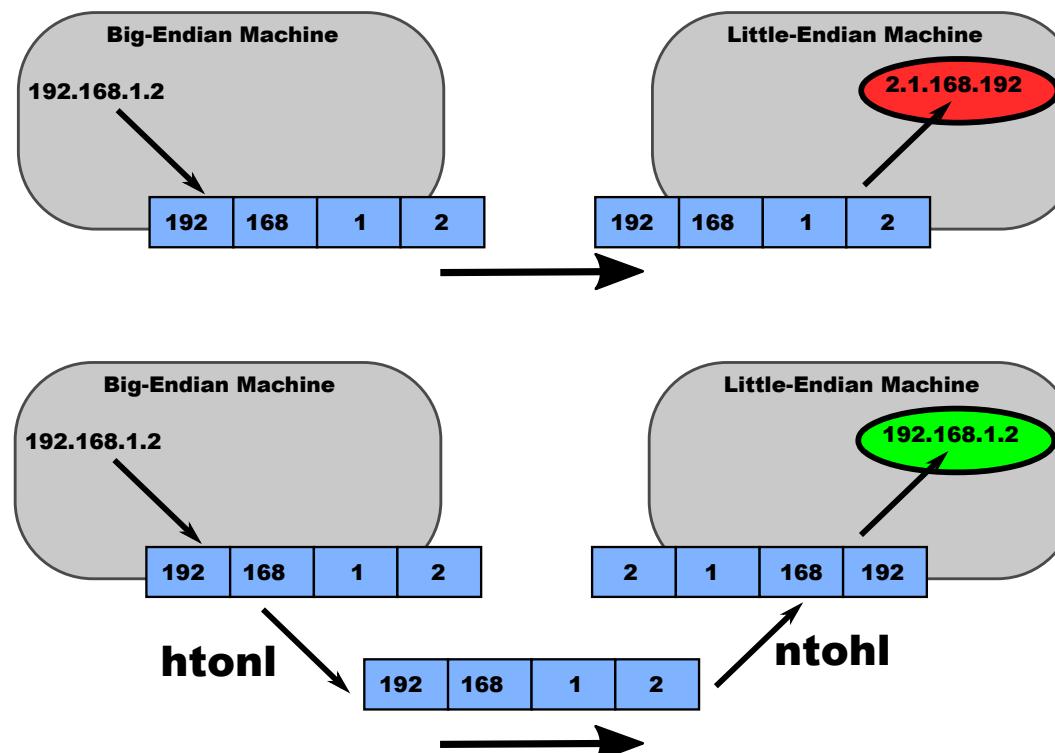
Message data
has **6 bytes**

| | |
|----------------|------------|
| 0x5496 | Big Endian |
| 0x03 | |
| 0x0C 0x2D 0x6D | |



Network/Host Formats

- Different computers architectures/OS use different byte orderings internally for their multibyte integer.
 - ◆ **htonl(i)**, **htons(i)**
 - ◆ 32-bit or 16-bit integer from host format to network format (Big-endian).
 - ◆ **ntohl(i)**, **ntohs(i)**
 - ◆ 32-bit or 16-bit integer from network format to host format.



Applications (Client-Server Model)

Fundamentos de Redes

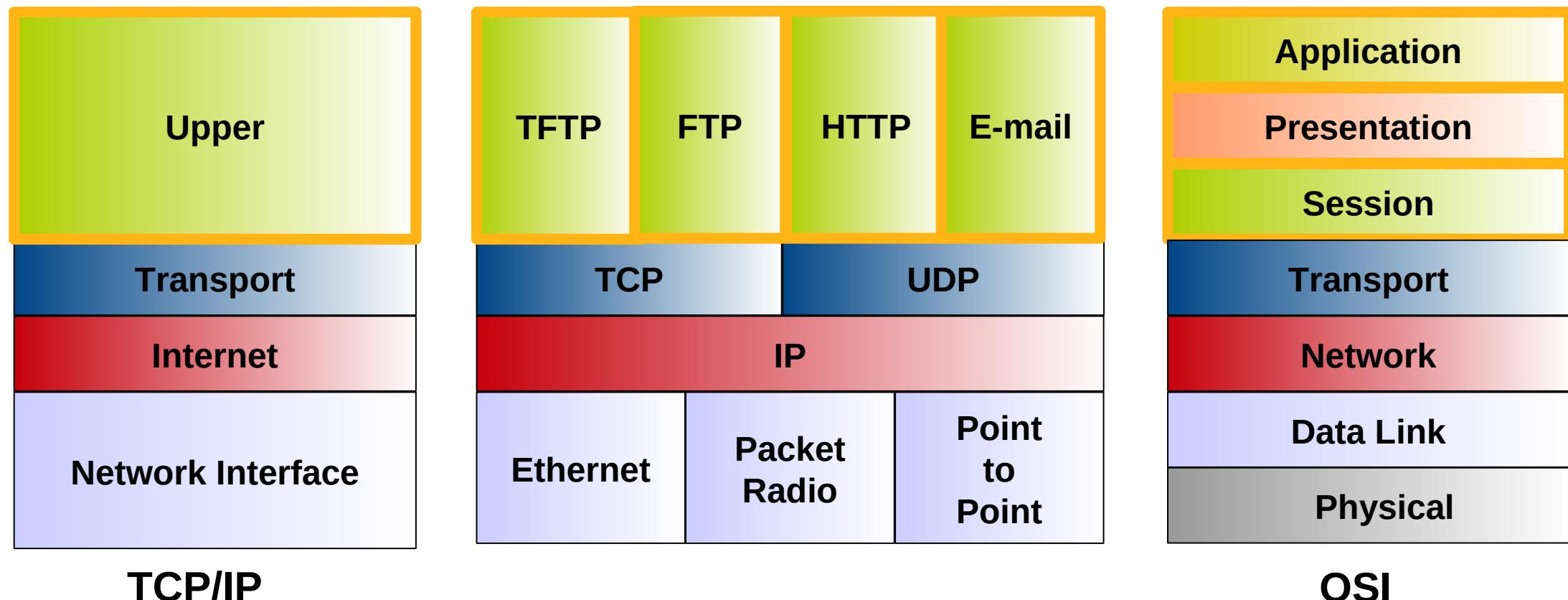
Mestrado Integrado em
Engenharia de Computadores e Telemática
DETI-UA



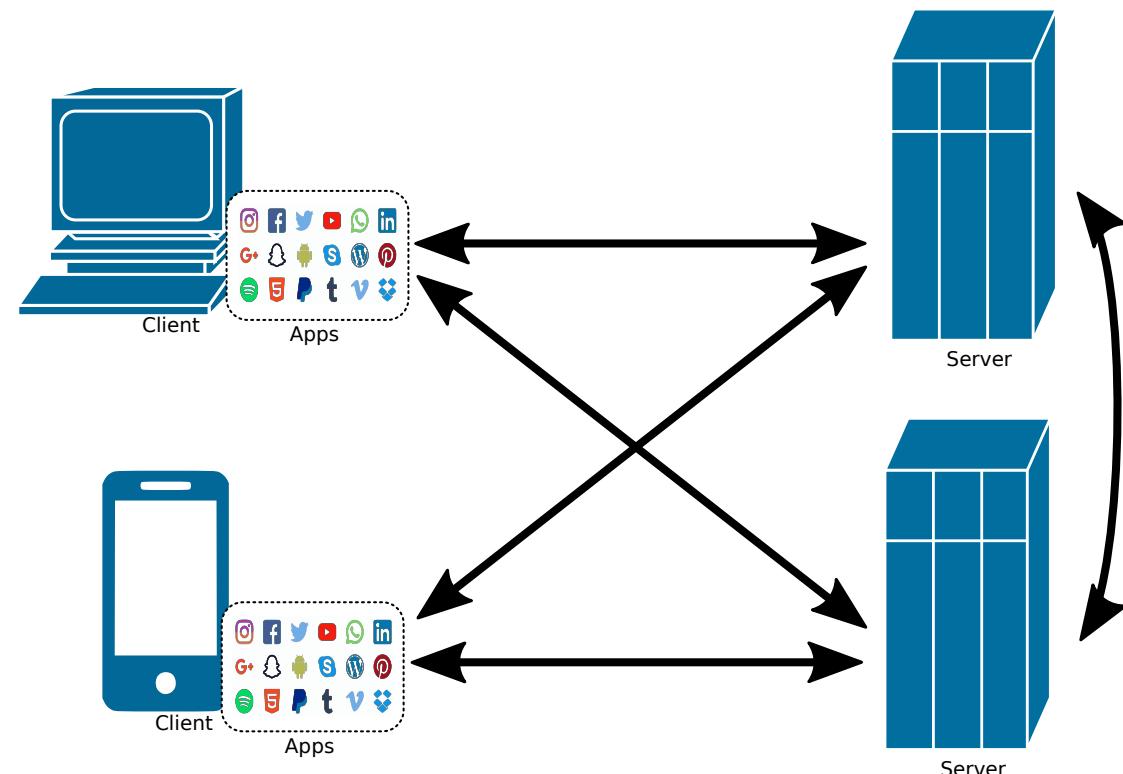
universidade de aveiro

deti.ua.pt

TCP/IP Reference Model



Client-Server Model



Servers:

- Always ON.
- IP address is always the same or exists a static association between a name and a dynamic IP address.
- May communicate between them.
 - May act as client.

Clients:

- Communicate with servers.
- Can be ON only when in operation.
- May have dynamic addresses.
- Within this model, they do not communicate between themselves.
 - P2P is another communication model.

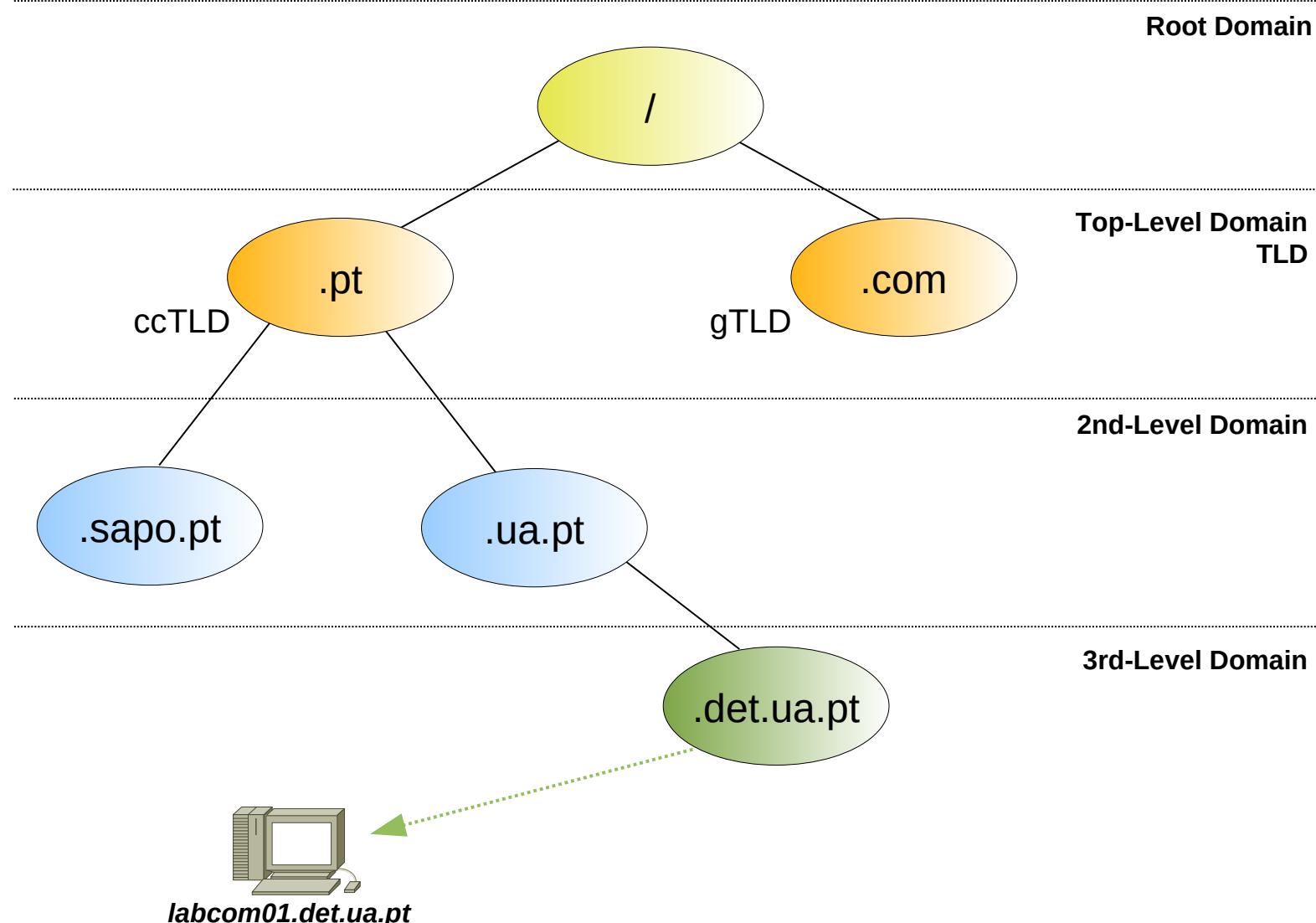


Domain Name System (DNS)

- Distributed database system that facilitates a translation service (resolution) between host names and IP addresses.
- Allows also the translation/resolution between IP addresses and host names
 - The name "DD.CC.BB.AA.**in-addr.arpa**" allows the resolution of the IPv4 address AA.BB.CC.DD
 - The name 0.0.0.8.b.d.0.1.0.0.2.**ip6.arpa** allows the resolution of the IPv6 address 2001:0db8:0000::/48
 - Resolution name-ip and ip-name is not symmetrical.
- Organizes the names in domains according to an hierarchical structure.
- Each DNS system defines one or more zones over which has the resolution authority.



Hierarchical Structure of Domain Names



Root Servers & Root Zone File

- Root servers



- Root Zone File (sample)

.....
COM. NS A.GTLD-SERVERS.NET.
COM. NS G.GTLD-SERVERS.NET.
COM. NS H.GTLD-SERVERS.NET.
COM. NS C.GTLD-SERVERS.NET.

.....
PT. NS NS.DNS.BR.
PT. NS NS2.NIC.FR.
PT. NS NS.DNS.PT.
PT. NS SUNIC.SUNET.SE.
PT. NS NS2.DNS.PT.
PT. NS NS-EXT.ISC.ORG.

.....
NET. NS A.GTLD-SERVERS.NET.
NET. NS G.GTLD-SERVERS.NET.
NET. NS H.GTLD-SERVERS.NET.
NET. NS C.GTLD-SERVERS.NET.

.....
INFO. NS B0.INFO.AFILIAS-NST.ORG.
INFO. NS C0.INFO.AFILIAS-NST.INFO.
INFO. NS D0.INFO.AFILIAS-NST.ORG.



Top-Level Domains (TLD)

- gTLDs (generic TLDs)

- .com, .edu, .gov, .mil, .net, .org, .int, .aero, .biz, .coop, .info, .museum, .name, .pro, .cat, .jobs, .mobi, .travel, .tel, .asia

- ccTLDs (country code TLDs)

- 2 letter domains that identify a specific country (ISO 3166)
 - Management is delegated (by ICANN) to a governmental institution from each country.
 - Those can (re)-delegate to private companies.
 - Ex: .pt, .es, .us, .fr, etc...

- New gTLDs

- Over 1300 new gTLDs could become available in the next few years.
 - Trademarks: **.goog, .goggle, .apple, .yahoo, .honda, .barcelona, ...**
 - **.xyz, .top, .wang, .win, .link, .site, .club, .app, .live, .cloud, .bank, .online, .bet, .book, .cars, .hotel, ...**



TLD Zone Files (sample)

•.ORG (Public Interest Registry)

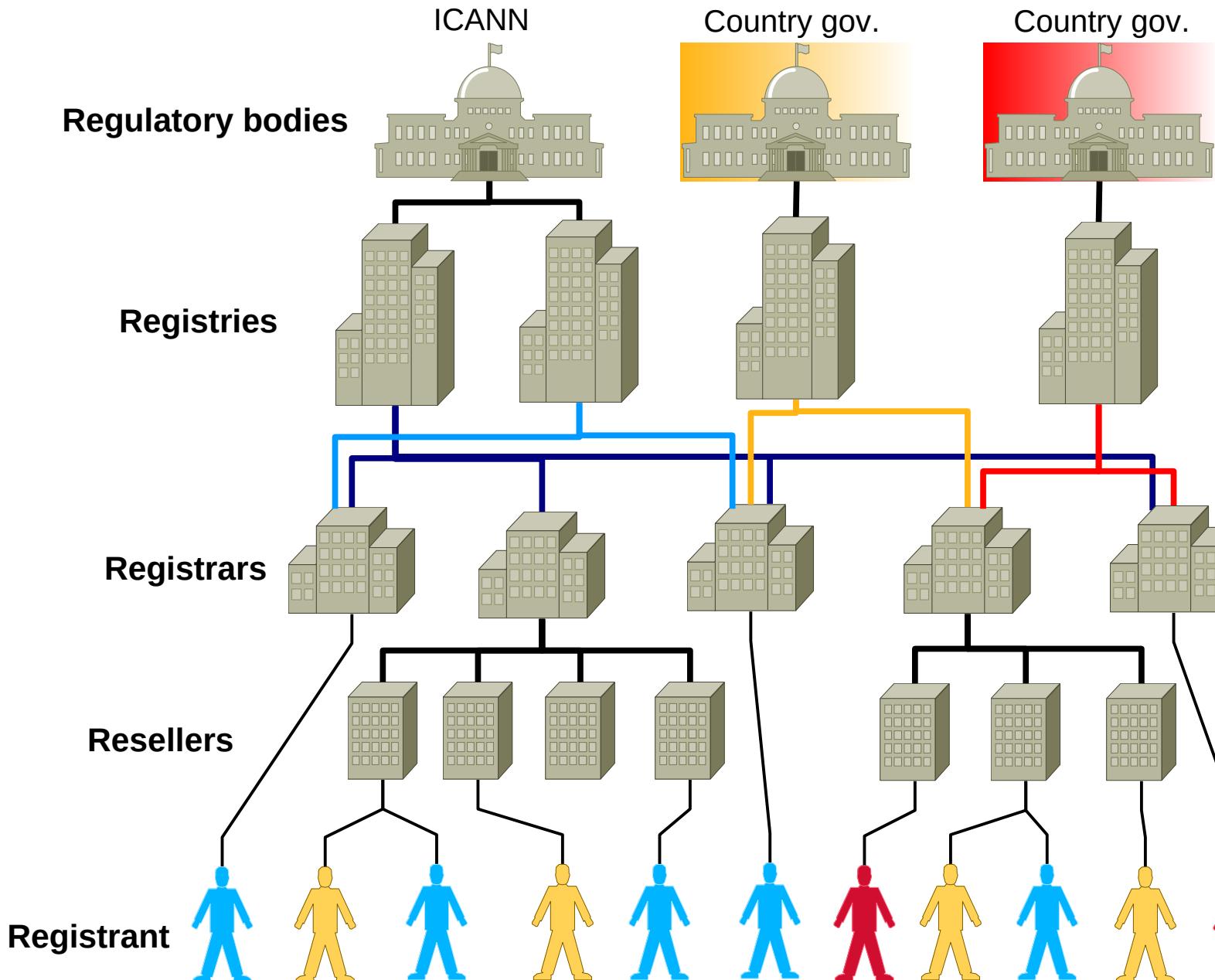
```
.....  
AASELFSTORAGE.ORG. NS DNS02.GPN.REGISTER.COM.  
AASELFSTORAGE.ORG. NS DNS03.GPN.REGISTER.COM.  
AASELFSTORAGE.ORG. NS DNS04.GPN.REGISTER.COM.  
AASELFSTORAGE.ORG. NS DNS05.GPN.REGISTER.COM.  
AASEMI.ORG. NS DPNS1.DNSNAMESEVER.ORG.  
AASEMI.ORG. NS DPNS2.DNSNAMESEVER.ORG.  
AASEMI.ORG. NS DPNS3.DNSNAMESEVER.ORG.  
AASEMI.ORG. NS DPNS4.DNSNAMESEVER.ORG.  
AASEN.ORG. NS NS1.MAILBANK.COM.  
AASEN.ORG. NS NS2.MAILBANK.COM.  
AASENIORMORTGAGE.ORG. NS NS13.DOMAINCONTROL.COM.  
AASENIORMORTGAGE.ORG. NS NS14.DOMAINCONTROL.COM.  
AASENT.ORG. NS NS51.1AND1.COM.  
AASENT.ORG. NS NS52.1AND1.COM.  
AASENTMORTGAGE.ORG. NS NS51.1AND1.COM.  
AASENTMORTGAGE.ORG. NS NS52.1AND1.COM.  
AASENY.ORG. NS NS27.1AND1.COM.  
AASENY.ORG. NS NS28.1AND1.COM.  
AASEP.ORG. NS NS1.CASTIRONCODING.COM.  
AASEP.ORG. NS NS2.CASTIRONCODING.COM.  
AASERV.ORG. NS NS1.RENEWYOURNAME.NET.  
.....
```

•.COM (Verisign)

```
.....  
AMERICANHUNTING NS NS1.HITFARM  
AMERICANHUNTING NS NS2.HITFARM  
ATSCAF NS CBRU.BR.NS.ELS-GMS.ATT.NET.  
ATSCAF NS CMTU.MT.NS.ELS-GMS.ATT.NET.  
ACTIONNETS NS NS.TULSAWEB  
ACTIONNETS NS NS.TIBP  
ACI-APPLICAD NS NS2.WEBNJ.NET.  
ACI-APPLICAD NS NS1.WEBNJ.NET.  
ANZAPACK NS DNS3.TERRA.ES.  
ANZAPACK NS DNS4.TERRA.ES.  
ALPHASOFTDE NS DNS1.EPAG.NET.  
ALPHASOFTDE NS DNS2.EPAG.NET.  
ALPHASOFTDE NS DNS01.KUTTIG.NET.  
AAI-TENN NS AUTH00.DNS.BELLSOUTH.NET.  
AAI-TENN NS AUTH01.DNS.BELLSOUTH.NET.  
AAI-TENN NS AUTH02.DNS.BELLSOUTH.NET.  
ALLIEDMAXCUT NS NS3.DHCNET.NET.  
ALLIEDMAXCUT NS NS0.DHCNET.NET.  
ATLANTAEXOTICS NS NS1.APHOST  
ATLANTAEXOTICS NS NS2.APHOST  
ATLANTA-EXOTICS NS NS3.LNHI.NET.  
ATLANTA-EXOTICS NS NS2.LNHI.NET.  
ATLANTA-EXOTICS NS NS1.LNHI.NET.  
.....
```



Domain Management Model (1)



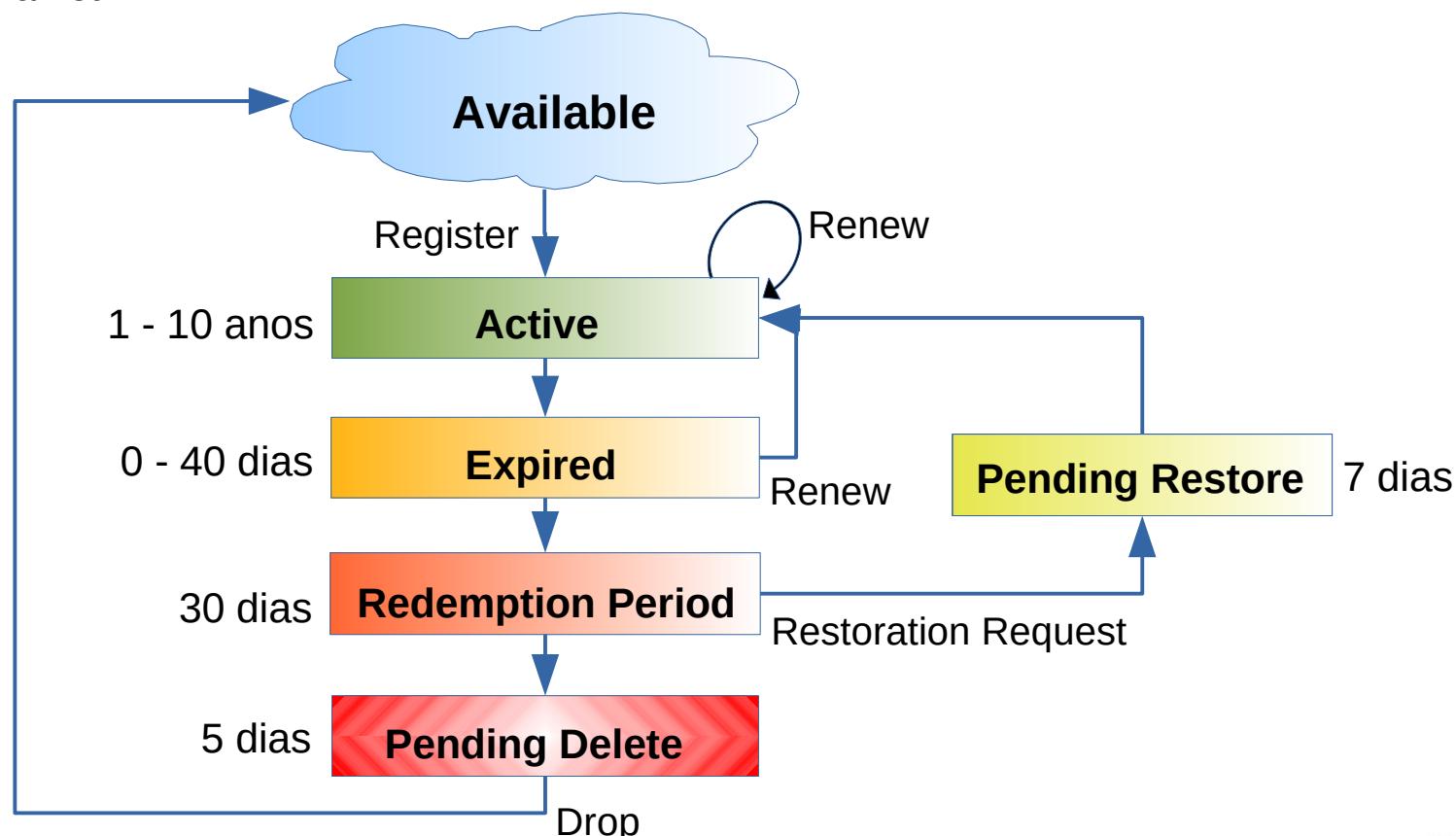
Domain Management Model (2)

- Delegation and Authority lie at the core of the domain name system hierarchy.
- The Authority for the root domain lies with Internet Corporation for Assigned Numbers and Names (ICANN).
 - gTLDs are authoritatively administered by ICANN and delegated to a series of accredited entities.
 - ccTLDs are delegated to the individual countries for administration purposes.
- The entity responsible by a specific domain is called **Registry**.
 - In charge of maintaining the Zone File of the TLD.
- **Registries** (usually) delegate in **Registrar** the operational management and marketing of a domain.
 - One **Registry** can delegate to multiple **Registrars**
 - The **Registrar** stores and manages the information and status of a domain.
- One **Registrar** may still accept **Resellers**
 - A **Reseller** sells domains from a **Registrar** (for a commission)
 - The management of the domains is not responsibility of a **Reseller**.
- A **Registrant** is any entity that wants to register a domain name.



Domain Name Life Cycle

- A domain can be registered for a period of 1 to 10 years.
 - After that period the domain must be renewed.
- In case of no renewal, it's initiated the process of deletion of the domain name from the DNS database.
 - Nowadays, the Registrars do not release the domain immediately after the redemption period, they initiate a reselling mechanism (usually some kind of auction) of the domain on the secondary market.



WHOIS Service and Information

- Contains information about the registrant of a domain
 - Name servers
 - Status of the domain
 - Registry-Registrar Protocol (RPP)
 - Extensible Provisioning Protocol(EPP)
 - Creation, expiration and last update dates.
 - Registrant contacts
 - General
 - Administrative
 - Technical
 - Billing
- This information can be retrieved using the WHOIS service
 - Executes recursive queries of Registry and Registrant databases.

Domain Name: NAME.COM
Registrar: NAME.COM LLC
Whois Server: whois.name.com
Referral URL: http://www.name.com
Name Server: NS1.NAME.COM
Name Server: NS2.NAME.COM
Name Server: NS3.NAME.COM
Name Server: NS4.NAME.COM
Status: ok
Updated Date: 30-jan-2009
Creation Date: 03-jan-1995
Expiration Date: 04-nov-2015

REGISTRANT CONTACT INFO
Name.com LLC
DNS Admin, 125 Rampart Way, Suite 300, Denver, CO 80230, US
Phone: +1.7202492374
Email Address: dns@name.com

ADMINISTRATIVE CONTACT INFO
Name.com LLC
DNS Admin, 125 Rampart Way, Suite 300, Denver, CO 80230, US
Phone: +1.7202492374
Email Address: dns@name.com

TECHNICAL CONTACT INFO
Name.com LLC
DNS Admin, 125 Rampart Way, Suite 300, Denver, CO 80230, US
Phone: +1.7202492374
Email Address: dns@name.com

BILLING CONTACT INFO
Name.com LLC
DNS Admin, 125 Rampart Way, Suite 300, Denver, CO 80230, US
Phone: +1.7202492374
Email Address: dns@name.com

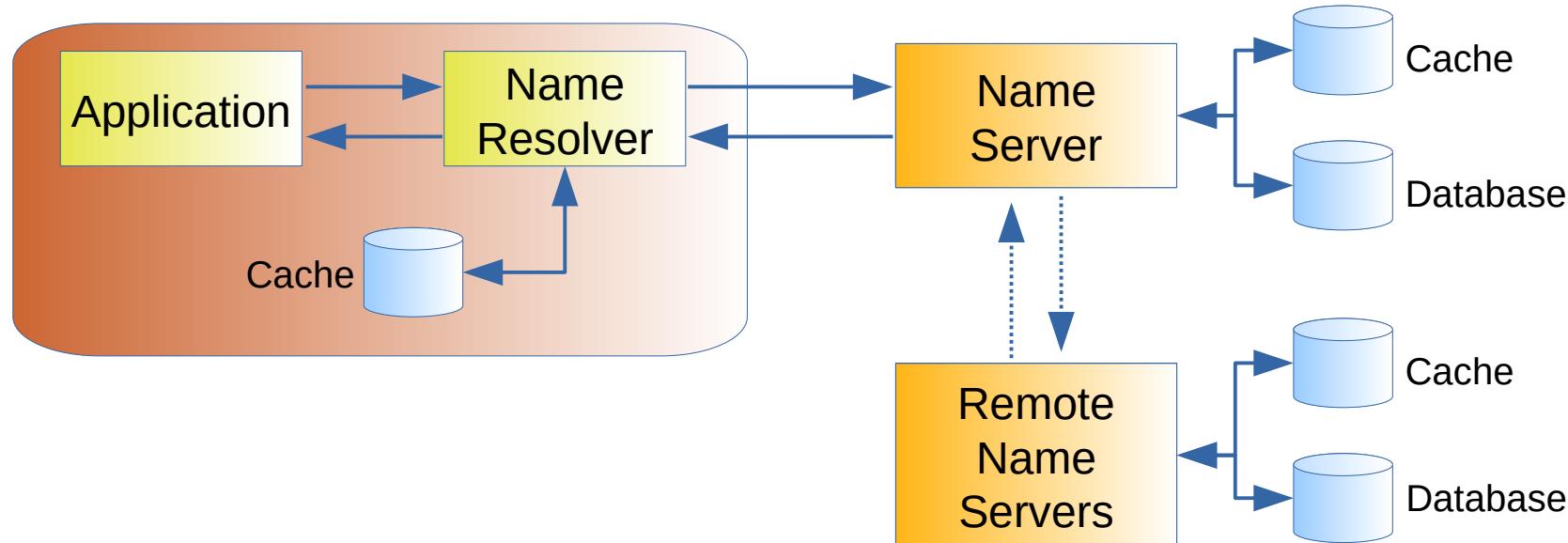


Name Servers Registration

- In order to set up a DNS server outside of your registrar, you need to:
 - Explicitly register your name server names and IPs.
 - i.e. Associate name with IP (ex: ns1.domain.com – 10.1.1.1).
 - Define server names (minimum 2) to your domain registration at your registrar.



Name Resolution



- Received answers are (may be) temporarily stored in cache (have an associated TTL)
 - Can be reused in future queries to speed up answers.
- Cache use improves the systems efficiency by eliminating unnecessary external queries.



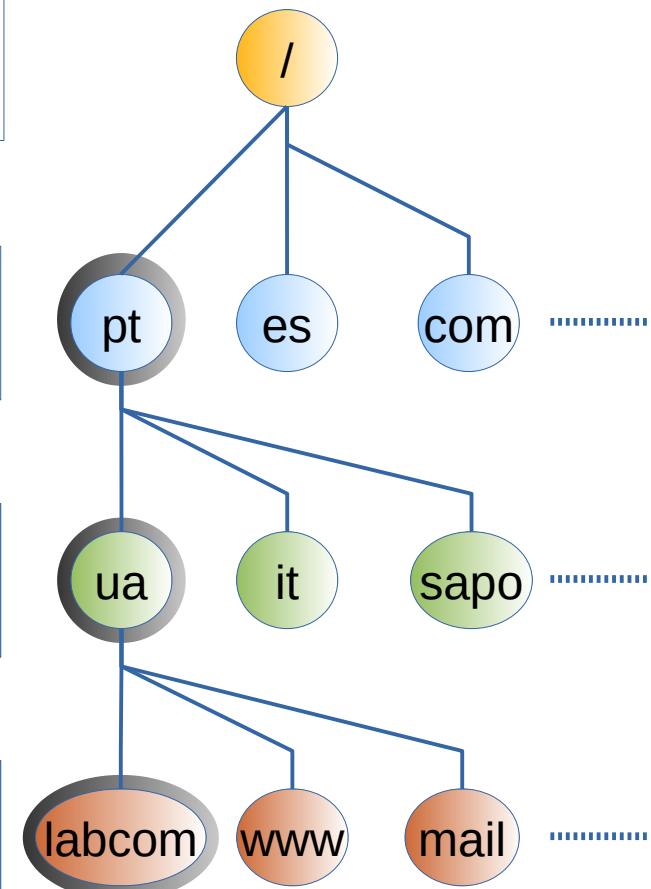
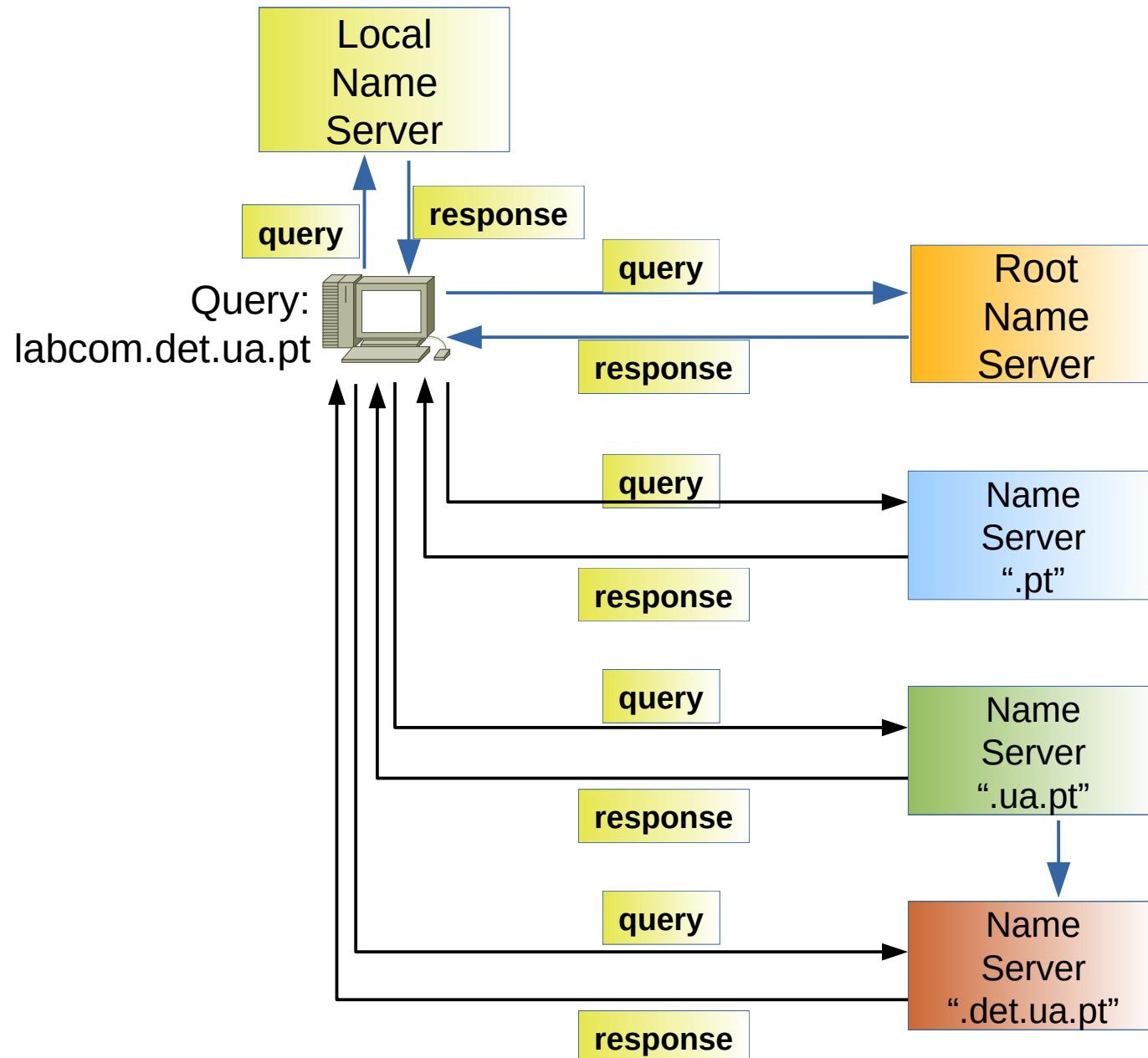
DNS Query & DNS Response

```
Frame 1928 (69 bytes on wire, 69 bytes captured)
Ethernet II, Src: 00:15:f2:9f:38:9d, Dst: 00:60:08:1f:b8:26
Internet Protocol, Src: 193.136.92.160, Dst: 193.136.92.65
User Datagram Protocol, Src Port: 54277, Dst Port: 53
    Source port: 54277 (54277)
    Destination port: 53 (53)
    Length: 35
    Checksum: 0x3c27 [incorrect, should be 0xabba (maybe
caused by "UDP checksum offload"?)]
Domain Name System (query)
    [Response In: 1929]
    Transaction ID: 0xf1e4
    Flags: 0x0100 (Standard query)
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
        www.ua.pt: type A, class I
```

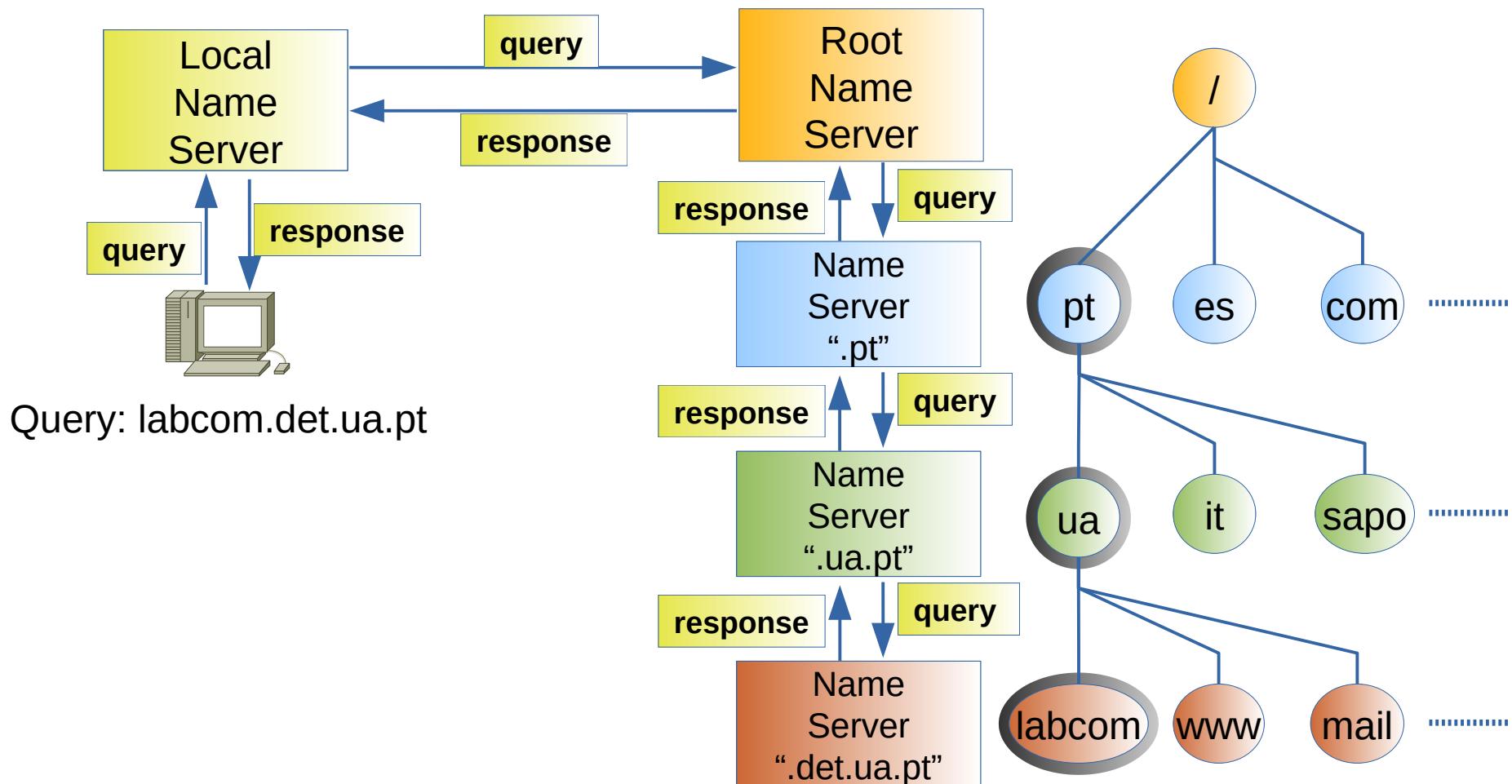
```
Frame 1929 (152 bytes on wire, 152 bytes captured)
Ethernet II, Src: 00:60:08:1f:b8:26, Dst: 00:15:f2:9f:38:9d
Internet Protocol, Src: 193.136.92.65, Dst: 193.136.92.160
User Datagram Protocol, Src Port: 53, Dst Port: 54277
    Source port: 53 (53)
    Destination port: 54277 (54277)
    Length: 118
    Checksum: 0x1167 [correct]
Domain Name System (response)
    [Request In: 1928]
    [Time: 0.005100000 seconds]
    Transaction ID: 0xf1e4
    Flags: 0x8180 (Standard query response, No error)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 2
    Additional RRs: 2
    Queries
        www.ua.pt: type A, class IN
    Answers
        www.ua.pt: type A, class IN, addr 193.136.173.25
    Authoritative nameservers
        ua.pt: type NS, class IN, ns ns2.ua.pt
        ua.pt: type NS, class IN, ns ns.ua.pt
    Additional records
        ns.ua.pt: type A, class IN, addr 193.136.172.18
        ns2.ua.pt: type A, class IN, addr 213.228.152.1
```



Iterative (Non-Recursive) Resolution



Recursive Resolution



Iterative vs. Recursive Resolution

- Iterative resolution:

- Less efficient: increases the average time between a DNS query and its response.
- Server loads are lower: each server responds immediately to a query,
 - Do not have to store any temporary information,
 - Do not perform any interaction with other DNS servers.

- Recursive resolution:

- More efficient: minimizes the average time between a DNS query and its response.
- Higher server loads: each server must simultaneously manage the state of multiple DNS queries.
 - More memory, more CPU.
 - Not a problem with current servers.



Zone Configuration

- A zone is defined by
 - A zone declaration, which holds the type of the zone, a pointer to the zone file and type specific configuration statements (optional).
 - A zone file, which holds the DNS resource records for all of the domain names associated with the zone.
- Zone files store all of the data served by a DNS server.
- The basic format of the zone file is a time to live (TTL) field followed by the Start Of Authority (SOA) records.
 - The overall TTL instructs non-authoritative DNS servers how long to cache records retrieved from the zone file.
 - With large values it will take more time to propagate changes.
 - With smaller value, the DNS server load will increase (non-authoritative servers will have to send the same requests more frequently).
 - Typical values: 1 hour to a 1 day.
 - The SOA record defines the zone name, an e-mail contact and various time and refresh values applicable to the zone.



Zone Files

- Zone files contain Resource Records that describe a domain or sub-domain.
 - Format of zone files is an IETF standard defined by RFC 1035.
- Contents
 - Data that indicates the top of the zone and some of its general properties,
 - A SOA Record.
 - Authoritative data for all nodes or hosts within the zone,
 - A (IPv4) or AAAA (IPv6) Records.
 - Data that describes global information for the zone
 - Mail MX Records and Name Server NS Records.
 - In the case of sub-domain delegation the name servers responsible for this sub-domain
 - One or more NS Records.
 - One or more A or AAAA Records



Name Server Records

- SOA (RFC 1035): Start of Authority. Defines the zone name, an e-mail contact and various time and refresh values applicable to the zone.
- A (RFC 1035): IPv4 Address record. An IPv4 address for a host.
- AAAA (RFC 3596): IPv6 Address record. An IPv6 address for a host.
- NS (RFC 1035): Name Server. Defines the authoritative name server(s) for the domain (defined by the SOA record).
- MX (RFC 1035) Mail Exchanger. A preference value and the host name for a mail server/exchanger.
- CNAME (RFC 1035): Canonical Name. An alias name for a host.
- PTR (RFC 1035): IP address (IPv4 or IPv6) to host. Used in reverse maps.
- TXT (RFC 1035): Text information associated with a name.



SOA Record (1)

- @ - represents the base domain
- IN - class of the zone (INternet)
- SOA - record identifier
- The master DNS server for the zone
 - The host where the file was created (nameserver.domain.com)
- Contact e-mail - The e-mail address of the person responsible for administering the domain's zone file.
 - "." is used instead of an "@" in the e-mail name
 - adm.domain.com <=> adm@domain.com email

```
@ IN SOA nameserver.domain.com. adm.domain.com. (
                      1           ; serial number
                      3600        ; refresh   [1h]
                      600         ; retry     [10m]
                     86400       ; expire    [1d]
                     3600 )      ; min TTL  [1h]
```



SOA Record (2)

- Serial number - The revision number of this zone file.

- Increment this number each time the zone file is changed.
- It is important to increment this value each time a change is made, so that the changes will be distributed to any secondary DNS servers.

- Refresh Time - The time, in seconds, a secondary DNS server waits before querying the primary DNS server's SOA record to check for changes.

- When the refresh time expires, the secondary DNS server requests a copy of the current SOA record from the primary.
- The secondary DNS server compares the serial number of the primary DNS server's current SOA record and the serial number in its own SOA record. If they are different, the secondary DNS server will request a zone transfer from the primary DNS server.
- The default value is 3,600.

- Retry time - The time, in seconds, a secondary server waits before retrying a failed zone transfer.

- Usually, the retry time is less than the refresh time. The default value is 600.

- Expire time - The time, in seconds, that a secondary server will keep trying to complete a zone transfer.

- If this time expires prior to a successful zone transfer, the secondary server will expire its zone file (stops answering queries).
- The default value is 86,400.

- Negative caching TTL – the time, in seconds, a negative answers (such as when a requested record does not exist) can be cached on non-authoritative servers.

- This field acts like the overall TTL but specifically for negative answers.
- Small values are appropriate (15m to 2h).

```
@ IN SOA nameserver.domain.com. adm.domain.com. (
    1 ; serial number
    3600 ; refresh [1h]
    600 ; retry [10m]
    86400 ; expire [1d]
    3600 ) ; min TTL [1h]
```



Other Records (1)

• IPv4 Address Record (A)

- Syntax: “*name ttl class rr ipv4*”

```
; zone fragment for example.com
$TTL 2d ; zone default = 2 days or 172800 seconds
joe      IN      A      192.168.0.3 ; joe & www = same ip
www      IN      A      192.168.0.3
www.example.com.  A      192.168.0.3
fred    3600  IN      A      192.168.0.4 ; TTL overrides $TTL default
ftp        IN      A      192.168.0.24 ; round robin with next
                IN      A      192.168.0.7
mail      IN      A      192.168.0.15 ; mail = round robin
mail      IN      A      192.168.0.32
mail      IN      A      192.168.0.3
```

• IPv6 Address Record (AAAA)

- Syntax: “*name ttl class rr ipv6*”

```
; zone fragment for example.com
$TTL 2d ; zone default = 2 days or 172800 seconds
$ORIGIN example.com.
joe      IN      AAAA     2001:db8::3 ; joe & www = same ip
www      IN      AAAA     2001:db8::3
; functionally the same as the record above
www.example.com. AAAA     2001:db8::3
fred    3600  IN      AAAA     2001:db8::4 ; TTL overrides $TTL default
ftp        IN      AAAA     2001:db8::5 ; round robin with next
                IN      AAAA     2001:db8::6
squat      IN      AAAA     2001:db8:0:0:1::13 ; address in another subnet
```



Other Records (2)

• Name Server Record (NS)

- Syntax: “*name ttl class rr name*”

```
        IN      NS      ns1 ; unqualified name
; the line above is functionally the same as the line below
; example.com. IN      NS      ns1.example.com.
; at least two name servers must be defined
        IN      NS      ns2
; the in-zone name server(s) have an A record
ns1          IN      A      192.168.0.3
ns2          IN      A      192.168.0.3
```

• Mail Exchange Record (MX)

- Syntax: “*name ttl class rr pref name*”
- The **pref** (Preference) field is relative to any other MX record for the zone (value 0 to 65535). Low values are more preferred.

```
        IN      MX      10 mail ; short form
; the line above is functionally the same as the line below
; example.com. IN      MX      10 mail.example.com.
; any number of mail servers may be defined
        IN      MX      20 mail2.example.com.
; use an external back-up
        IN      MX      30 mail.example.net.
; the local mail server(s) need an A record
mail          IN      A      192.168.0.3
mail2         IN      A      192.168.0.3
```



Other Records (3)

• Canonical Name Record (CNAME)

- Syntax: “*name ttl class rr canonical_name*”

```
; zone fragment for example.com
$TTL 2d ; zone default = 2 days or 172800 seconds
$ORIGIN example.com.

...
server1    IN      A      192.168.0.3
www        IN      CNAME   server1
ftp         IN      CNAME   server1
```

- Do not use CNAME records with NS and MX records,
 - Usually it works, but is theoretically not permitted!

Wrong!

| | | | | |
|---------|----|-------|----|-------------------|
| | IN | MX | 10 | mail.example.com. |
| mail | IN | CNAME | | server1 |
| server1 | IN | A | | 192.168.0.3 |

Correct!

| | | | | |
|---------|----|-------|----|-------------------|
| | IN | MX | 10 | mail.example.com. |
| server1 | IN | CNAME | | mail |
| mail | IN | A | | 192.168.0.3 |



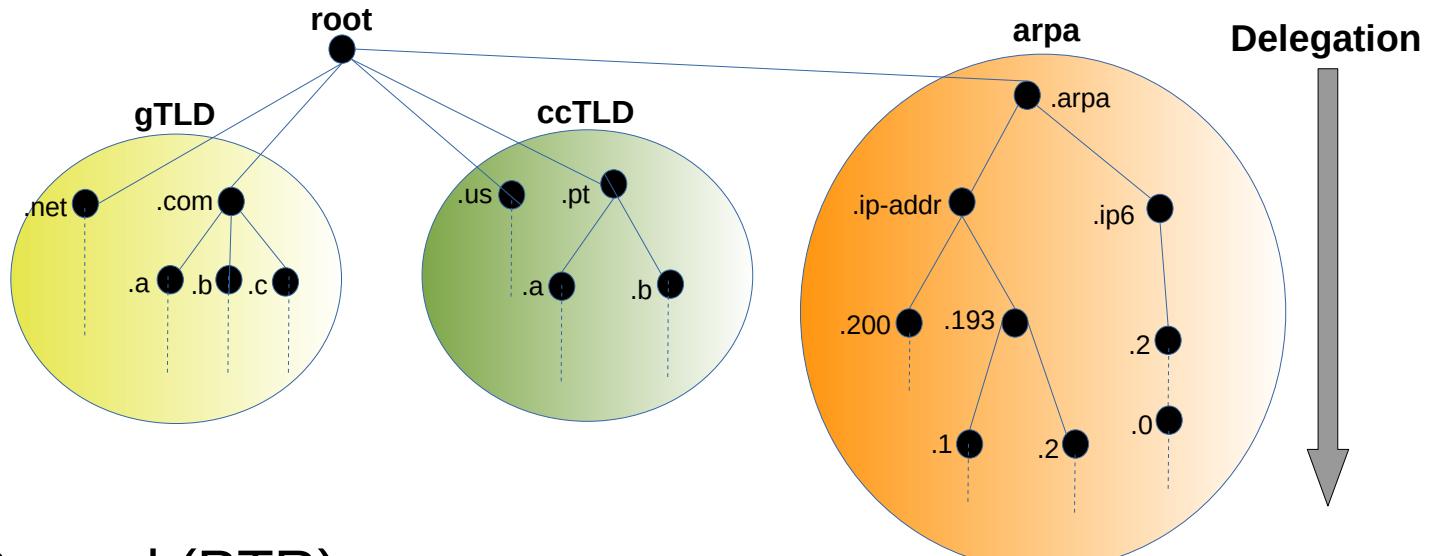
Example

```
$ORIGIN teste.com.  
@ IN SOA teste.com. adm.teste.com. (  
    199609206 ; serial, todays date + todays serial #  
    8H ; refresh, seconds  
    2H ; retry, seconds  
    4W ; expire, seconds  
    1D ) ; minimum, seconds  
NS ns1.teste.com.  
NS ns2.teste.com.  
MX 10 teste.com. ; Primary Mail Exchanger  
TXT "TESTE Corp"  
  
localhost A 127.0.0.1  
router A 206.6.177.1  
teste.com. A 206.6.177.2  
ns1 A 206.6.177.3  
ns2 A 206.6.177.4  
www A 207.159.141.192  
  
ftp CNAME teste.com.  
mail CNAME teste.com.  
news CNAME teste.com.  
  
funn A 206.6.177.2  
  
; Workstations  
ws-177200 A 206.6.177.200  
ws-177201 A 206.6.177.201
```



Reverse DNS

- In order to perform Reverse Resolution using normal recursive and Iterative queries the DNS designers defined a special (reserved) Domain Name called:
 - IN-ADDR.ARPA for IPv4 addresses,
 - Resolves <reversed_(partial)_IPv4_Address>.in-addr.arpa
 - IP6.ARPA for IPv6 addresses.
 - Resolves <reversed_(partial)_IPv6_Address>.ip6.arpa



- Uses the Pointer Record (PTR)
 - Pointer records are the opposite of A and AAAA.
 - Syntax: "name ttl class rr name"



IPv4 Reverse DNS - Example

```
zone "200.136.193.in-addr.arpa" {
    type master;
    file "zones/193.136.200";
};
```

```
$TTL 3D
@           IN      SOA     land-5.com. root.land-5.com. (
                           199609206      ; Serial
                           28800        ; Refresh
                           7200         ; Retry
                           604800       ; Expire
                           86400)       ; Minimum TTL
                           NS      land-5.com.
                           NS      ns2.psi.net.

;      Servers
1      PTR      router.land-5.com.
2      PTR      land-5.com.
2      PTR      funn.land-5.com.

;      Workstations
200    PTR      ws-177200.land-5.com.
201    PTR      ws-177201.land-5.com.
202    PTR      ws-177202.land-5.com.
203    PTR      ws-177203.land-5.com.
```



IPv6 Reverse DNS – Example

```
$TTL 2d      ; default TTL for zone 172800 secs
$ORIGIN 0.0.0.8.b.d.0.1.0.0.2.IP6.ARPA.

@       IN      SOA    ns1.example.com. hostmaster.example.com. (
                      2003080800 ; sn = serial number
                      12h        ; refresh = refresh
                      15m        ; retry = update retry
                      3w        ; expiry = expiry
                      2h        ; min = minimum
)
; name servers Resource Recordsfor the domain
      IN      NS      ns1.example.com.
; the second name servers is
; external to this zone (domain).
      IN      NS      ns2.example.net.
; PTR RR maps a IPv6 address to a host name
; hosts in subnet ID 1
1.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0      IN      PTR      ns1.example.com.
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0      IN      PTR      mail.example.com.
; hosts in subnet ID 2
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.0      IN      PTR      joe.example.com.
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.0      IN      PTR      www.example.com.
```

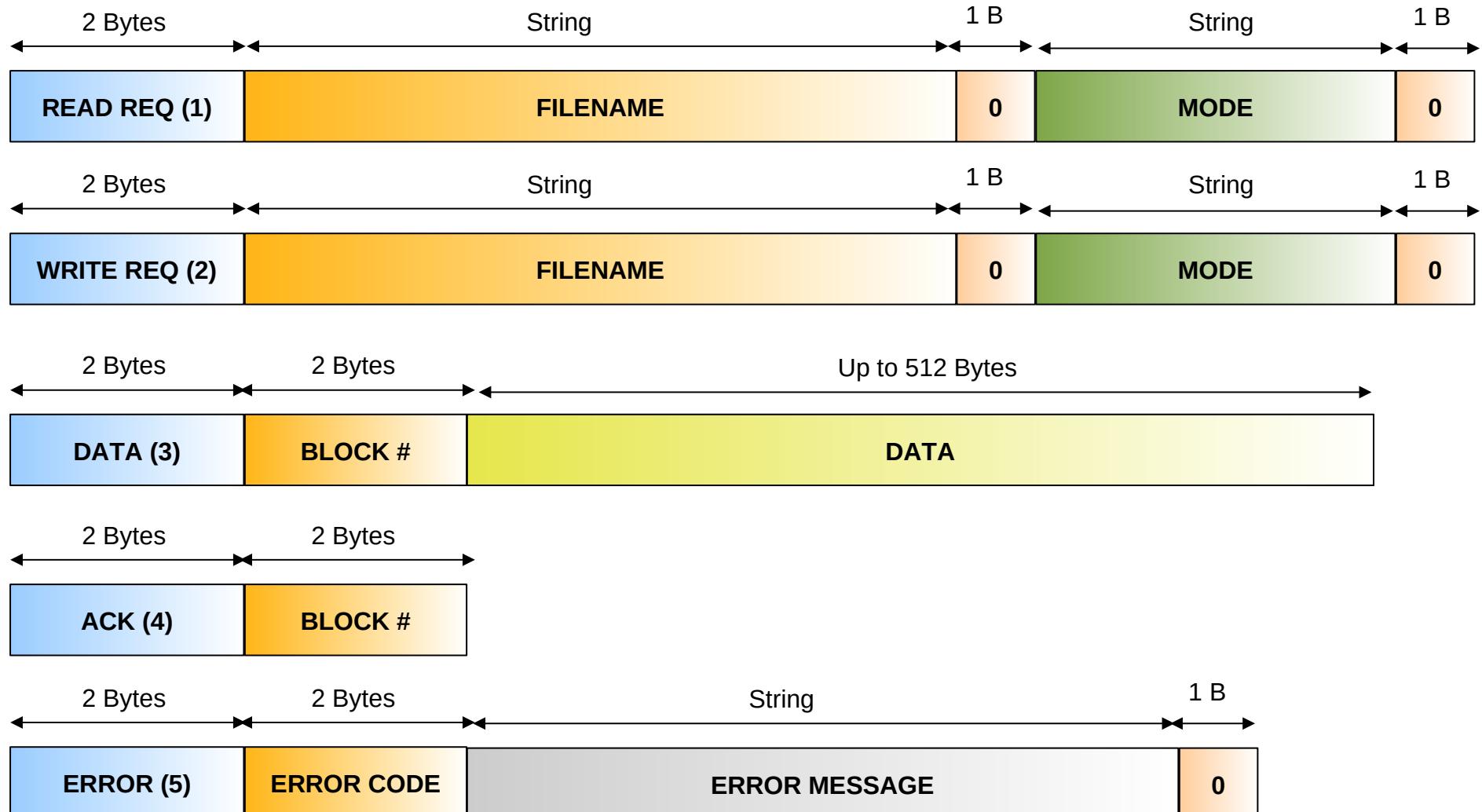


Trivial File Transfer Protocol (TFTP)

- Simple file transference service (IETF RFC 1350)
 - Does not allow directory/folder listing.
 - Does not incorporate any user authentication mechanism.
- Uses UDP.
 - The first packet from the client is sent to the server's port (default is port UDP 69).
 - The server chooses another (ephemeral) port number, and responds from that port.
 - Next client packets are sent to the second (chosen) server port.
- Implements “Stop and Wait” as flow control mechanism.
- Based on five primitives:
 - Read Request (RRQ)
 - Write Request (WRQ)
 - Data
 - Acknowledgement (ACK)
 - Error (ERR)



TFTP Messages Format (1)

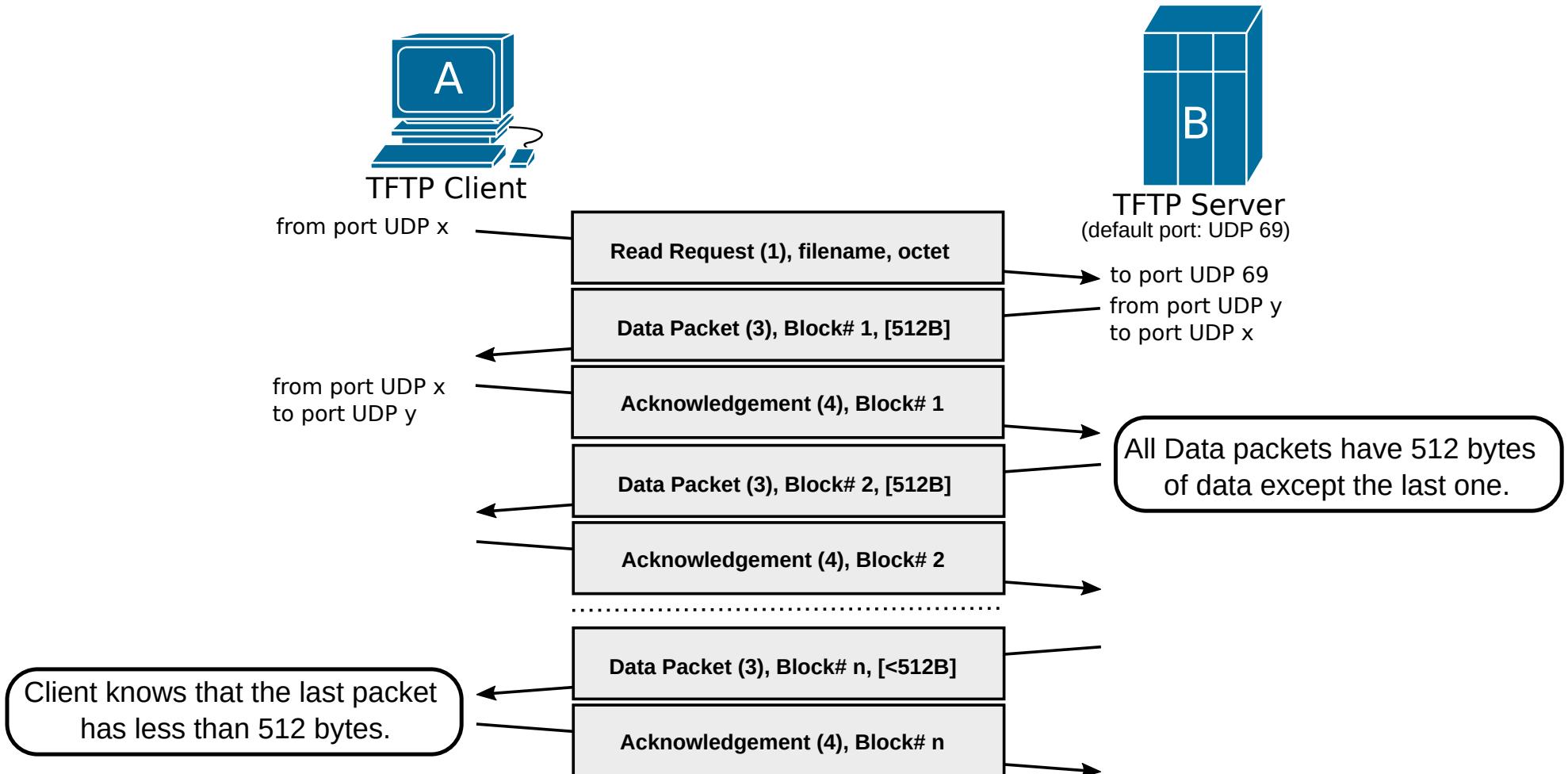


TFTP Messages Format (2)

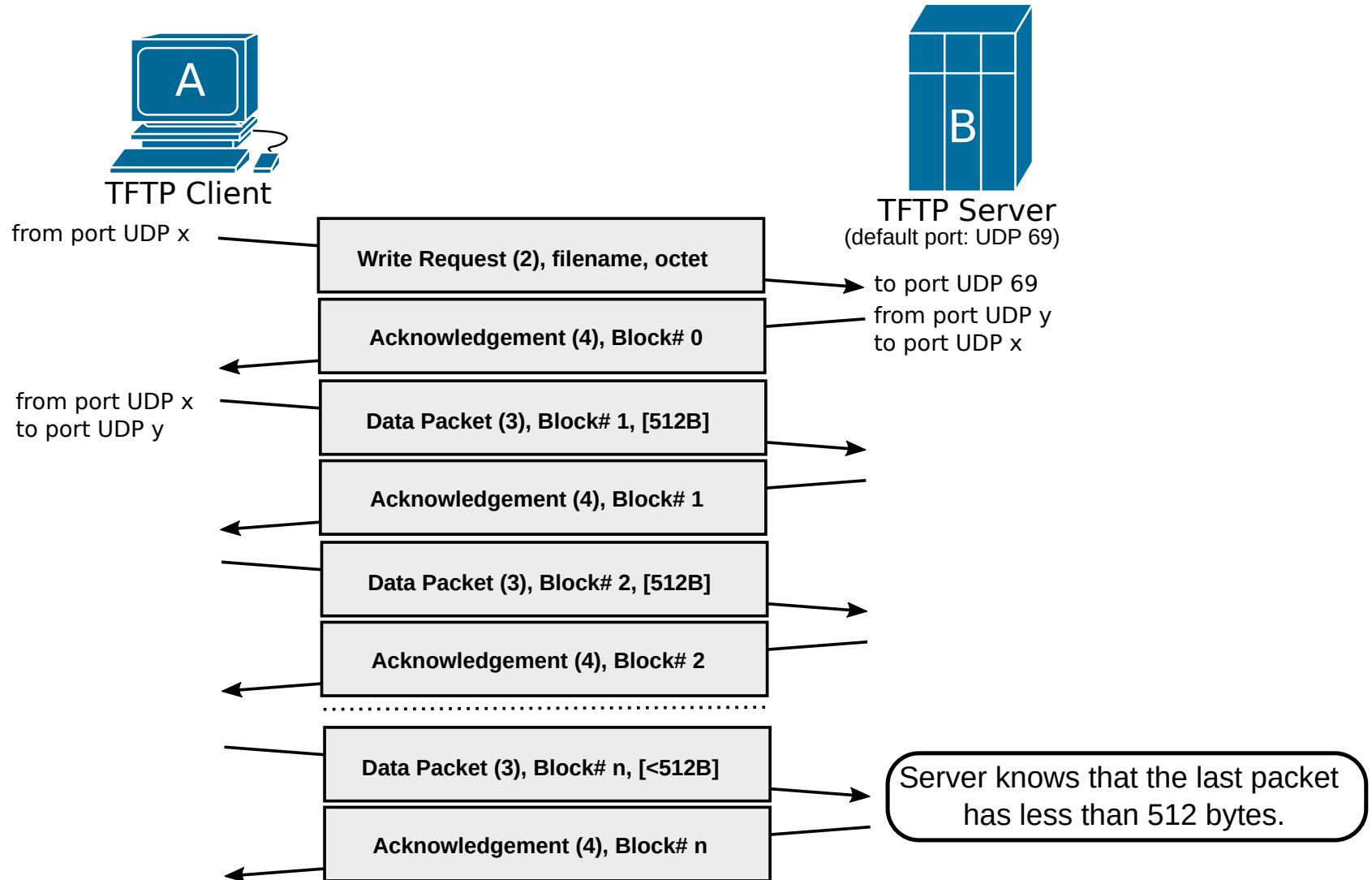
- Read and Write Request messages: Used to request a file download or upload, respectively.
 - FILENAME – ASCII character string that specifies the name of the file to write or read.
 - MODE – ASCII character string that specifies how the file will be exchanged (netascii or octet).
- Data message:
 - BLOCK # - Identifies the order/index of the data block.
- Acknowledge message:
 - BLOCK # - It is used to define which data block is being acknowledged.
- ERROR message: Acts as a NACK (not acknowledge). It may trigger a retransmission or end of connection.
 - ERROR MESSAGE – ASCII character string that specifies an error.
 - ERROR CODE: 00 – Not defined; 01 – File not found; 02 – Access violation; 03 – Disk full; 04 – Invalid operation code; 05 – Unknown port number; 06 – File already exists; 07 – No such user.



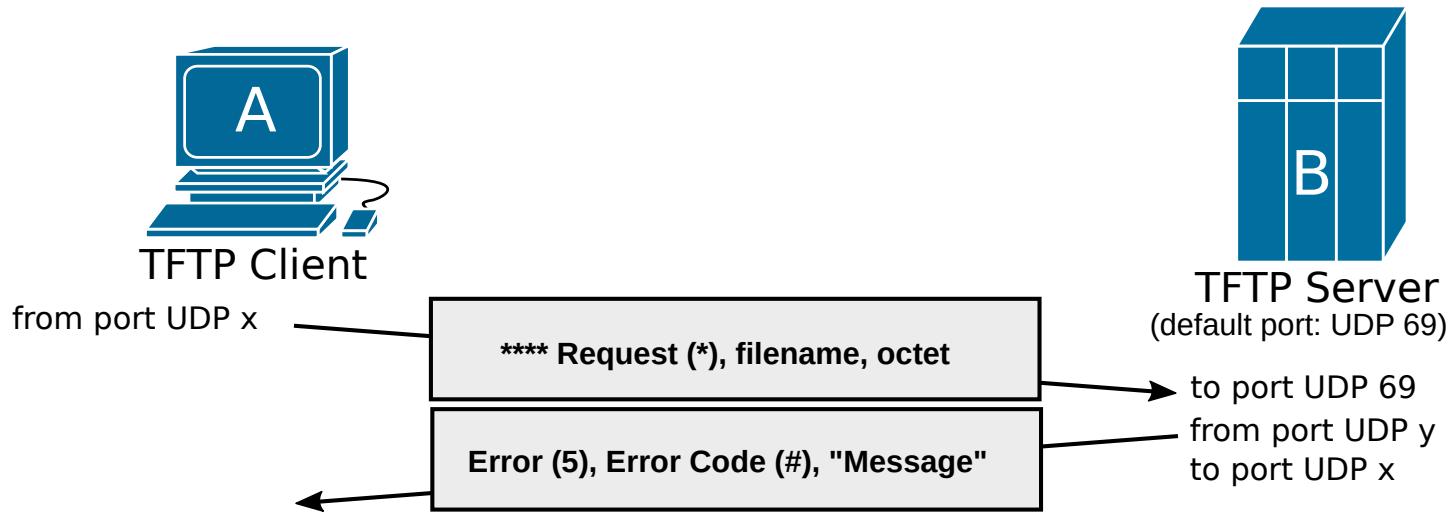
TFTP Read Request



TFTP Write Request



TFTP Errors



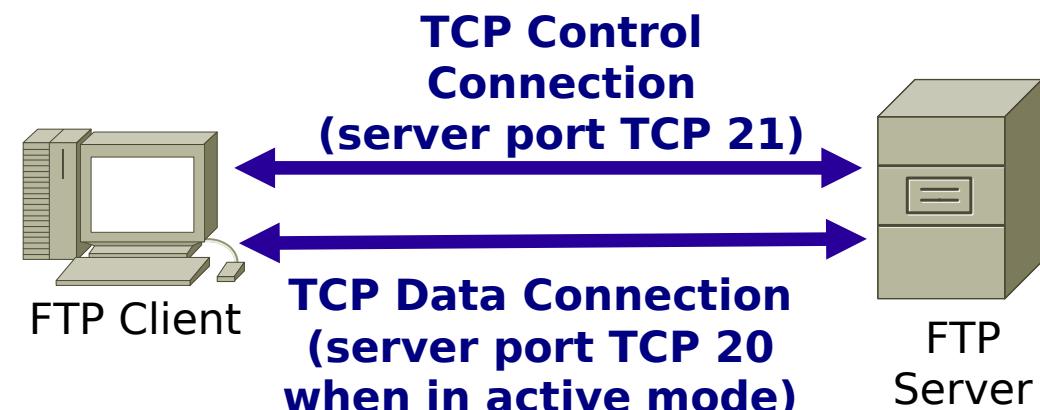
Error Codes

- **1** File not found.
- **2** Access violation.
- **3** Disk full or allocation exceeded.
- **4** Illegal TFTP operation.
- **5** Unknown transfer ID.
- **6** File already exists.



File Transfer Protocol (FTP)

- File Transference Service.
- Uses TCP.
- Server uses two TCP ports:
 - Control: TCP port 21.
 - Data: TCP port 20.
- Supports user authentication
 - Username + Password or Username Anonymous.
 - Credentials are transmitted in open text.
- The client establishes a TCP control connection with the server, by which the commands and responses are exchanged.
 - Commands and Responses are sent as ASCII text.
 - The TCP control connection will be maintained active until the end of the FTP session.
- Every time data must be exchanged, it is established a TCP data connection between the server and the client.
 - After each data set is exchanged, the TCP data connection is closed.
- Supports two data formats: ASCII and Binary (modes).
- Supports two data transfer modes:
 - Active: The server opens the TCP data connection to a client address and (ephemeral) port, announced by the client using a PORT command.
 - Passive: The client opens the TCP data connection to a server address and (ephemeral) port, announced by the server in a “227” response to a PASV command sent by the client.



FTP Control Requests and Responses

Sample Client Request commands:

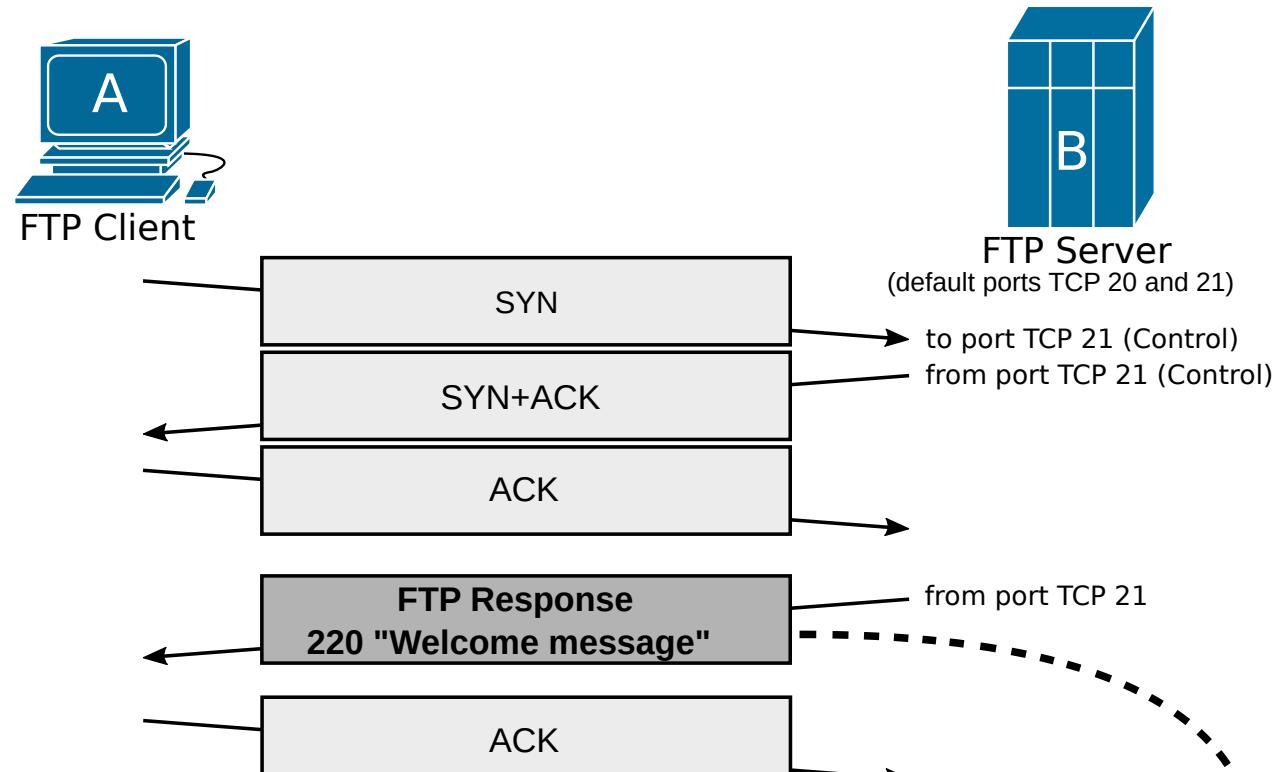
- **USER** Authentication username.
- **PASS** Authentication password.
- **TYPE** Sets the transfer mode (ASCII/Binary).
- **CDUP** Change to Parent Directory.
- **CWD** Change working directory.
- **PWD** Print working directory. Returns the current directory of the host.
- **LIST** Returns information of a file or directory if specified, else information of the current working directory is returned.
- **PASV** Enter passive mode.
- **PORT** Specifies an address and port to which the server should connect.
- **RETR** Retrieve a copy of the file
- **STOR** Accept the data and to store the data as a file at the server site
- **DELE** Delete file.
- **QUIT** Disconnect.

Sample Server Response codes

- **200** The requested action has been successfully completed.
- **220** Service ready for new user.
- **221** Service closing control connection.
- **226** Closing data connection. Requested file action successful.
- **227** Entering Passive Mode
- **230** User logged in, proceed.
- **331** Username OK, password required.
- **125** data connection already open transfer starting.
- **150** File status okay; about to open data connection.
- **425** Can't open data connection.
- **452** Error writing file.

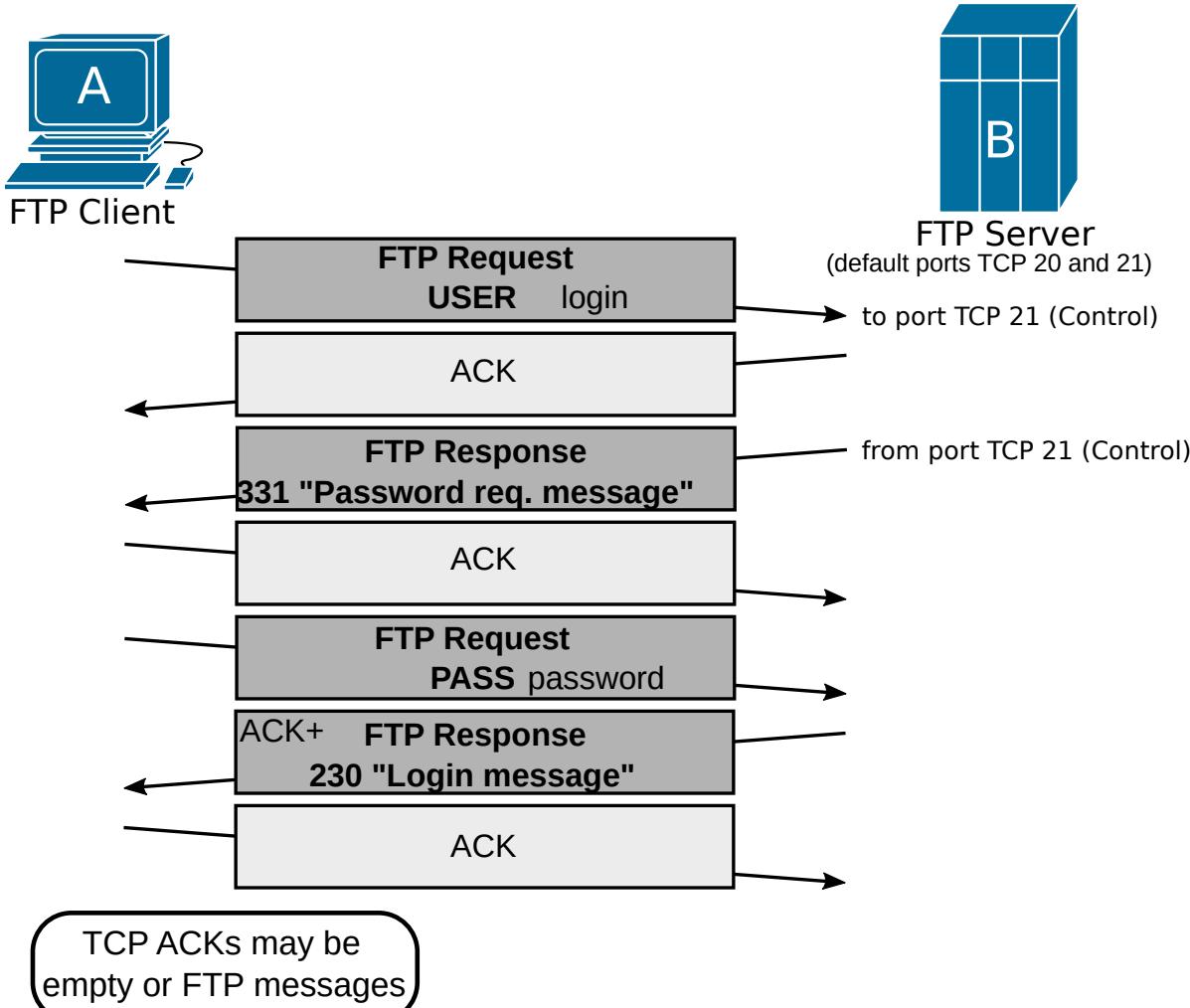


Initial FTP Connection



- ▶ Transmission Control Protocol, Src Port: 21, Dst Port: 59062
- ▼ File Transfer Protocol (FTP)
 - ▼ 220 (vsFTPD 3.0.2)\r\nResponse code: Service ready for new user (220)
 - Response arg: (vsFTPD 3.0.2)

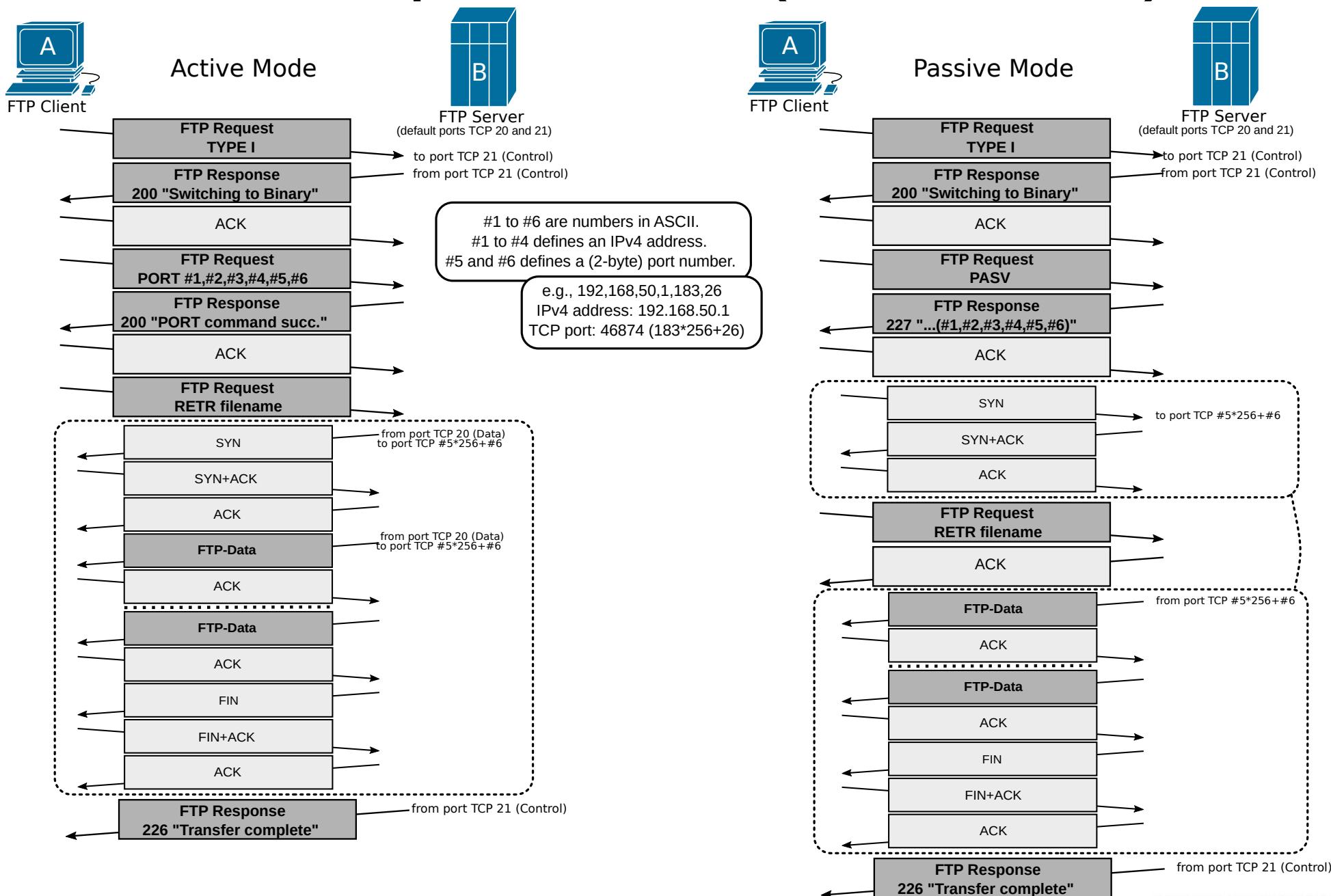
FTP Authentication



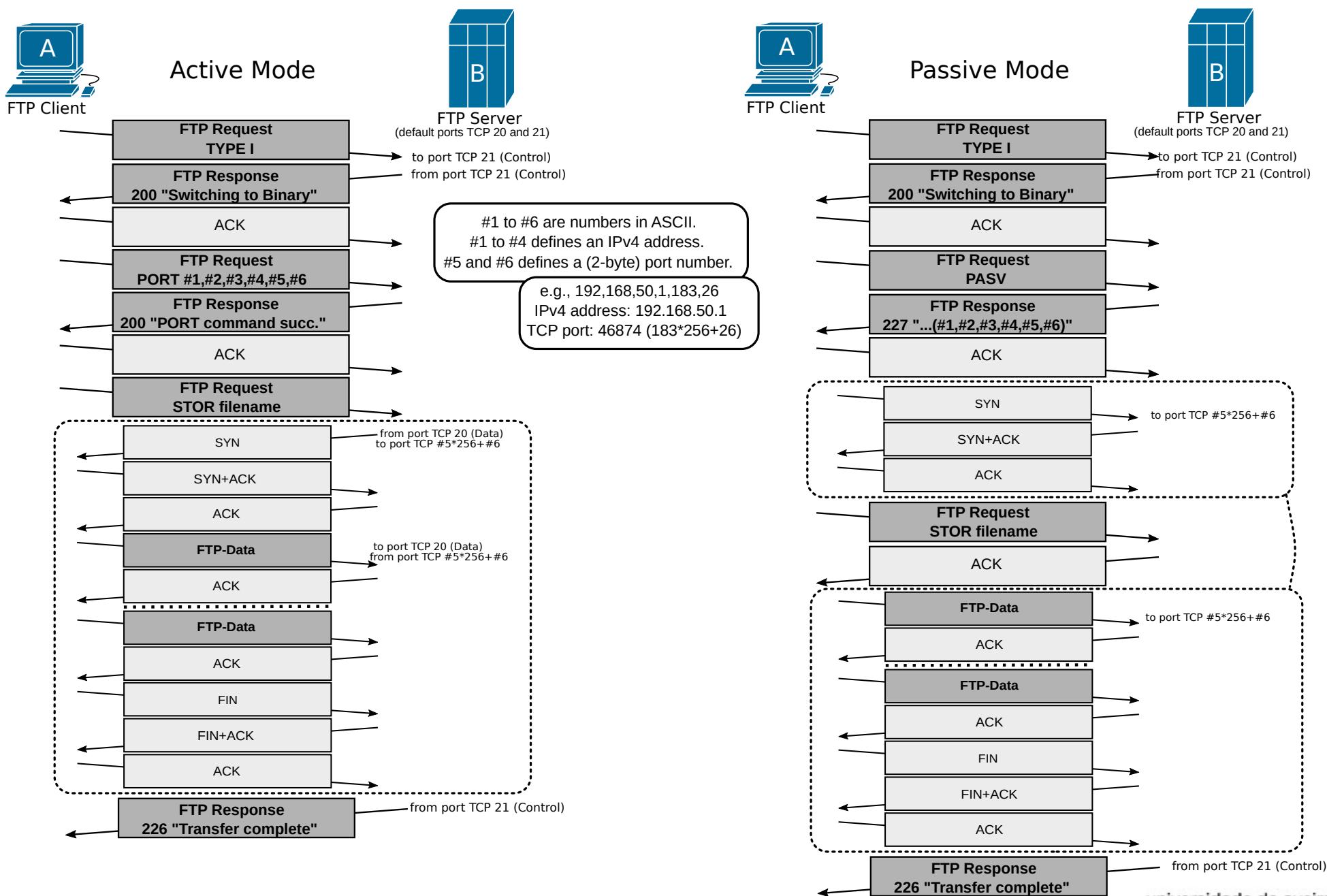
- ▶ Transmission Control Protocol, Src Port: 59062, Dst Port: 21
- └ File Transfer Protocol (FTP)
 - └ USER labcom\r\n Request command: USER
 - Request arg: labcom
- ▶ Transmission Control Protocol, Src Port: 21, Dst Port: 59062
- └ File Transfer Protocol (FTP)
 - └ 331 Please specify the password.\r\n Response code: User name okay, need password (331)
 - Response arg: Please specify the password.
- ▶ Transmission Control Protocol, Src Port: 59062, Dst Port: 21
- └ File Transfer Protocol (FTP)
 - └ PASS labcom\r\n Request command: PASS
 - Request arg: labcom
- ▶ Transmission Control Protocol, Src Port: 21, Dst Port: 59062
- └ File Transfer Protocol (FTP)
 - └ 230 Login successful.\r\n Response code: User logged in, proceed (230)
 - Response arg: Login successful.



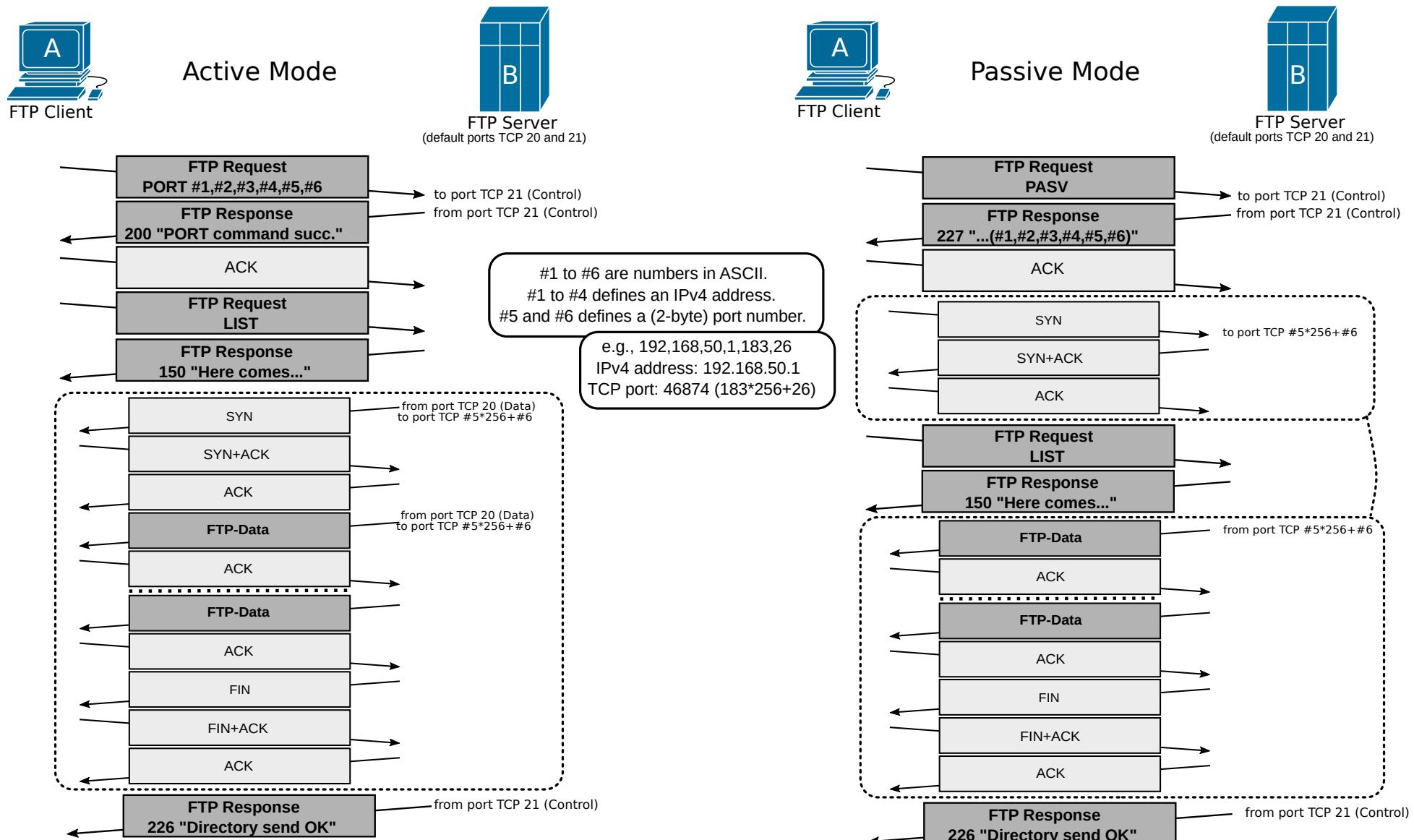
FTP Operations (Download)



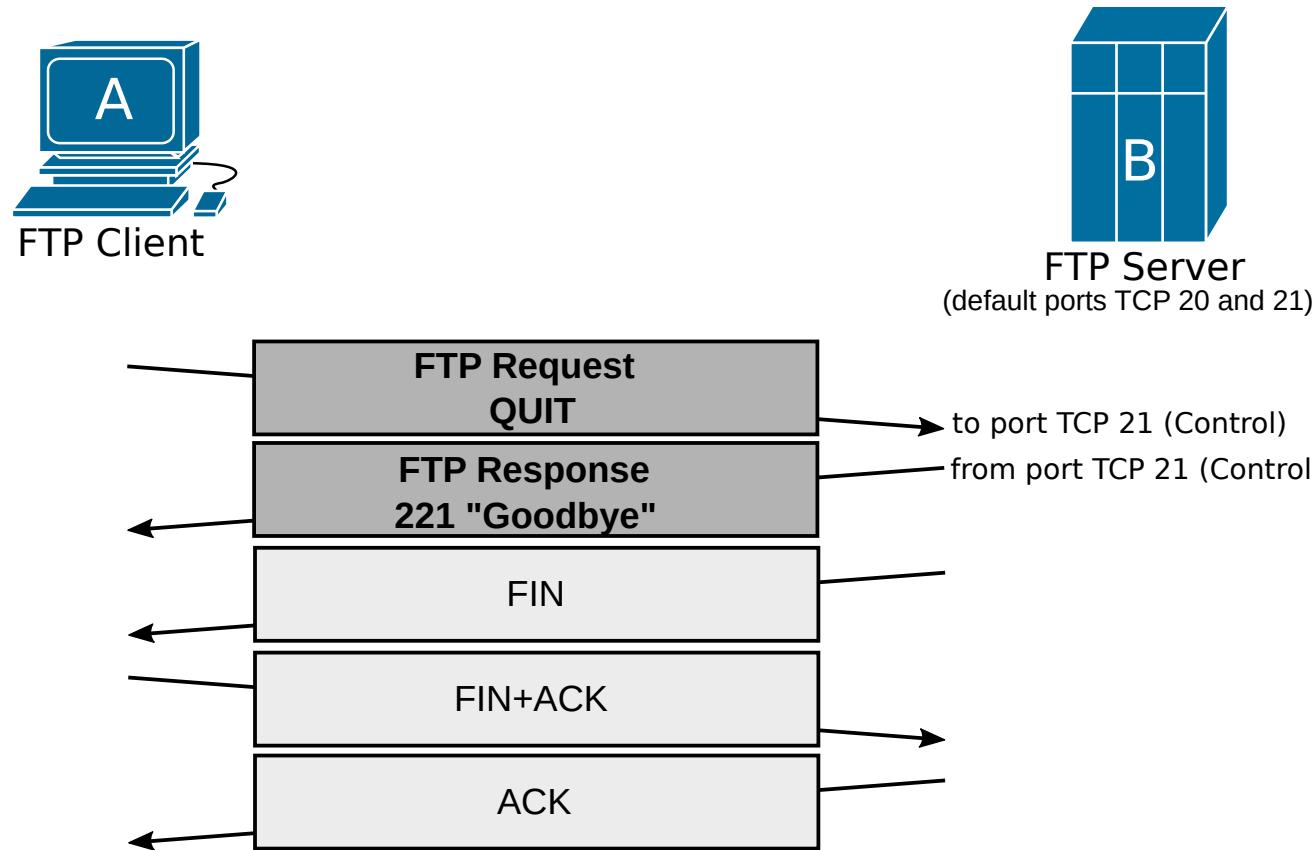
FTP Operations (Upload)



FTP Operations (Directory Listing)

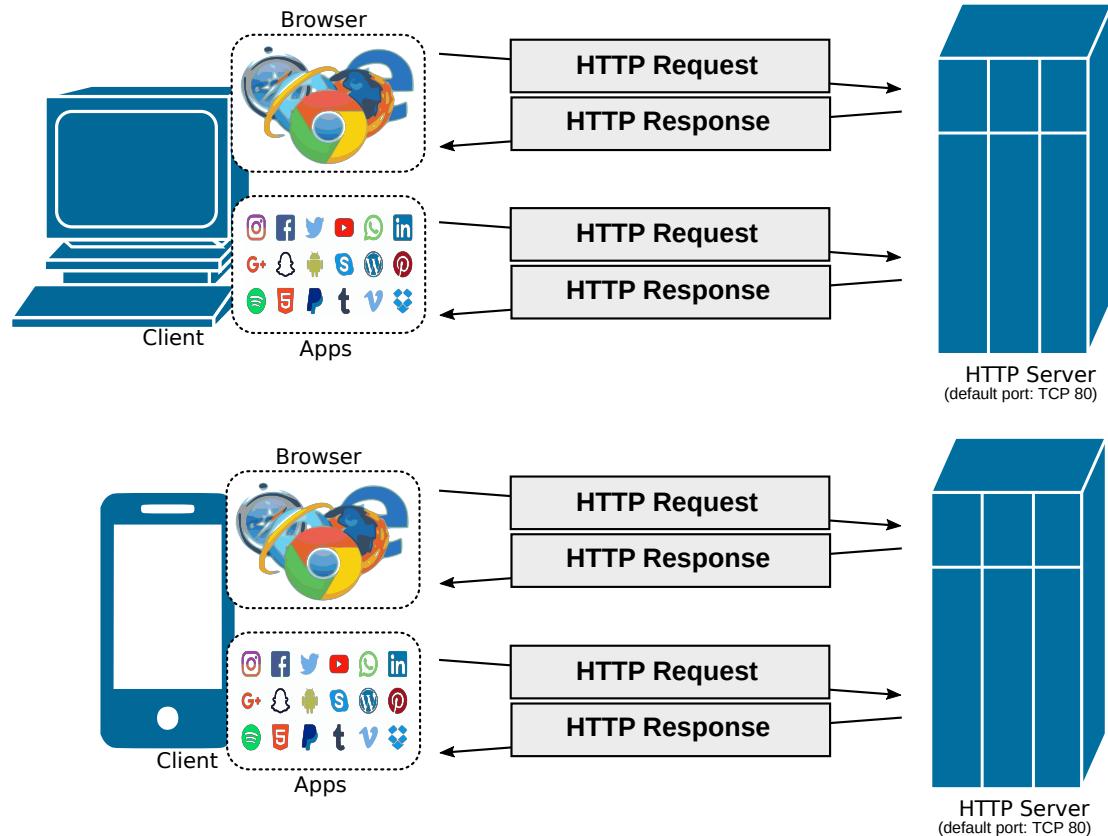


End of FTP Connection



Hyper-Text Transport Protocol (HTTP)

- Web's application layer protocol.
- Follows a client/server model.
 - Client:
 - Browser that requests, receives, and “displays” Web objects.
 - Application that sends and receives data.
 - Server: Web server sends data objects in response to requests. Also, receives data from clients.
- Client side sends HTTP Requests to server.
- Server responds with HTTP Responses.



HTTP Connections

- HTTP uses TCP.
 - With non-persistent connections.
 - The client establishes a TCP connection to send the Request and the server terminates the TCP connection after sending the Response.
 - Performance is penalized by the establishment time of each TCP connection and its slow start behavior.
 - Possibility of using parallel TCP connections (configurable number in browsers) to decrease server response time.
 - With persistent connections.
 - The client establishes a TCP connection to send the Request, the Request includes a directive indicating to the server not terminate the TCP connection after sending the Response.
 - The client can use established connections with pipelining or without pipelining.
 - Pipelining=Sending multiple requests without waiting for the responses of previous requests.
 - The server waits for a timeout (typically configurable) time to terminate connections that are not used and were not closed.



HTTP Versions

HTTP 1.0 was defined in RFC 1945

- Supports only non-persistent connections.
- Motivation to use it nowadays:
 - Low complexity web/data objects.
 - Limited server capabilities.

HTTP 1.1 was defined in RFC 2616

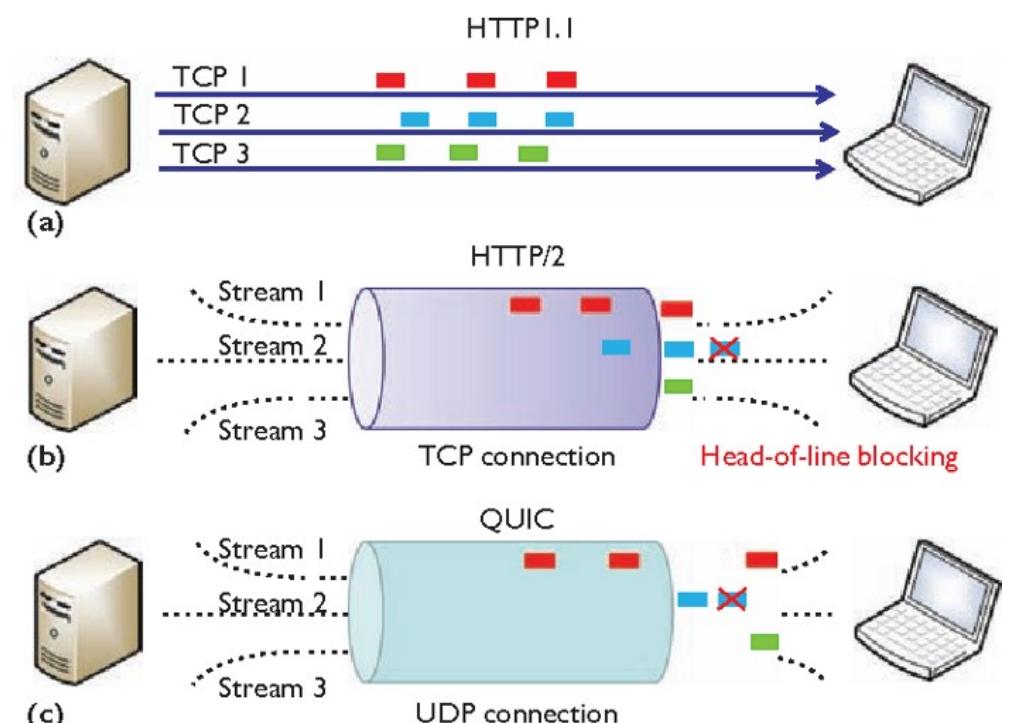
- By default, works with persistent TCP connections and pipelining.
- Compatible with version 1.0.

HTTP 2.0 was defined in RFC 7540 (2015)

- Derived from Google's SPDY protocol.
- Data compression of HTTP headers.
- Requests prioritization.
- Server Push
 - Server sends data before client request.
- Highly compatible with version 1.1.

HTTP 3.0 (in draft)

- Based on Google's QUIC.
- Multiple data streams over an UDP connection.
- Faster establishment.
- Does not block data streams after one packet loss.
- Usable in (very) low packet losses networks.

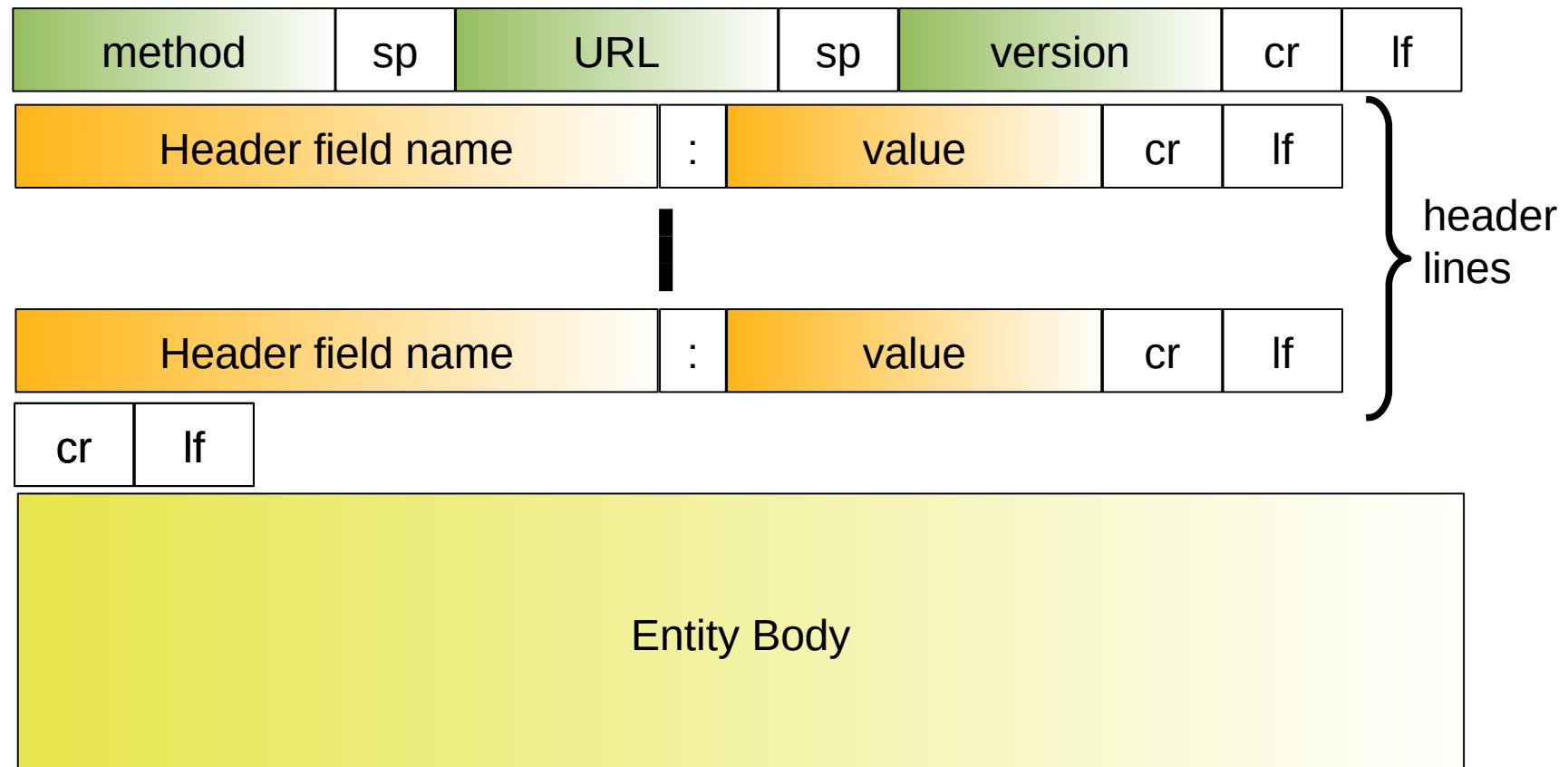


Uniform Resource Locator (URL)

- `http://www.someschool.edu:1024/somedir/page.html`
 - `http://`
 - Protocol used to communicate with server.
 - Protocols supported by browsers: http, https, file, ftp, mailto.
 - `www.someschool.edu`
 - DNS name associated with the server's IP address.
 - Can be an IP address (193.136.173.5 ou www.ua.pt)
 - `:1024/`
 - Indicates the server's port (optional)
 - If omitted, the default port is used.
 - `somedir/`
 - Path for the data object (optional)
 - If omitted, it is assumed that the data object is in root folder.
 - `page.html`
 - Name of the data object.



HTTP Request Format



HTTP Methods

- **GET**

- Requests a representation of the specified resource.
- Requests using GET should only retrieve data.

- **HEAD**

- Asks for a response identical to that of a GET request, but without the response body.

- **POST**

- Used to submit an entity to the specified resource, often causing a change in state or side effects on the server

- **PUT**

- Replaces all current representations of the target resource with the request payload.

- **DELETE**

- Deletes the specified resource.

- **CONNECT**

- Establishes a tunnel to the server identified by the target resource.

- **OPTIONS**

- Used to describe the communication options for the target resource.



HTTP Request (sample)

```
GET /PageText.aspx?id=259 HTTP/1.1\r\n
Host: www.ua.pt\r\n
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-GB; rv:1.9.0.10)
Gecko/2009042523 Ubuntu/9.04 (jaunty) Firefox/3.0.10\r\n
    Accept: text/html,application/xhtml+xml,application/xml;\r\n
    Accept-Language: en-gb,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    Referer: http://www.ua.pt/\r\n
```

- Messages composed in ASCII format.
- Start with a line defining the *method* (GET, POST, HEAD, ...).
- Includes a variable number of header lines:
 - Host: identifies the server.
 - Connection: Defines a persistent (keep-alive) or non-persistent connection.
 - User-agent: identifies and describes the client OS and Browser (i.e., Firefox, Linux Ubuntu 9.04).



HTTP Response (sample)

```
HTTP/1.1 200 OK
Connection:close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0
Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT
Content-Length: 6821
Content-Type: text/html
(carriage return, line feed)
(data, data, data, ...)
```

- First line contains the HTTP version and Response Code.
- Includes a variable number of header lines.
- Ends with the requested content.
 - Before defines the length and type of the content.



HTTP Response Codes

- 200 OK
 - Request succeeded, requested object later in this msg.
- 301 Moved Permanently
 - Requested object moved, new location specified later in this msg (Location:).
- 400 Bad Request
 - Request msg not understood by server.
- 404 Not Found
 - Requested document not found on this server.
- 505 HTTP Version Not Supported



HTTP Native Authentication

- HTTP includes natively an authentication process that allows to limit access to files based on a username and password.
- A request message sent by a browser to a protected file is answered by the server with a response message in which the response line is:
 - 401 Authorization Required
- This response includes a header line of the type **WWW-Authenticate** indicating the authentication method to use.
- The new request includes an **Authorization** header line with the username and password information generated by the method requested by the server.
- Typically, the browser stores the username and password information in memory for use in future request messages.
- Nowadays, the authentication is handle at the application level.



HTTP Cookies

- Cookies are a way for the server to identify a terminal in different requests made over time.
- Allows the server to differentiate the information to be made available to the client.
- The first time a terminal sends a request to a server, the server includes in the response a header line of type:
 - Set-Cookie: uu = ad14cc7cf29d446b0b8e73c5606135efcbe4e58c; expires = Wed, 17-Jun-2009 15:47:29 GMT \r \n
- If the browser is configured to accept cookies, it saves this number along with the server identifier.
- In future orders, the browser includes the header line:
 - Cookie: uu = ad14cc7cf29d446b0b8e73c5606135efcbe4e58c \r \n
- In this way, the server identifies the client.



Conditional GET

- If the browser supports Web caching, it is possible to:
 - Minimize response times,
 - Minimize network traffic.
- If a file is cached on the client, the browser makes a request with a header line of type **If-modified-since**.
- If not modified, the server just responds with a **304** response code.

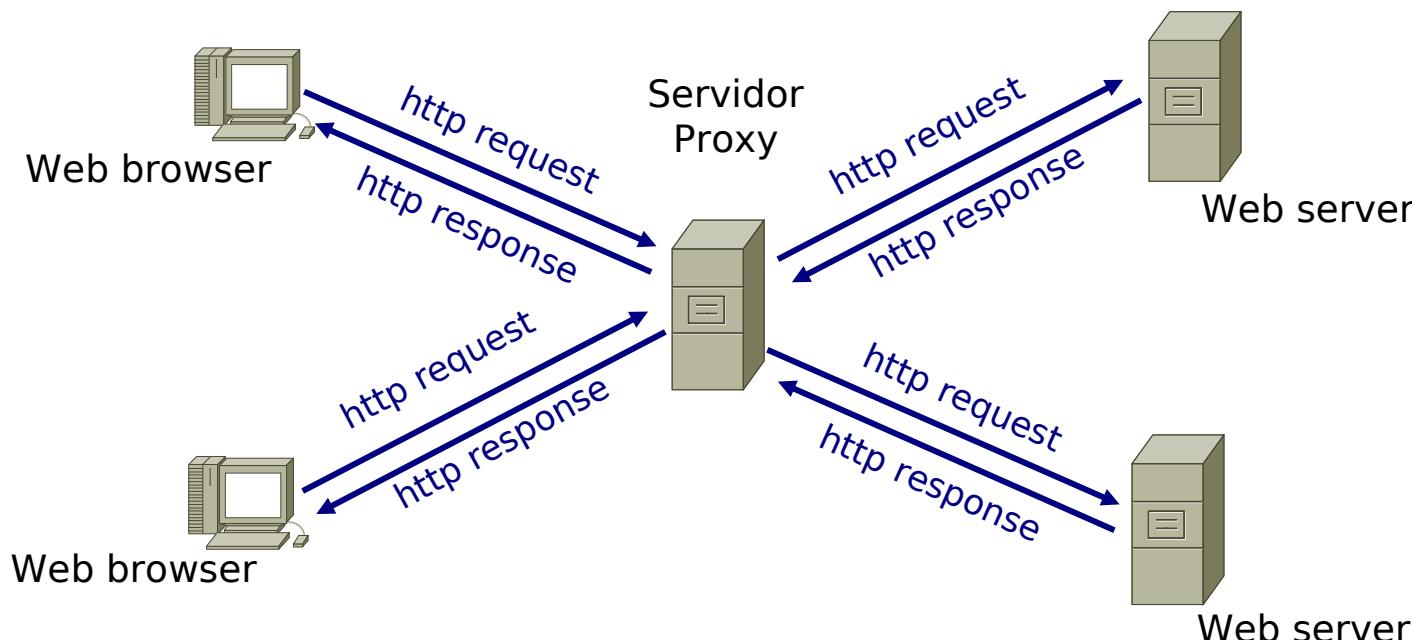
```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
If-modified-since: Mon, 22 Jun 1998 09:23:24 GMT
(carriage return, line feed)
```

```
HTTP/1.1 304 Not Modified
Date: Thu, 19 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0
(carriage return, line feed)
```



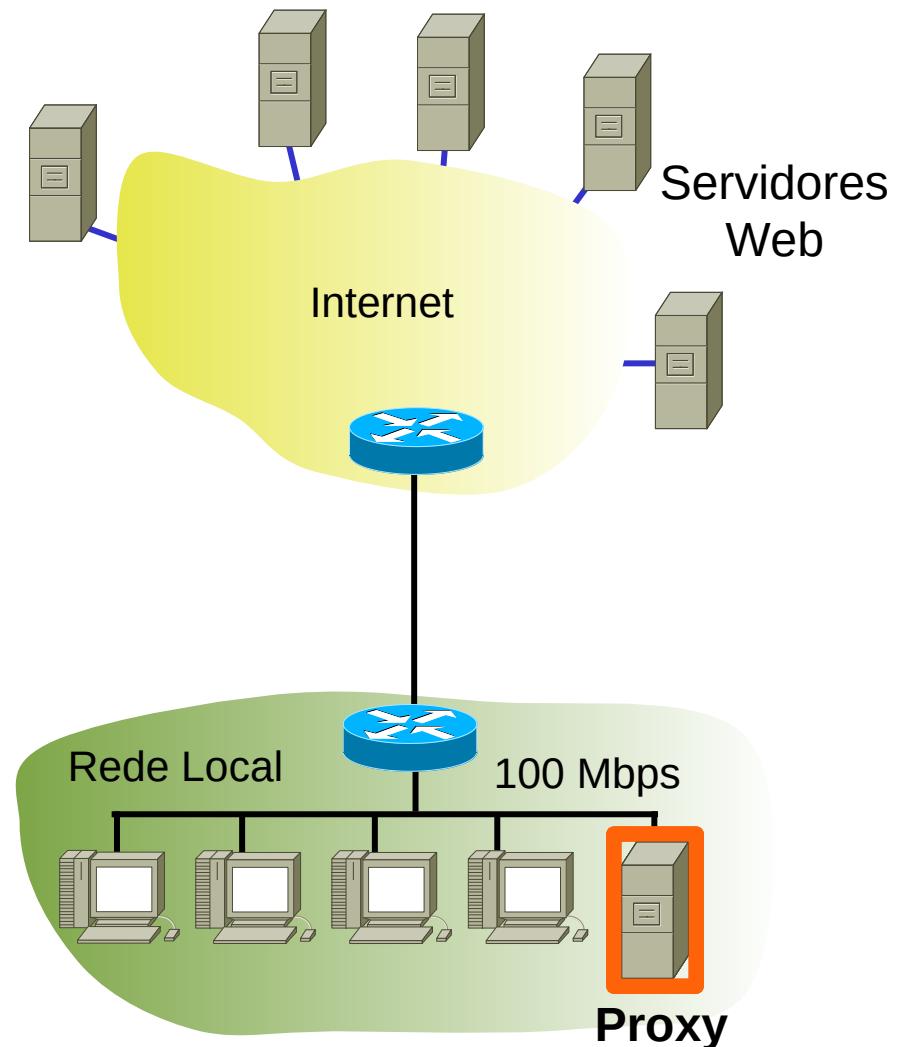
HTTP Proxy Servers (1)

- Acts as an intermediate element between the client and the server:
 - The client interacts with the Proxy Server as if it were the Web server.
 - The Proxy Server interacts with the Web servers on behalf of the clients (for the Web server, the Proxy Server is the client).
- The Proxy server stores all files requested by clients (up to the limit of their storage capacity)
- When a client requests a file that already exists on the Proxy server, it does not have to be requested again from the Web server (it only sends a conditional get message).



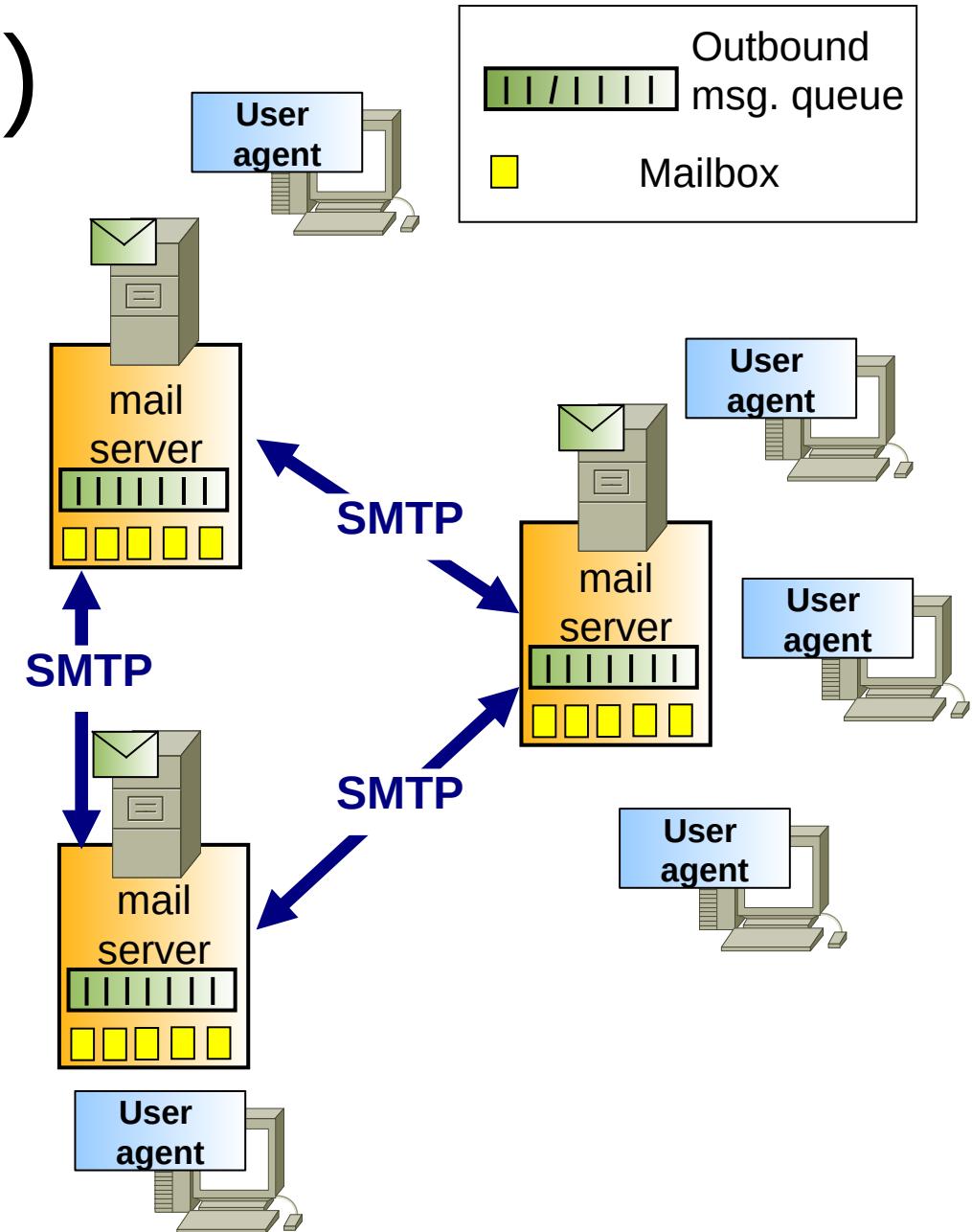
HTTP Proxy Servers (2)

- Proxy servers in companies or institutions:
 - Decrease interaction times.
 - Reduce traffic to the public network.
- Proxy servers on Internet Service Providers (ISPs):
 - They allow an infrastructure of automatic distribution of the most requested Web contents.
 - Usually the ISP Proxy is closer to the client than the server.
 - Response times are lower.
- Nowadays, the more dynamic web contents make proxies less relevant.



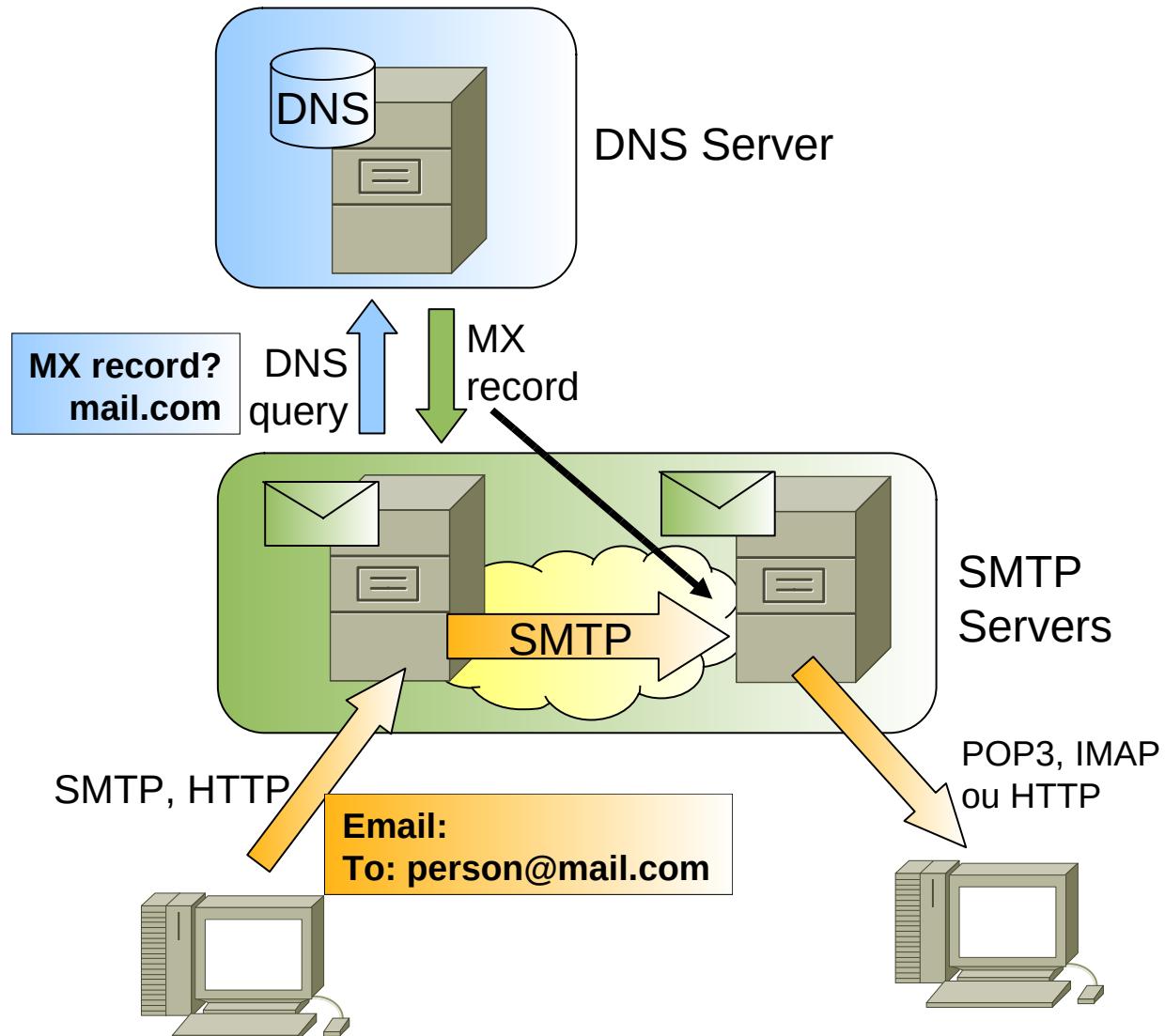
Electronic Mail (e-Mail)

- User agents: application in which the user writes, sends, receives and reads e-mail messages.
 - The user agent exchanges messages with your mail server.
- Mail server: an e-mail server that sends and receives messages to and from clients, and to and from other servers.
- The mail server includes:
 - A mailbox for each client where the messages are stored.
 - A queue of outgoing messages with messages from its clients customers that have not yet been sent.



E-Mail Protocols

- Message Forwarding (between mail servers)
 - SMTP (Simple Mail Transfer Protocol)
- Message Sending (from user-agent to client server)
 - SMTP (Simple Mail Transfer Protocol)
 - HTTP (Hyper-Text Transport Protocol)
- Mailbox access (from server to user-agent)
 - POP3 (Post Office Protocol – version 3)
 - IMAP (Internet Mail Access Protocol)
 - HTTP (Hyper-Text Transport Protocol)



Simple Mail Transfer Protocol (SMTP)

- It runs over TCP and the default port number of the server is TCP 25.
- The communication is established by the entity that wants to send information (push protocol).
- Direct communications: by default, the sender's mail server sends the messages directly to the recipients mail server.
- The protocol follows a client/server philosophy:
 - The client issues commands.
 - The server responds to commands.
 - Like HTTP, each response consists of a 3-digit code followed by an optional string.
- E-mail in 7-bit ASCII format ending with "CRLF.CRLF"
 - CR- carriage return, LF- line feed



E-Mail Messages Format

```
From: antonio@av.it.pt  
To: bruno@ua.pt  
Subject: Tens fome?
```

```
Boas!  
A que horas vamos comer?  
Antonio  
.
```

- Messages in ASCII format.
- They start with a set of header lines followed by an empty line followed by the body of the message:
 - Some lines (From and To) are required in the original message.
 - Other header lines (Subject, etc ...) are optional.
 - Some header lines (Received, etc ...) are inserted by mail servers.



SMTP Sample Interaction

```
Server: 220 mail.ua.pt
Client: HELO mail.av.it.pt
Server: 250 Hello mail.av.it.pt
Client: MAIL FROM: <antonio@av.it.pt>
Server: 250 antonio@av.it.pt... Sender ok
Client: RCPT TO: <bruno@ua.pt>
Server: 250 junior@det.ua.pt... Recipient ok
Client: DATA
Server: 354 Enter mail, end with "." on a line by itself
Client: Boas!
Client: A que horas vamos comer?
Client: Antonio
Client: .
Server: 250 Message accepted for delivery
Client: QUIT
Server: 221 mail.ua.pt closing connection
```



Multipurpose Internet Mail Extensions (MIME)

- For non-ASCII data.
- Allows you to send messages of different types of information
- Adds the following header lines:
 - Content-Transfer-Encoding – defines the algorithm for encoding the information content in ASCII format
 - Exemples: base64
 - Content-type: type/subtype; parameters – defines the type of information
 - Exemples: text/plain; charset="ISO-8859-1"
 - text/html
 - image/gif
 - image/jpeg
 - video/mpeg
 - video/quicktime
 - application/msword



SMTP Messages with MIME

```
From: antonio@av.it.pt
To: bruno@ua.pt
Subject: Imagem fixe!
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

(base64 encoded data .....
.....
..... base 64 encoded data)
```

- At the user agent of the receiver, the content of the message is:
 - Decoded by the base64 algorithm to obtain the original content, and
 - Delivered to a JPEG decoder to view the sent image.



Multipart MIME Extentions

```
From: antonio@av.it.pt
To: bruno@ua.pt
Subject: Imagem fixe!
MIME-Version: 1.0
Content-Type: multipart/mixed; Boundary=98766789
```

--98766789

Content-Type: text/plain

Olá Bruno,
Junto envio a fotografia combinada.

Texto

--98766789

Content-Transfer-Encoding: base64

Content-Type: image/jpeg

(base64 encoded data
.....
..... base 64 encoded data)

Imagen

--98766789

Content-Type: text/plain

Antonio

Texto

- The multipart / mixed header line allows to compose a message with multiple types of information.
- Its Boundary parameter identifies the separation between different types of information in the body of the message.



Mensagem SMTP recebida pelo destinatário

```
Received: from mail.meo.pt by mail.ua.pt; 12 Oct 17 15:30:01
GMT
Received: from mail.av.it.pt by mail.meo.pt; 12 Oct 17 15:27:39
GMT
From: antonio@av.it.pt
To: carlos@meo.pt
Subject: Imagem fixe!
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

(base64 encoded data .....
..... base 64 encoded data)
```

- Each mail server inserts a Received header line that identifies the server it sent, the server it received, and the instant of time the message was received.
 - In the example, the carlos user configured his server mail.meo.pt to forward the messages to the server mail.ua.pt.



Post Office Protocol – version 3 (POP3)

- It runs over TCP and the defaults server port number is TCP 110.
- The communication is established by the entity (user agent) that wishes to receive information (pull protocol).
- Message transfer is done in one of two ways:
 - send-and-remove: messages are removed from the server's mailbox after being sent.
 - send-and-store: messages are kept in the mailbox after being sent.
- The protocol is executed in 3 phases:
 - authentication: the user agent sends the user name and password.
 - transaction: the mail server sends the messages that are in the mailbox of the user; the user agent indicates for each message whether or not it should be removed from the mailbox.
 - update: mail server removes from the mailbox the messages indicated by the user agent for removal.



POP3 Commands

- **USER userid**
- **PASS password**
- **STAT**
 - The response contains the number of messages and the total size in bytes of the messages.
 - Sample: +OK 3 345910
- **LIST**
 - The answer is a list, where each line identifies the number and size in bytes of each message. Ends with a line with only one dot.
 - Sample:
 - +OK 3 messages
 - 1 1205
 - 2 305
 - 3 344400
- **RETR msg#**
 - Retrieves the message with number msg#
 - Example: RETR 2
- **DELE msg#**
 - Deletes message with number msg#
 - Example: DELE 3
- **RSET**
 - Clears all marked messages.
- **QUIT**



POP3 Interaction (sample)

- Authentication phase:

- user agent Commands:
 - user: username
 - pass: password
 - mail server Responses:
 - +OK
 - -ERR

- Transaction phase, user agent:

- list: Lists messages; with the number and size of each one.
 - retr: Retrieves a message based on its number.
 - dele: Deletes a message based on its number.
 - quit: Terminates the POP3 interaction.

```
S: +OK POP3 server ready
C: user antonio
S: +OK
C: pass hungry
S: +OK user successfully logged on
```

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 2 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```



Internet Mail Access Protocol (IMAP)

- It runs over TCP and the server port number is TCP 143.
- IMAP allows the user additional important functionalities:
 - Create and manage a messaging directory system on the server; do search operations on the directory system.
 - Requesting the sending of portions of the mail messages.
- In one session, the server is in one of 4 states:
 - Unauthenticated status: the initial state before the user agent sends the user's name and password.
 - Authenticated state: The user agent must identify a directory before sending any command that affects mail messages.
 - Status selected: user agent can send message management commands (view, remove, transfer, etc ...).
 - Logout status: When the session ends.



IMAP (Client) Commands

- CAPABILITY
- LOGIN
- SELECT
 - Selects a folder/directory within the mailbox.
 - Same as EXAMINE.
- CREATE/DELETE/RENAME
 - Creates/deletes/renames a folder/directory in the mailbox.
- LIST
 - Lists folders/directories within the mailbox.
- SUBSCRIBE/UNSUBSCRIBE
 - Change the active state of a folder/directory.
- STATUS
 - Verifies the state of a folder/directory.
- FETCH
 - Retrieves a message or folder (fully or partially).
- LOGOUT
- Others: APPEND, EXPUNGE, SEARCH, STORE, COPY, ...



VoIP Voice (and Video) over IP

Voice over IP

- Network loss: IP datagram lost due to network congestion (router buffer overflow).
- Delay loss: IP datagram arrives too late for playout at receiver.
 - Delays: processing, queueing in network; end-system (sender, receiver) delays.
 - Typical maximum tolerable delay: 400 ms.
- Loss tolerance: depending on voice encoding, packet loss rates between 1% and 10% can be tolerated.
- Speaker's audio: alternating talk/speech with silent periods.
 - 64 kbps during talk/speech.
 - Packets generated only during talk/speech.
 - 20 msec chunks at 8 Kbytes/sec: 160 bytes data.
- Requires session establishment.
- VoIP protocols/frameworks:
 - Session Initiation Protocol (SIP)
 - Session Description Protocol (SDP)
 - H.323
- VoIP and PSTN interoperability in large/ISP scalable scenarios require complex control frameworks:
 - Media Gateway Controller Protocol (MGCP);
 - H.248/Megaco.



Session Initiation Protocol (SIP)

- Defined by RFC 3261.
- Designed for creating, modifying and terminating sessions between two or more participants.
 - Not limited to VoIP calls.
- Is a text-based protocol similar to HTTP.
 - Transported over UDP or TCP protocols.
 - Security at the transport and network layer provided with TLS (requires TCP) or IPSec.
- Offers an alternative to the complex H.323 protocols.
- Due to its simpler nature, the protocol is becoming more popular than the H.323 family of protocols.
- SIP is a peer-to-peer protocol. The peers in a session are called user agents (UAs):
 - User-agent client (UAC) - A client application that initiates the SIP request.
 - User-agent server (UAS) - A server application that contacts the user when a SIP request is received and that returns a response on behalf of the user.
- A SIP endpoint is capable of functioning as both UAC and UAS.



SIP Functionality

- SIP supports five facets of establishing and terminating multimedia communications:
 - User location - determination of the end system to be used for communication;
 - User availability - determination of the willingness of the called party to engage in communications;
 - User capabilities - determination of the media and media parameters to be used;
 - Session setup - "ringing", establishment of session parameters at both called and calling party;
 - Session management - including transfer and termination of sessions, modifying session parameters, and invoking services.



SIP Clients and Servers

• SIP Clients

- Phones (software based or hardware).
- Gateways
- User Agents
- A User Agent acts as a
 - Client when it initiates a request (UAC),
 - Server when it responds to a request (UAS).

• SIP Servers

- Proxy server
 - Receives SIP requests from a client and forwards them on the client's behalf.
 - Receives SIP messages and forward them to the next SIP server in the network.
 - Provides functions such as authentication, authorization, network access control, routing, reliable request retransmission, and security.
- Redirect server
 - Provides the client with information about the next hop or hops that a message should take and then the client contacts the next-hop server or UAS directly.
- Registrar server
 - Processes requests from UACs for registration of their current location.
 - Registrar servers are often co-located with a redirect or proxy server.



SIP Messages

- SIP used for Peer-to-Peer Communication though it uses a Client-Server model.
- SIP is a text-based protocol and uses the UTF-8 charset.
- A SIP message is either a **request** from a client to a server, or a **response** from a server to a client.
 - A request message consists of a Request-Line, one or more header fields, an empty line indicating the end of the header fields, and an optional message-body;
 - A response message consists of a Status-Line, one or more header fields, an empty line indicating the end of the header fields, and an optional message-body.
 - All lines (including empty ones) must be terminated by a carriage-return line-feed sequence (CRLF).



SIP Requests

- Requests are also called “Methods”.
- SIP uses SIP Uniform Resource Indicators (URI) to indicate the user or service to which a request is being addressed.
- The general form of a SIP Request-URI is:
 - `sip:user:password@host:port;uri-parameters`
 - `sip:John@doe.com`
 - `sip:+14085551212@company.com`
 - `sip:alice@atlanta.com;maddr=239.255.255.1;ttl=15`
 - Proxies and other servers route requests based on Request-URI.
- Requests are distinguished by starting with a Request-Line.
 - A Request-Line contains a **Method** name, a **Request-URI**, and **SIP-Version** separated by a single space (SP) character.
 - Request-Line = Method SP Request-URI SP SIP-Version CRLF
 - RFC 3261 defines six methods: INVITE, ACK, OPTIONS, BYE, CANCEL, and REGISTER.
 - SIP extensions provide additional methods: SUBSCRIBE, NOTIFY, PUBLISH, MESSAGE, ...
 - SIP-Version should be “SIP/2.0”.
 - Example:
 - Request-Line: INVITE sip:2001@192.168.56.101 SIP/2.0
- The remaining of a request message is one or more header fields, an empty line indicating the end of the header fields, and an optional message-body.

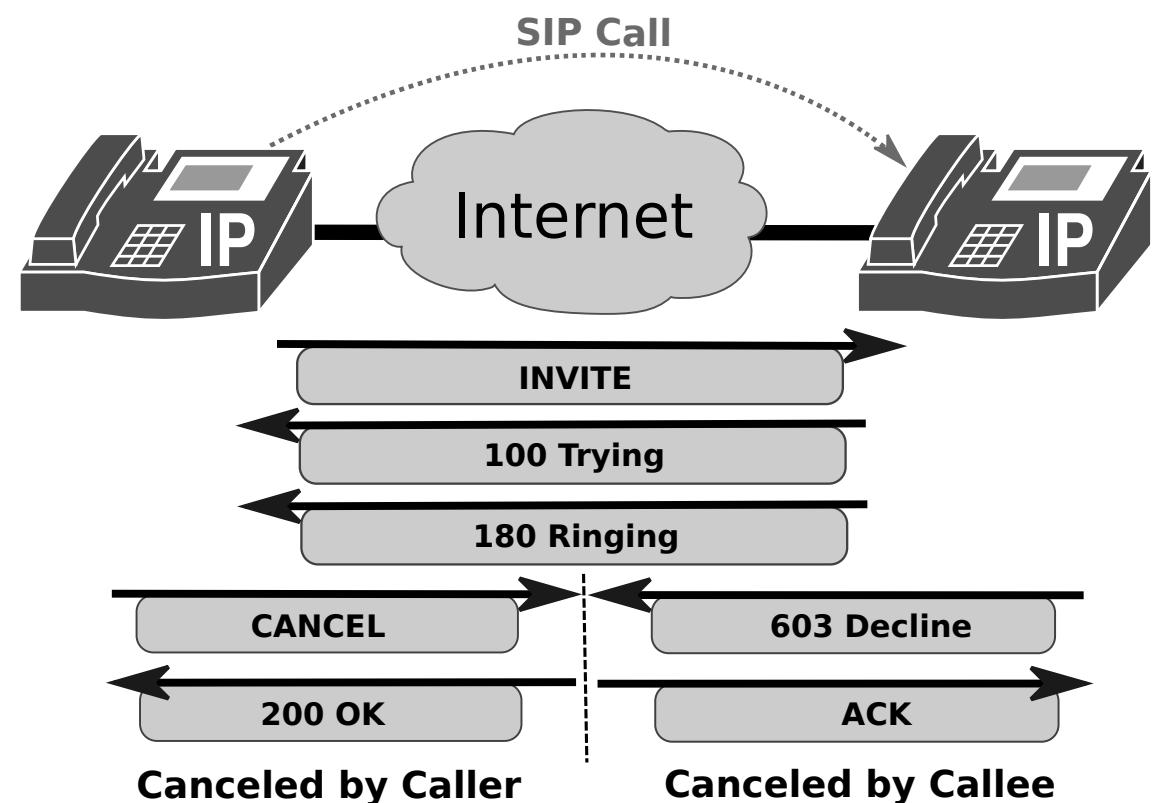
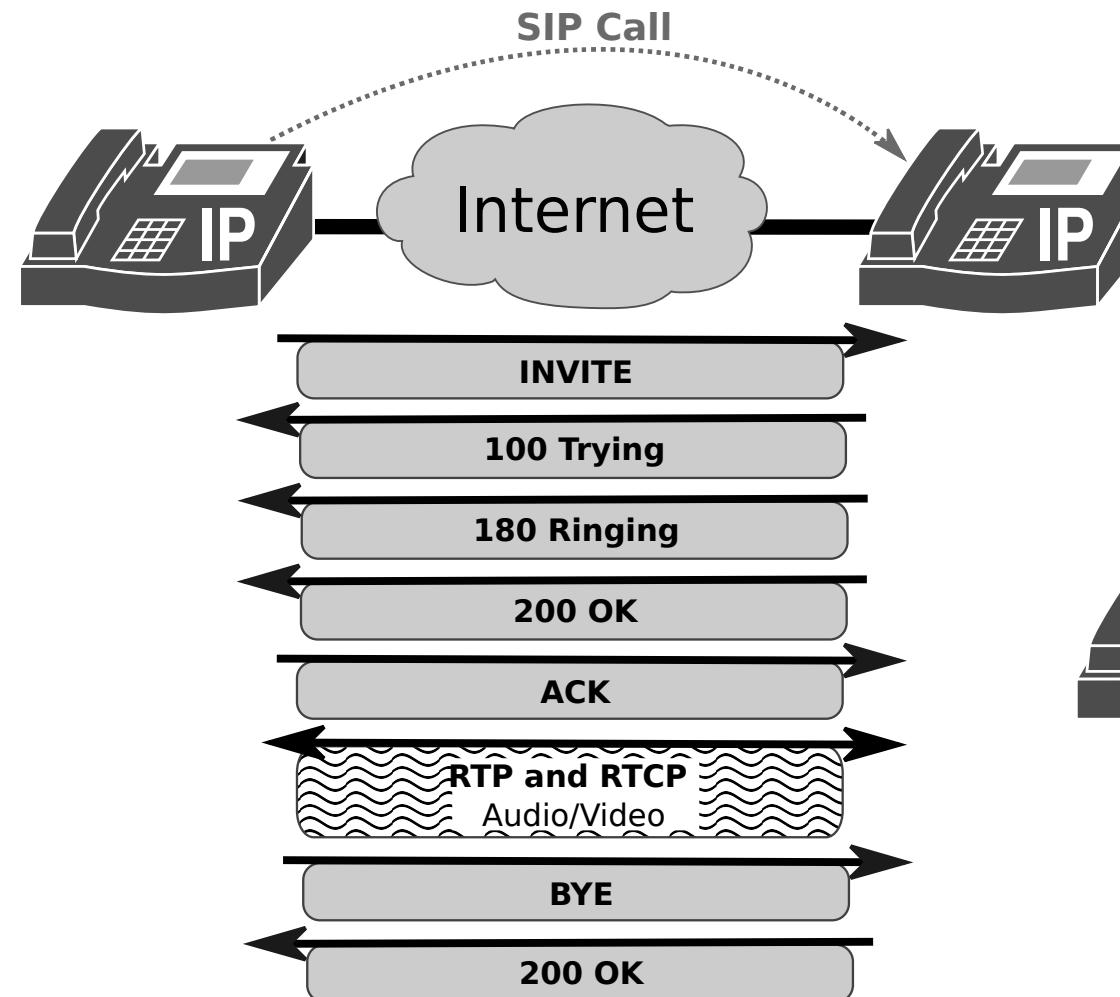


Session Description Protocol (SDP)

- SIP carries (encapsulates) SDP messages.
- When initiating multimedia teleconferences, VoIP calls, streaming video, or other sessions, is required to transmit to participants media details, transport addresses, and other session description metadata.
- SDP (RFC 4566) provides a standard representation for such information, irrespective of how that information is transported.
 - SDP is purely a format for session description.
 - SDP is intended to be general purpose so that it can be used in a wide range of network environments and applications.
 - SDP does not support negotiation of session content or media encodings.

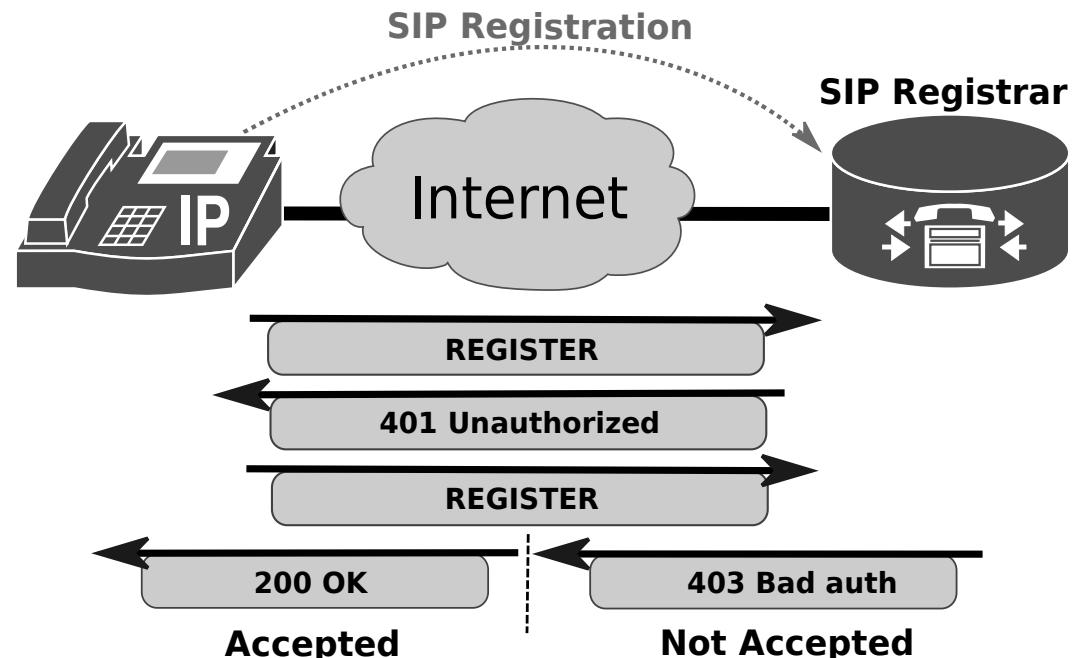


SIP Signaling – Direct Call



SIP Registrar Server

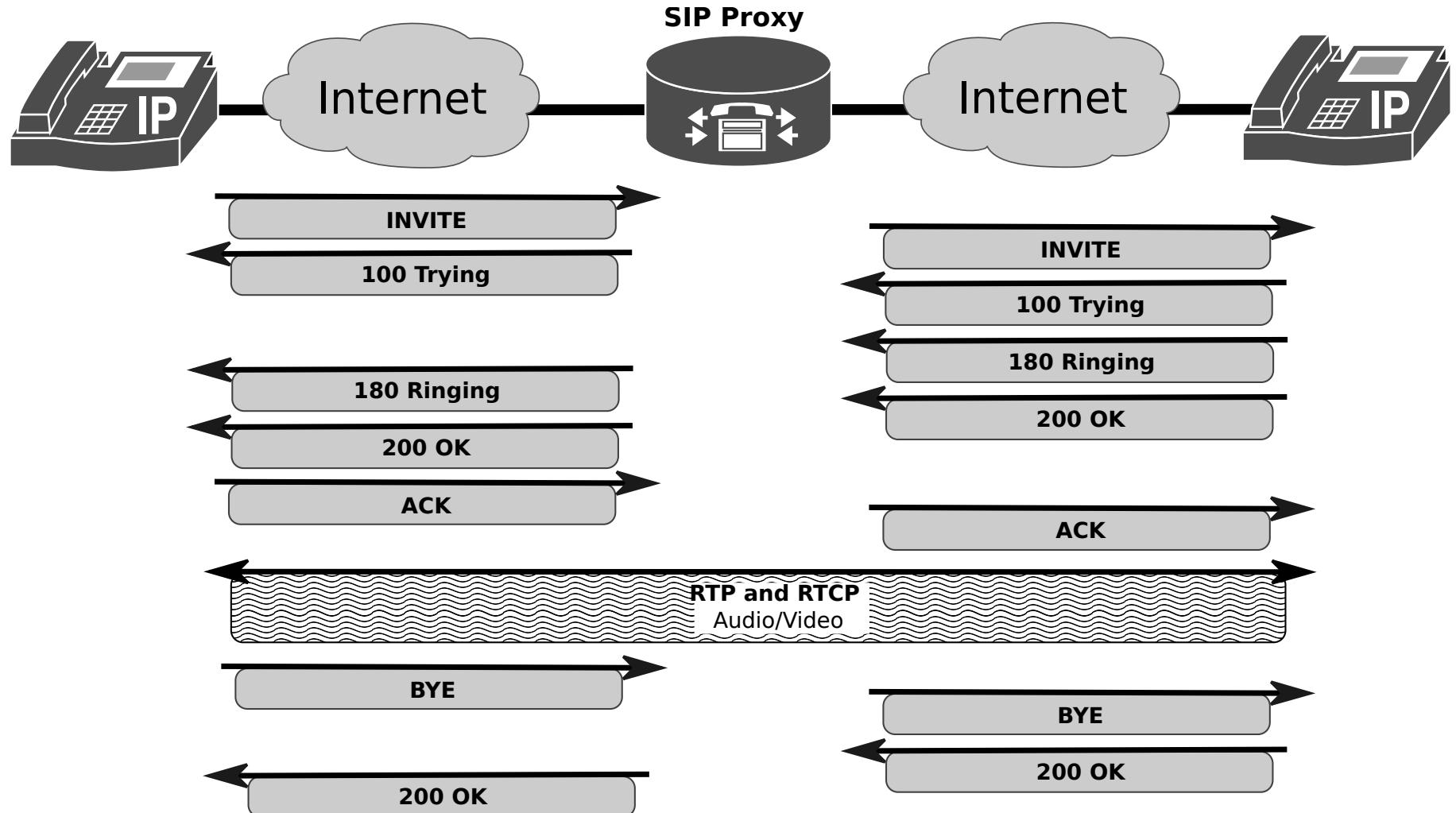
- SIP Registrar servers store the location of SIP endpoints.
- A user has an account created which allows them to REGISTER contacts with a particular server.
- The account specifies a SIP “Address of Record (AOR)”
- Each SIP endpoint Registers with a Registrar server with a SIP REGISTER request.
 - Using its Address of Record and Contact address.
- Address of Record is in From header:
 - From: <sip:Vieira@192.168.56.102>
- Contact header tells Registrar server where to send messages:
 - Contact: <sip:Vieira@192.168.56.1:5060>
- SIP Proxy servers query SIP Registrar servers for routing information.



- Registration usually requires authentication.
- If REGISTER has no authentication credentials, the SIP Registrar server responds with 401 Unauthorized.
- End-point resends REGISTER with an Authorization header with credentials.
 - Authorization: Digest
username="Vieira", realm="asterisk",
nonce="7d88f81c",
uri="sip:2001@192.168.56.102",
algorithm=MD5,
response="b70474b5bbece20a68472e7ad4e37197"
- Server accepts registration with a 200 OK response.
- Server rejects credentials with a 401 Bad Auth response.



SIP Proxy Server

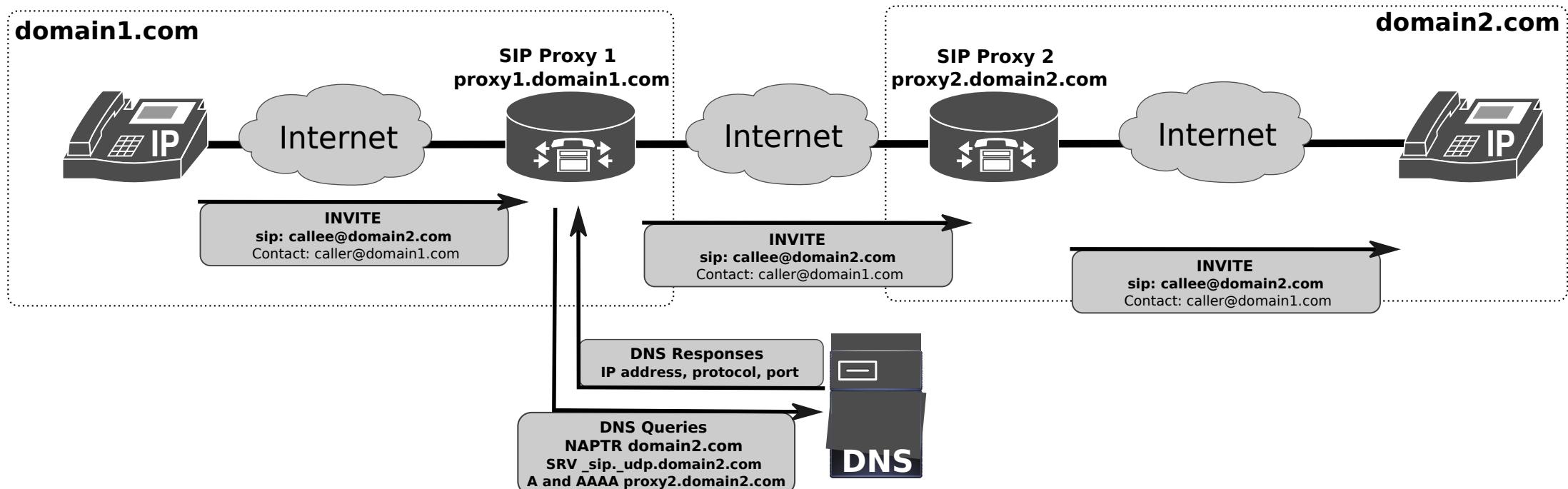


Locating SIP Servers

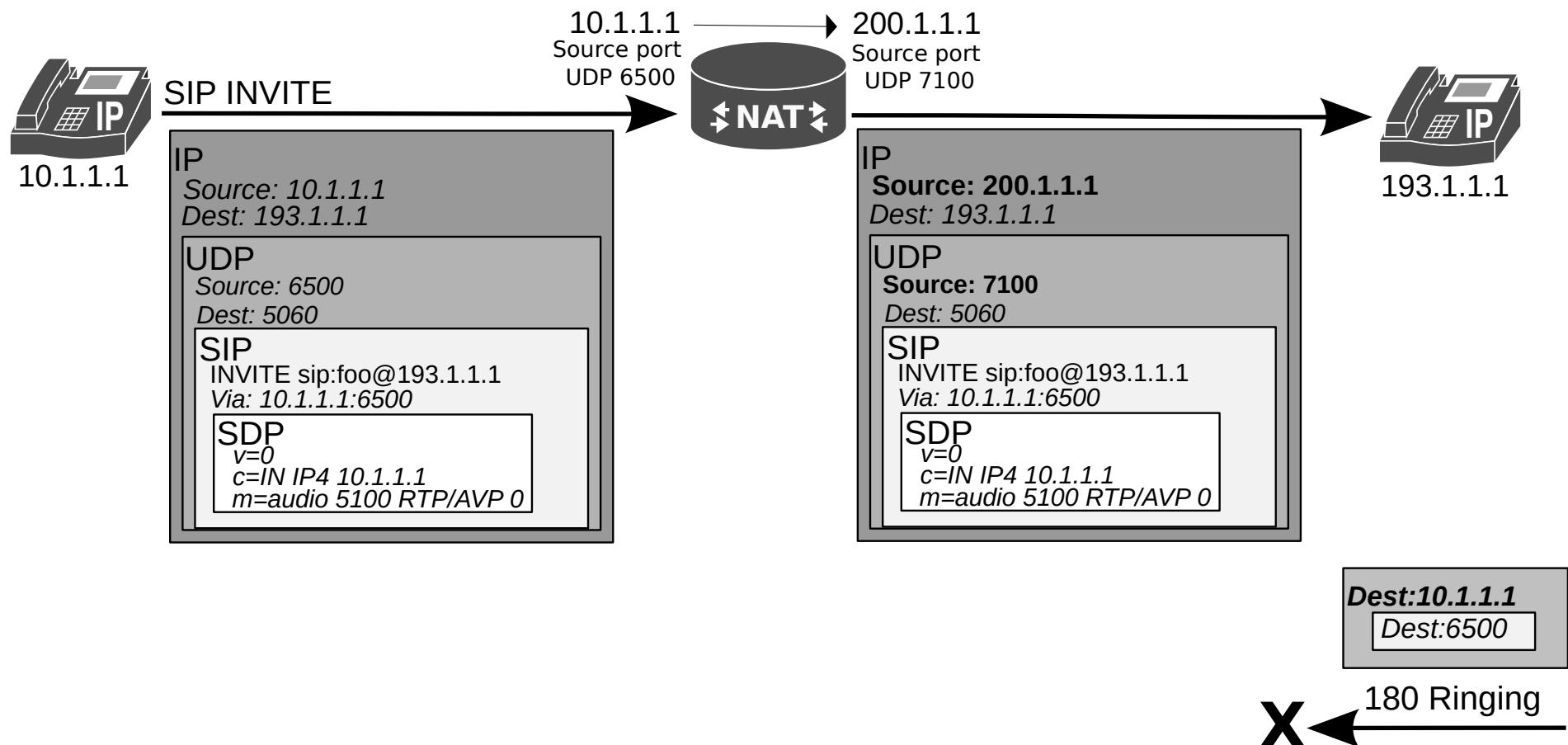
- RFC 3263 defines a set of DNS procedures to locate SIP Servers.
- SIP elements need to send requests/responses to a resource identified by a SIP URI.
 - The SIP URI may identify the desired target resource or a intermediate hop towards that resource.
 - Requires **Transport protocol, IP address and Port**.
 - If the URI specifies any of them, then it should be used.
 - Otherwise, must be retrieved from a DNS server.
 - Using **Service (SRV)** and **Name Authority Pointer (NAPTR)** DNS records.
- NAPTR records provide a mapping from a domain name to:
 - A SRV record (that contains the resource responsible server name),
 - And, the specific transport protocol.
- Example:
 - A client/server that wishes to resolve “sip:user@example.com”,
 - Performs a NAPTR query for domain “example.com”,
 - IN NAPTR 100 50 "s" "SIP+D2U" "" _sip._udp.example.com.
 - Has UDP as possible transport protocol, performs a SRV query for “_sip._udp.example.com”
 - IN SRV 0 1 5060 server1.example.com
 - IN SRV 0 2 5060 server2.example.com
 - Has two possible servers, performs A and AAAA queries for the chosen server.



SIP Proxy Forwarding



SIP and NAPT



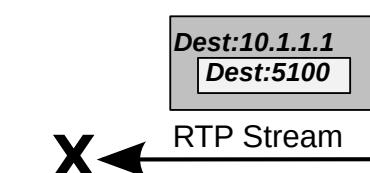
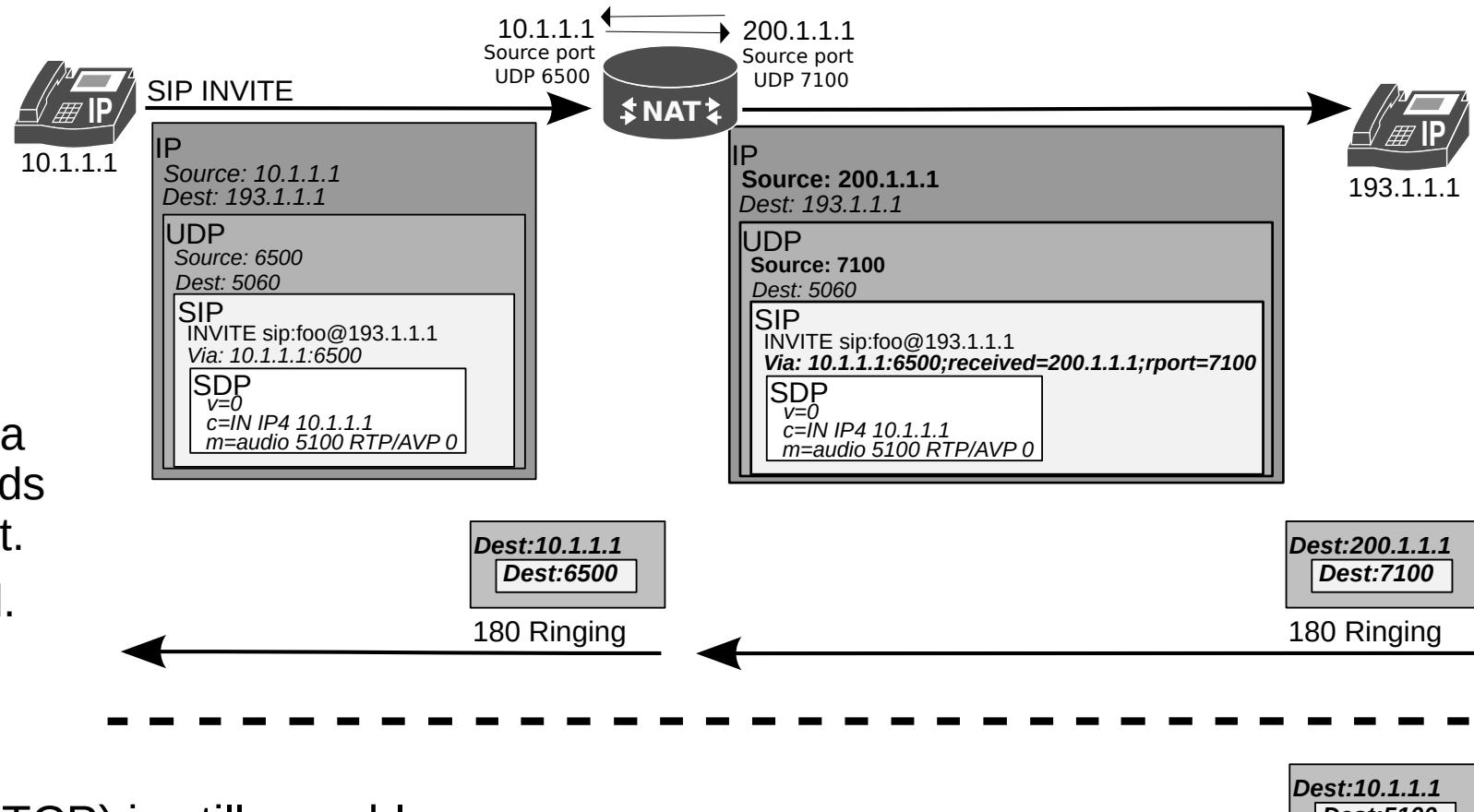
SIP NAPT Traversal

- Symmetric Response Routing (RFC 3581).

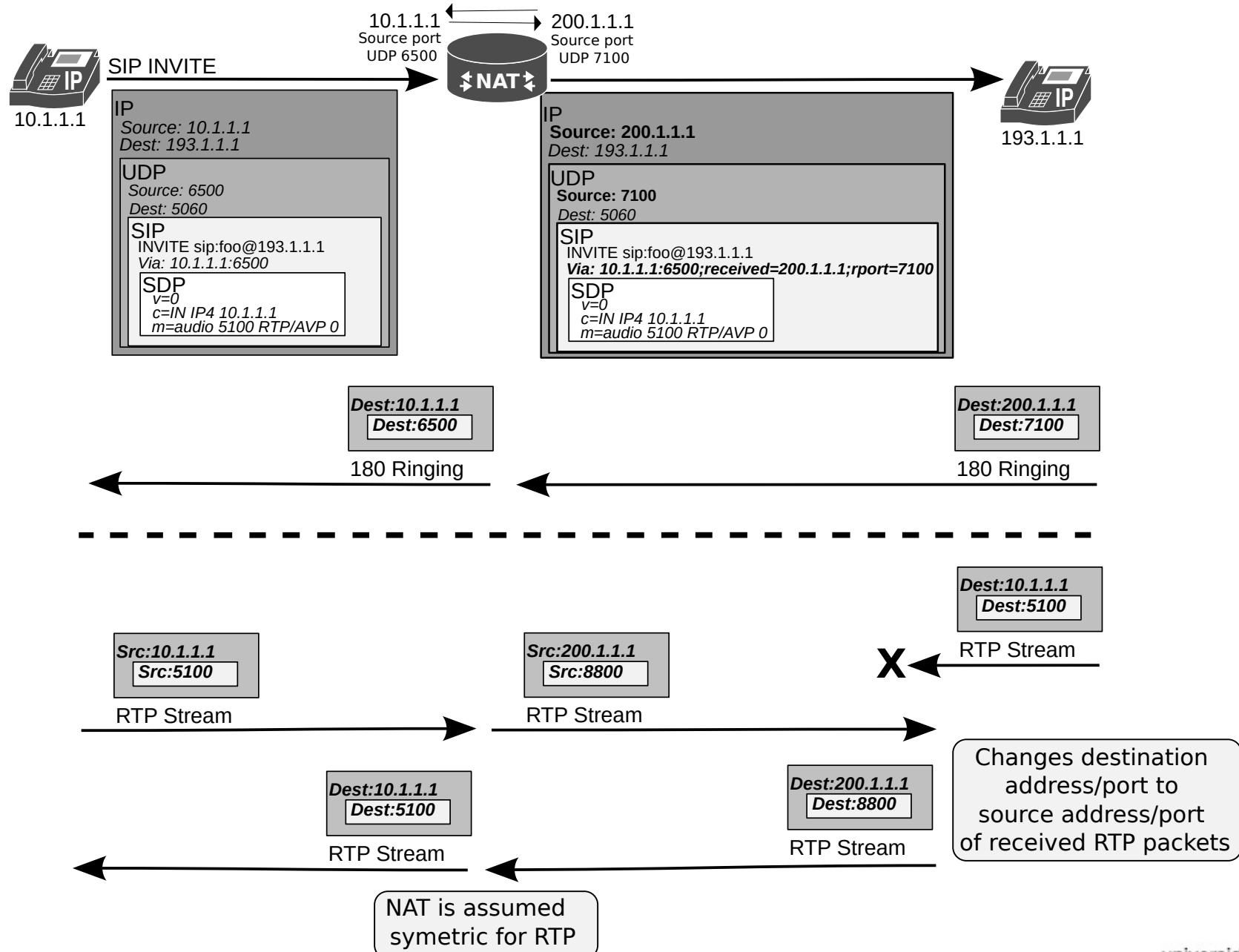
- SIP payload is also “translated”, by adding a **received** and **rport** fields with public address/port.
- SDP remains unchanged.

- Media traversal (RTP/RTCP) is still a problem.

- SDP contents mismatch with public address/port.
- Possible solutions
 - Let clients (on private network) find out their public address/port and rewrite SDP payload.
 - Manual configuration (when NAT uses static translations).
 - Automatic discovery (when NAT is dynamic) using STUN protocol.
 - Symmetric (RTP/RTCP) NAT (RFC 4961).
 - NAT SIP Application Layer Gateway (ALG).

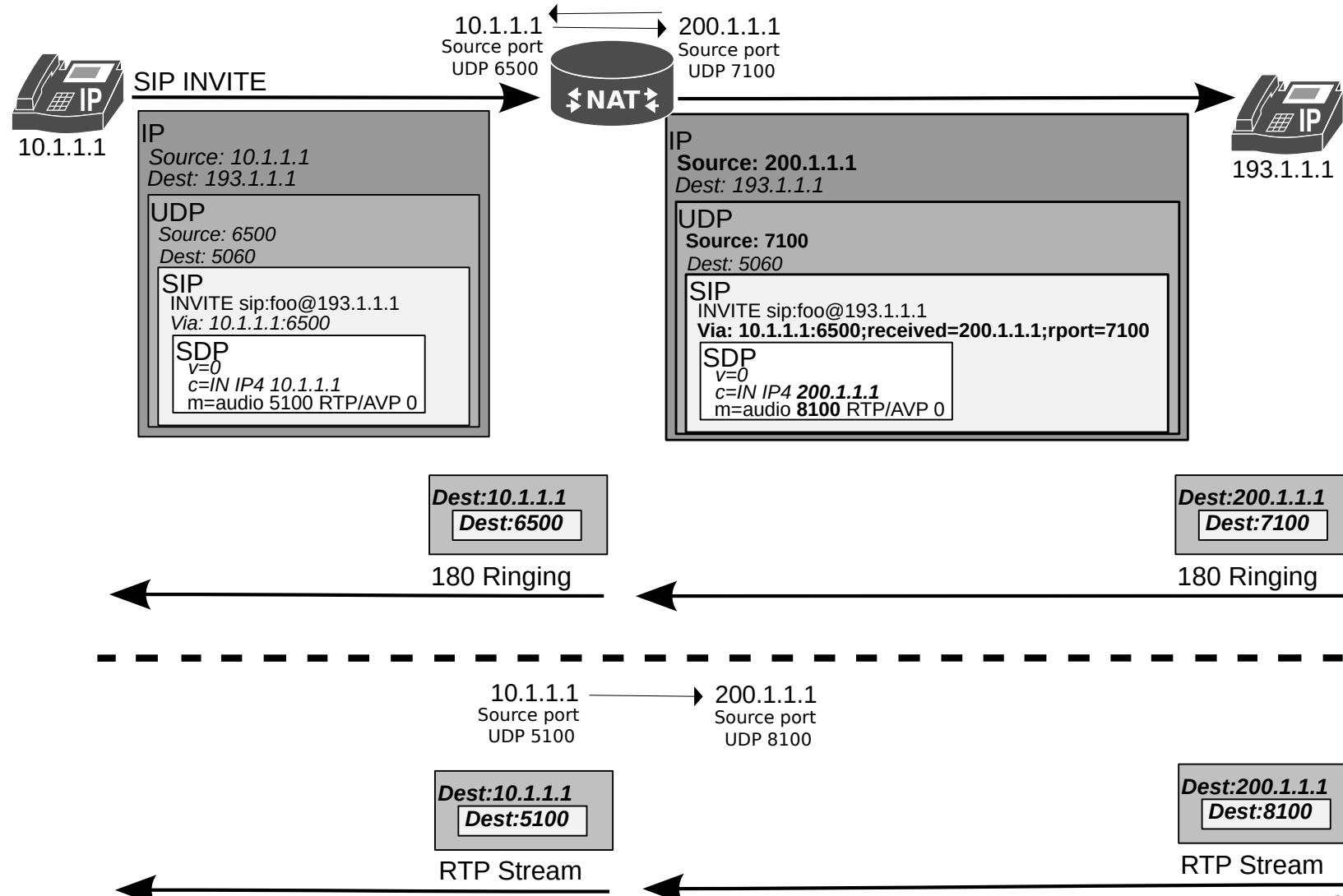


Symmetric (RTP/RTCP) NAT



NAT SIP Application Layer Gateway (ALG)

- Required to translate SDP payloads.
- Heavy on NAT gateway.



Network Physical Layer

Fundamentos de Redes

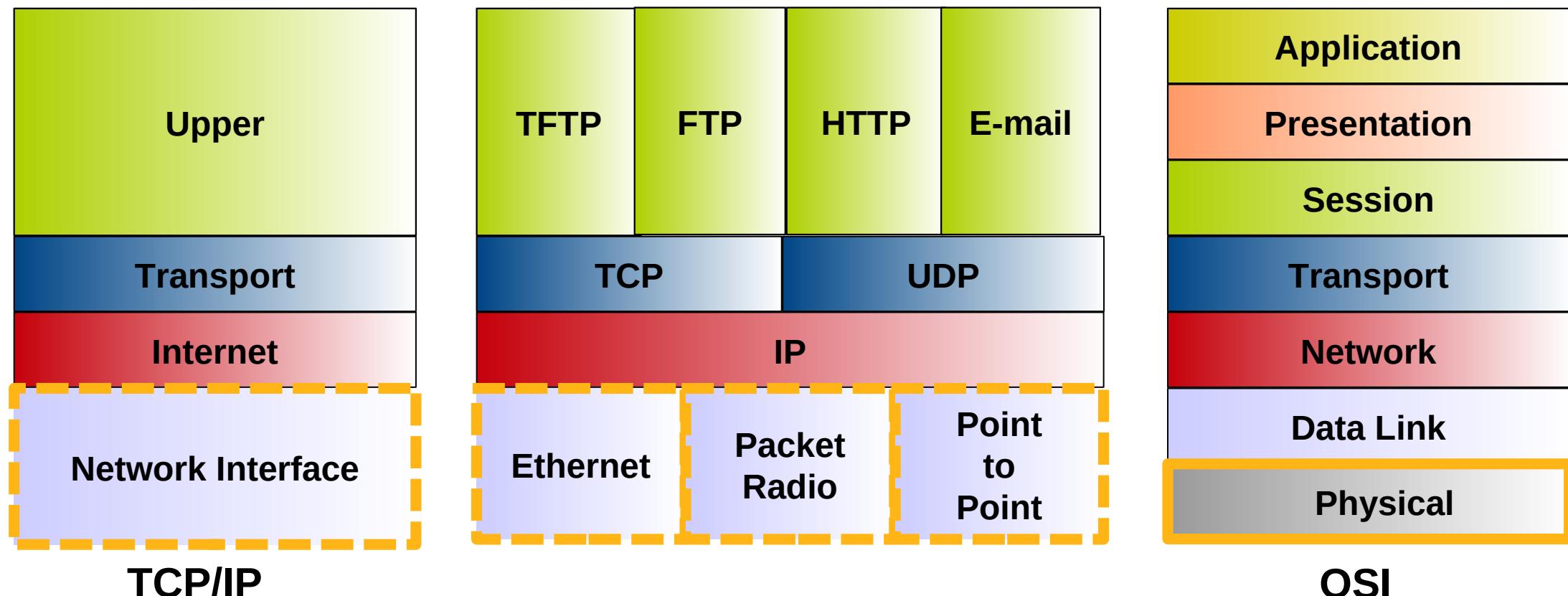
**Mestrado Integrado em
Engenharia de Computadores e Telemática
DETI-UA**



universidade de aveiro

deti.ua.pt

TCP/IP Reference Model



Shared Medium Access



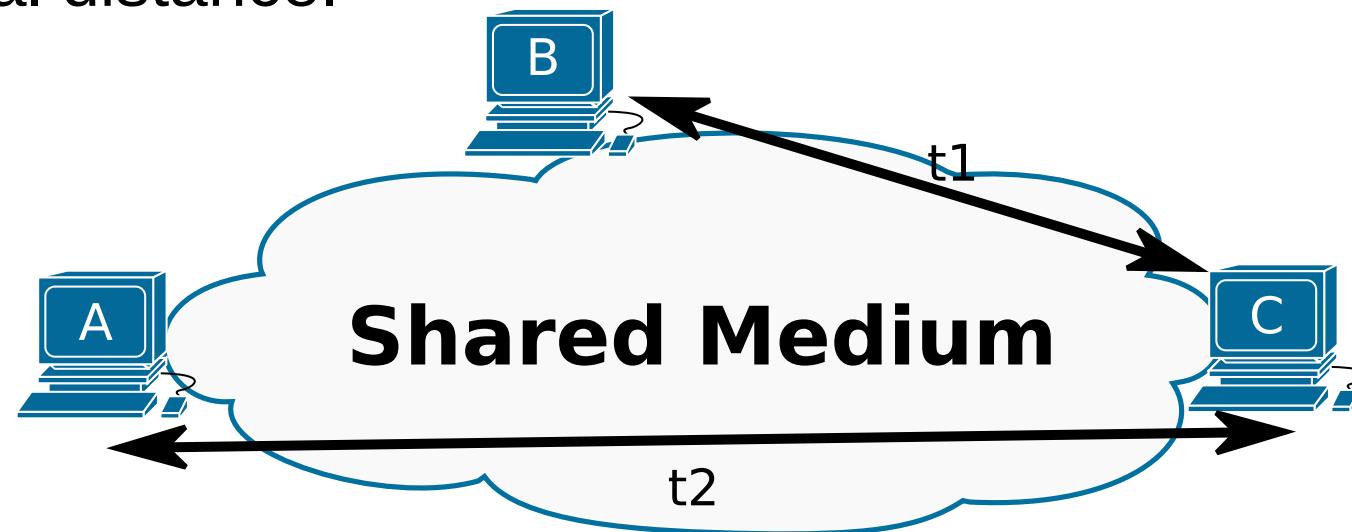
ALOHA

- The first version of the protocol (called "Pure ALOHA"):
 - If a station has data to send, sends it.
 - If, while transmitting data, the station receives any data from another station, there has been a message collision.
 - After a collision, all transmitting stations will need to try resend data later.
- A more efficient version (called “Slotted ALOHA”):
 - Introduced discrete timeslots.
 - A station can start a data transmission only at the beginning of a timeslot, and thus collisions are reduced.
 - Increased the maximum throughput.
 - Still used in very-low-data-rate satellite communications networks.
- Both are very inefficient.



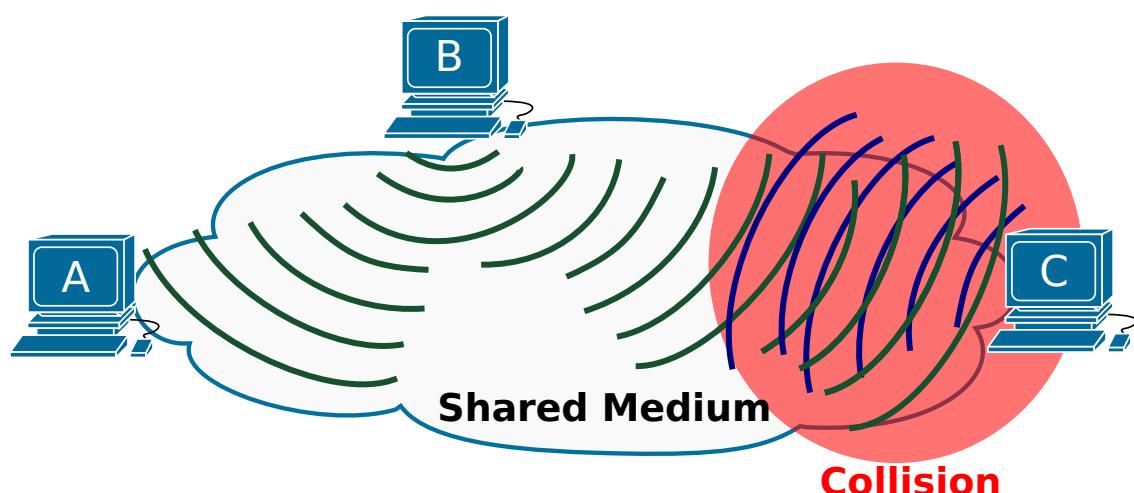
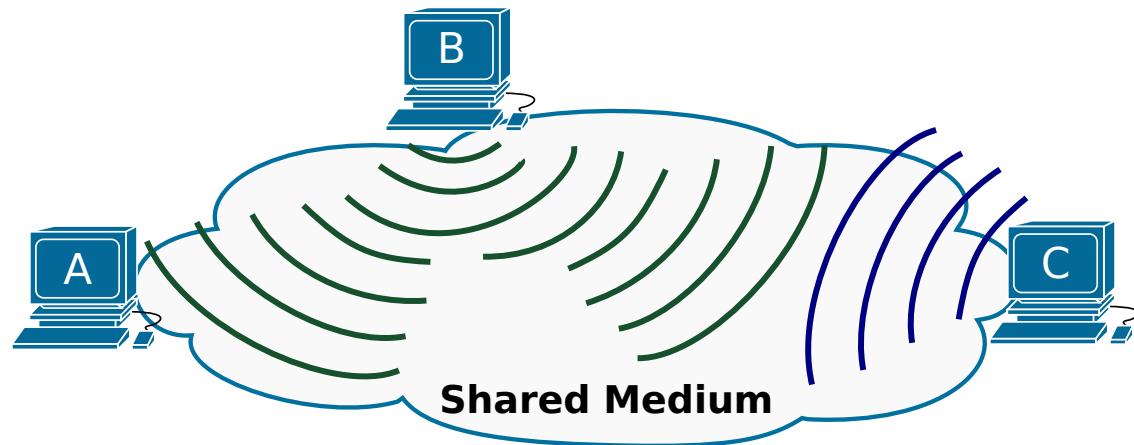
Carrier Sense Multiple Access (CSMA)

- Stations transmit and receive on the same channel.
- The stations listen to the medium before transmitting (carrier sense - CS).
- Only transmit if the medium is detected free.
- The number of collisions is minimized.
- Collisions may occur due to transmission times over some physical distance.



CSMA with Collision Detection (CSMA/CD)

- When stations detect a collision:
 - Stop transmitting,
 - May send a jam signal to reinforce collision detection,
 - Wait a random time before trying to resend message.
- To ensure that all stations detect a collision, all messages have a minimum size.
 - The time that takes all bots from a message to be transmitted must be larger than the time it takes (the first bit) to reach the farthest station on the shared medium, and return (round-trip time).



Ethernet vs. WiFi Medium Access

• Ethernet

- Uses CSMA/CD,
- In modern Ethernet networks (with no hubs) there is no collisions.
 - Switches avoid collisions.
 - Medium is not really shared.

• WiFi

- Used CSMA/CD, however:
 - Medium is shared,
 - Signal power reduces with the square distance,
 - Sender can apply CS and CD, but collisions occur in the receiver!
 - Sender may not listen the collision (CD does not work),
 - CS may not work either with hidden nodes.



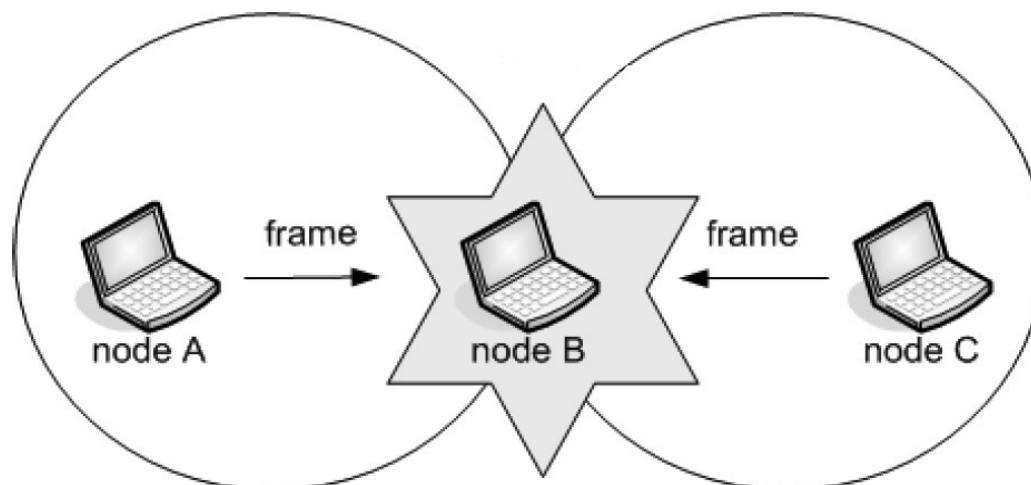
Hidden Nodes

- Hidden terminals

- A and C do not hear each other.
- Collision in B, if A and C send at the same time.
- Neither A or C understand that a collision occurred.

Solution

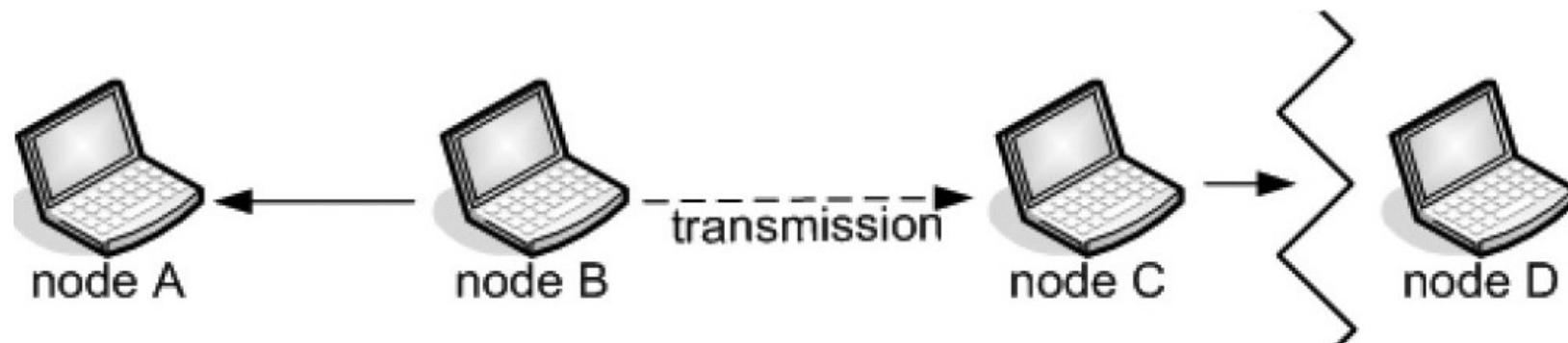
- Detect collisions in the receiver.
- “Virtual carrier sensing”: sender asks the receiver if he is receiving traffic; in the case of absence of answer, he assumes that the channel is busy.



Exposed Nodes/Terminals

- Exposed terminals

- B transmits to A;
- Node C wants to transmit to node D but mistakenly thinks that this will interfere with B's transmission to A, so C refrains from transmitting.
 - D is not in the range of B and A is not in the range of C, so traffic could have been transmitted.
- B and C are exposed terminals.
- The "exposed node" problem leads to loss of efficiency.



MACA: Multiple Access with Collision Avoidance

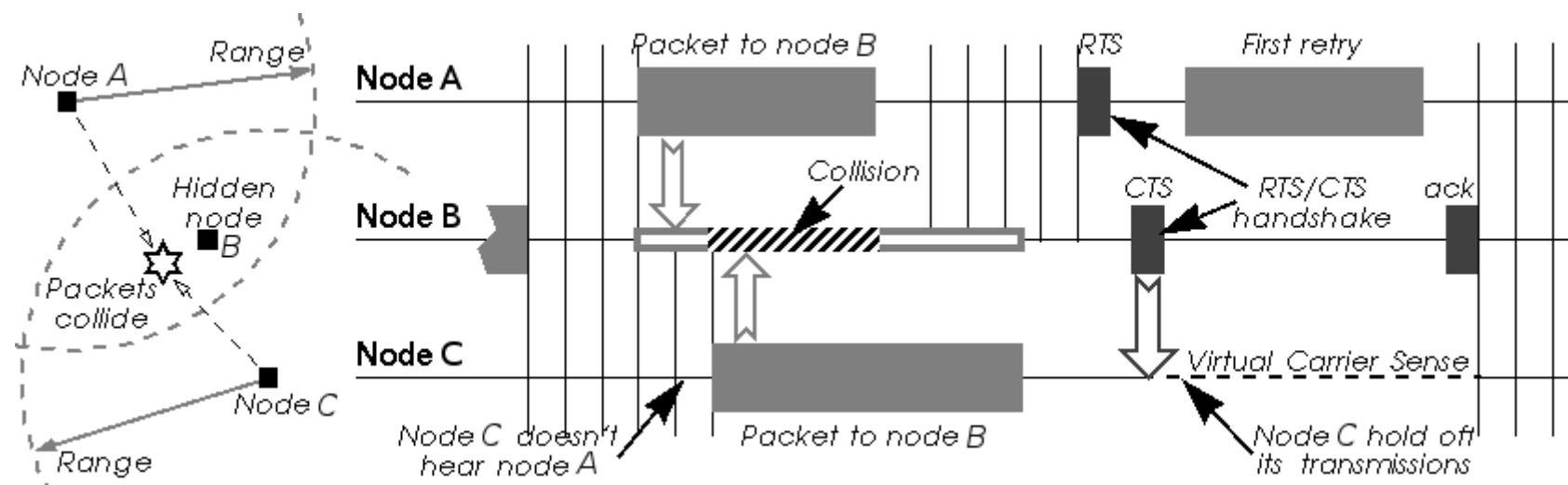
- MACA: avoids collisions using signalling packets
 - RTS (request to send)
 - A small packet is sent before transmitting
 - CTS (clear to send)
 - Receiver provides the right to transmit, when it is able to receive
- Signalling packets (RTS/CTS) contain
 - Sender address
 - Receiver address
 - Packet length (to be transmitted)
- Used in networks scenarios with a large amount of traffic/collisions.



MACA Advantages (1)

- MACA and hidden nodes

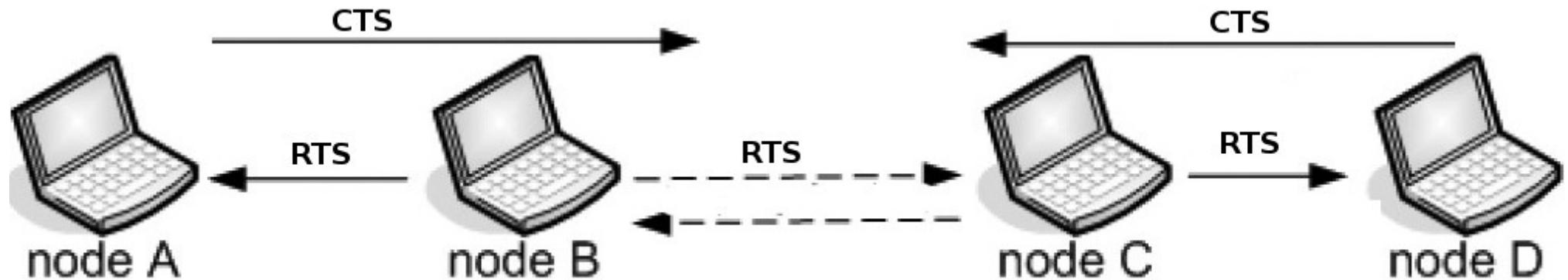
- A, C → B (Collision!)
- A RTS → B
- B CTS → A
- C hears CTS of B.
- C waits for the period announced in A transmission.



MACA Advantages (2)

- MACA and exposed nodes

- $B \rightarrow A, C \rightarrow D(?)$
- $B \text{ RTS} \rightarrow A$
- $A \text{ CTS} \rightarrow B$
- C ears RTS of B.
- C does not ear CTS of A.
- $C \text{ RTS} \rightarrow D$



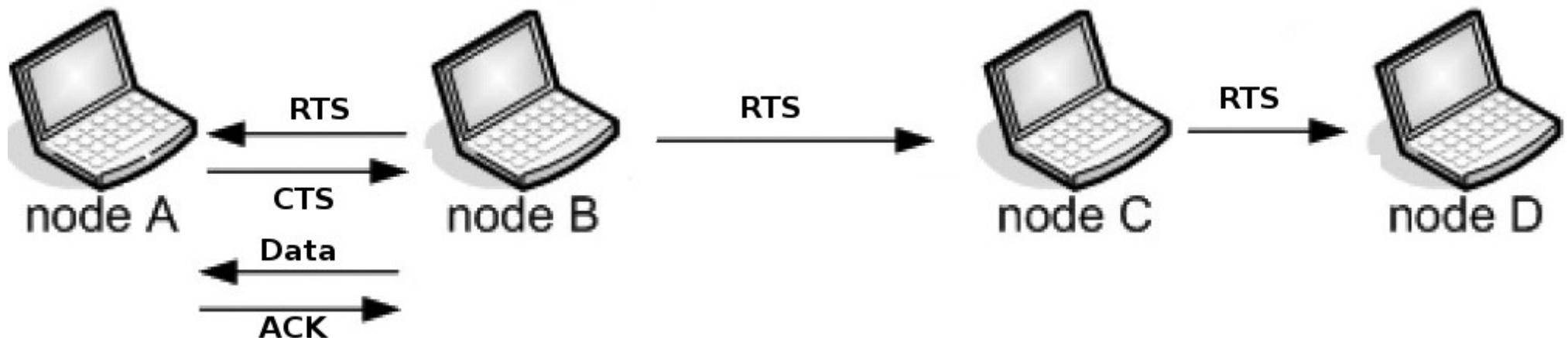
MAC Reliability

- Wireless connections are very prone to errors.

- Transport is not reliable!

- Solution: use **Acknowledgements**

- When A receives DATA from B, answers with ACK.
- If B does not receive ACK, B retransmits.
- C and D will not transmit until the ACK (to avoid collisions).
- Total expected duration (including ACK) is included in the RTS/CTS packets.



RST/CTS Frames

- IEEE 802.11 Request-to-send, Flags:c

Type/Subtype: Request-to-send (0x001b)

- Frame Control Field: 0xb400

.000 0111 0000 0100 = Duration: 1796 microseconds

← From Data Transmitter

Receiver address: Cisco_2b:d3:70 (f4:cf:e2:2b:d3:70)

Transmitter address: Microsof_0a:43:e3 (c0:33:5e:0a:43:e3)

Frame check sequence: 0xe058c51c [unverified]

[FCS Status: Unverified]

From Data Receiver →

- IEEE 802.11 Clear-to-send, Flags:c

Type/Subtype: Clear-to-send (0x001c)

- Frame Control Field: 0xc400

.000 0110 0010 1010 = Duration: 1578 microseconds

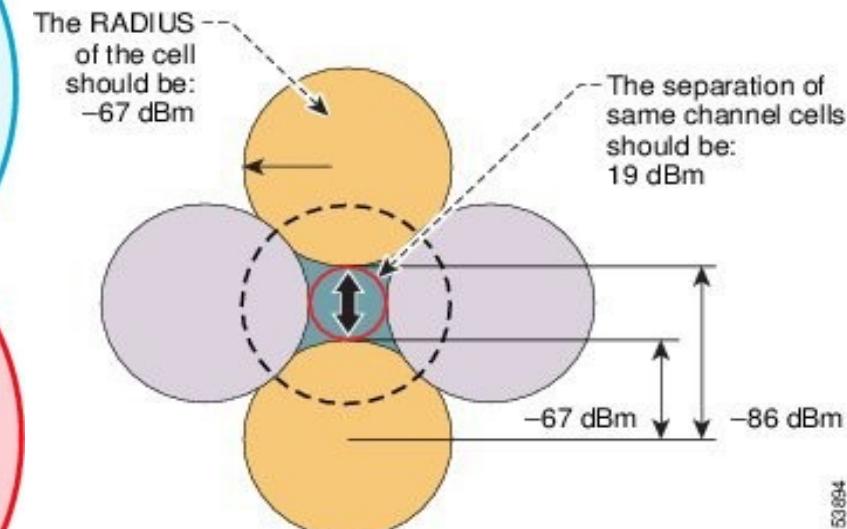
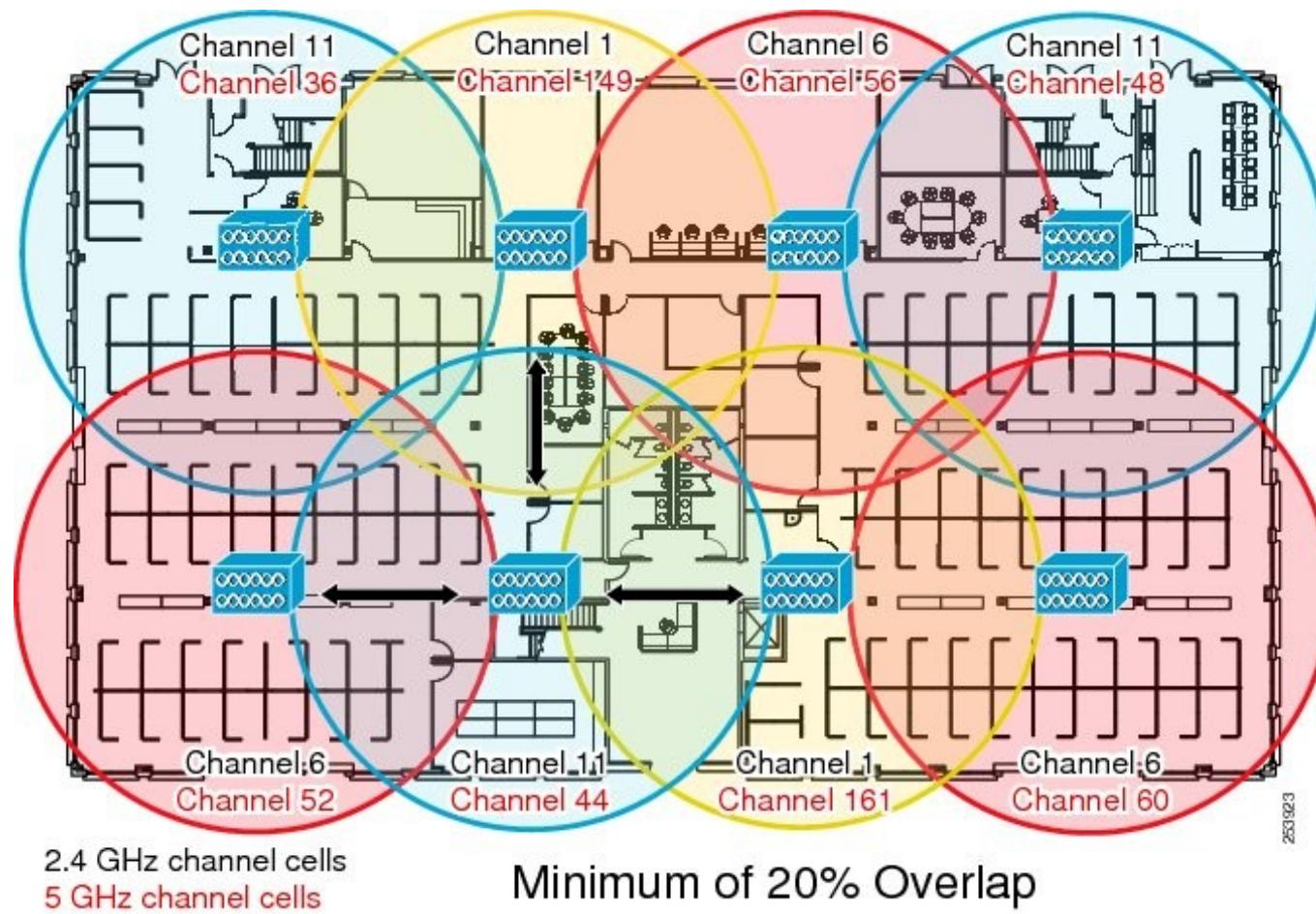
Receiver address: Microsof_0a:43:e3 (c0:33:5e:0a:43:e3)

Frame check sequence: 0xaac303a8 [unverified]

[FCS Status: Unverified]



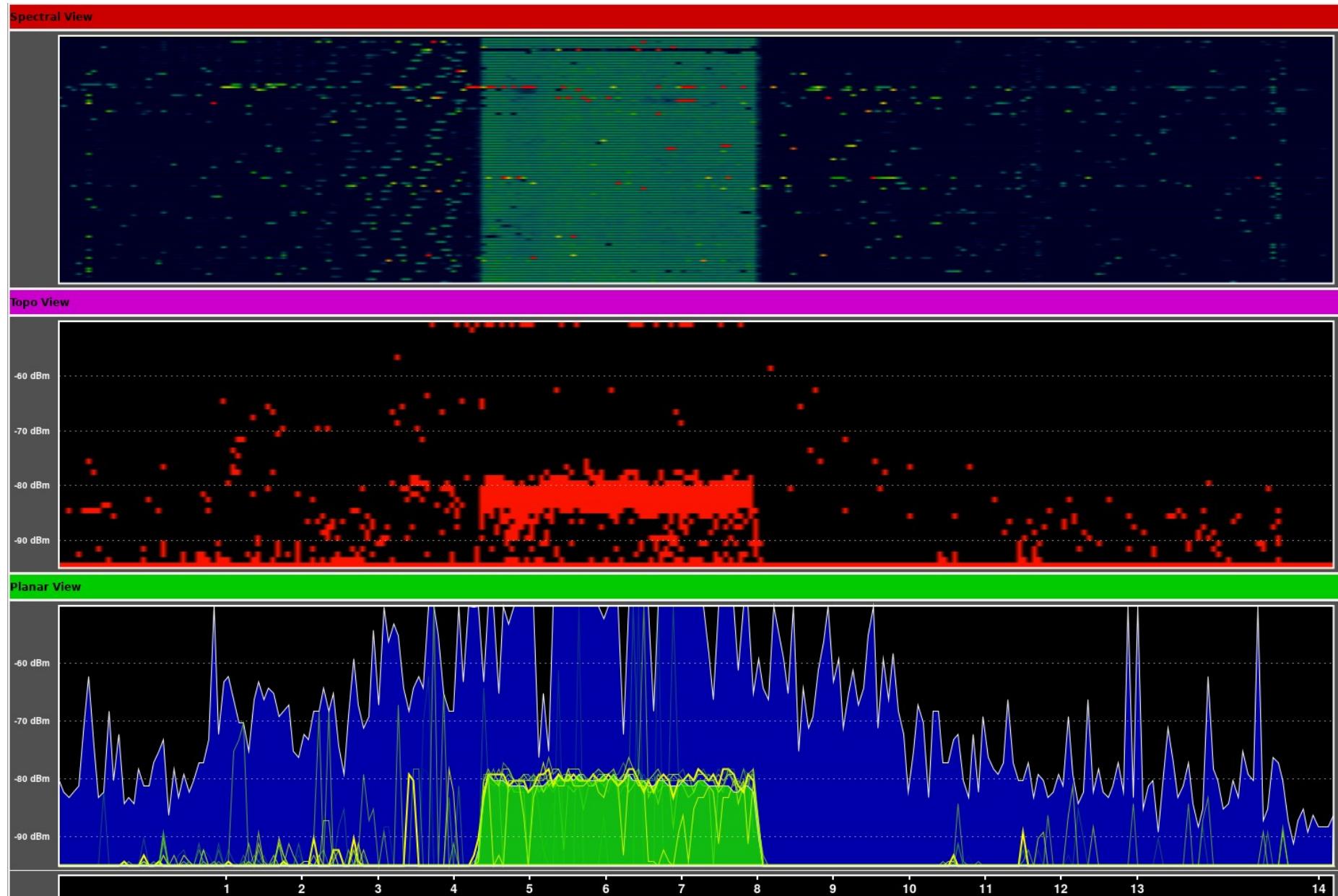
AP Placement and Channel Allocation



- 802.11n or 802.11ac 5GHz deployment does not have the overlap or collision domain issues of 2.4GHz.



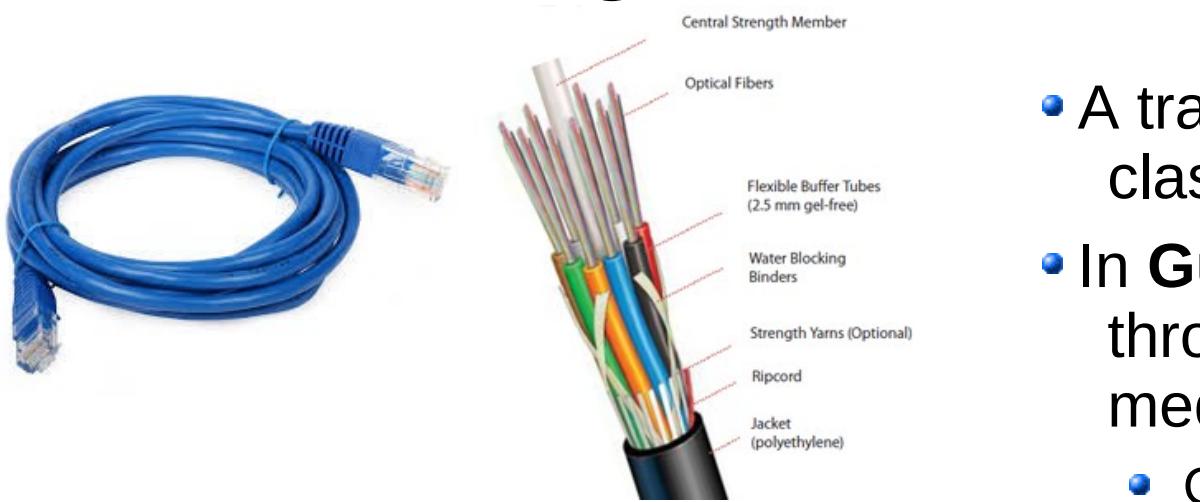
Usage of Spectrum Analysis



Transmission Systems and Technologies



Guided/Unguided Transmission Systems



Microwave link



Free Space Optics (FSO)



Directional LTE



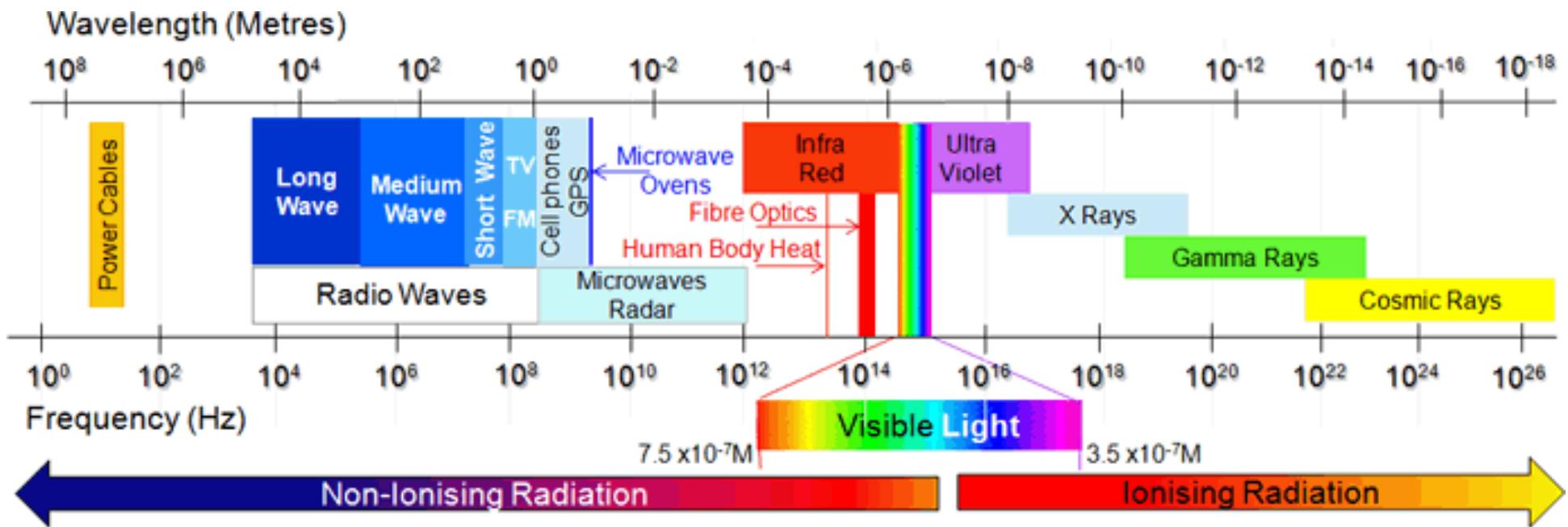
Omnidirectional LTE



802.11 Omnidirectional

- A transmission system can be classified as **Guided** or **Unguided**.
- In **Guided** systems, a signal travels through a bounding physical medium.
 - Copper cable, Optical fibre, ...
- In **Unguided** media, a signal travels through a boundless medium
 - Air, Water, Vacuum, ...
 - Can be directional or omni-directional.
 - In directional configuration, the source emits a focused beam in a particular direction.
 - The receiver should be aligned for receiving the signals.
 - In omni-directional configuration, the source emits equally in all directions.

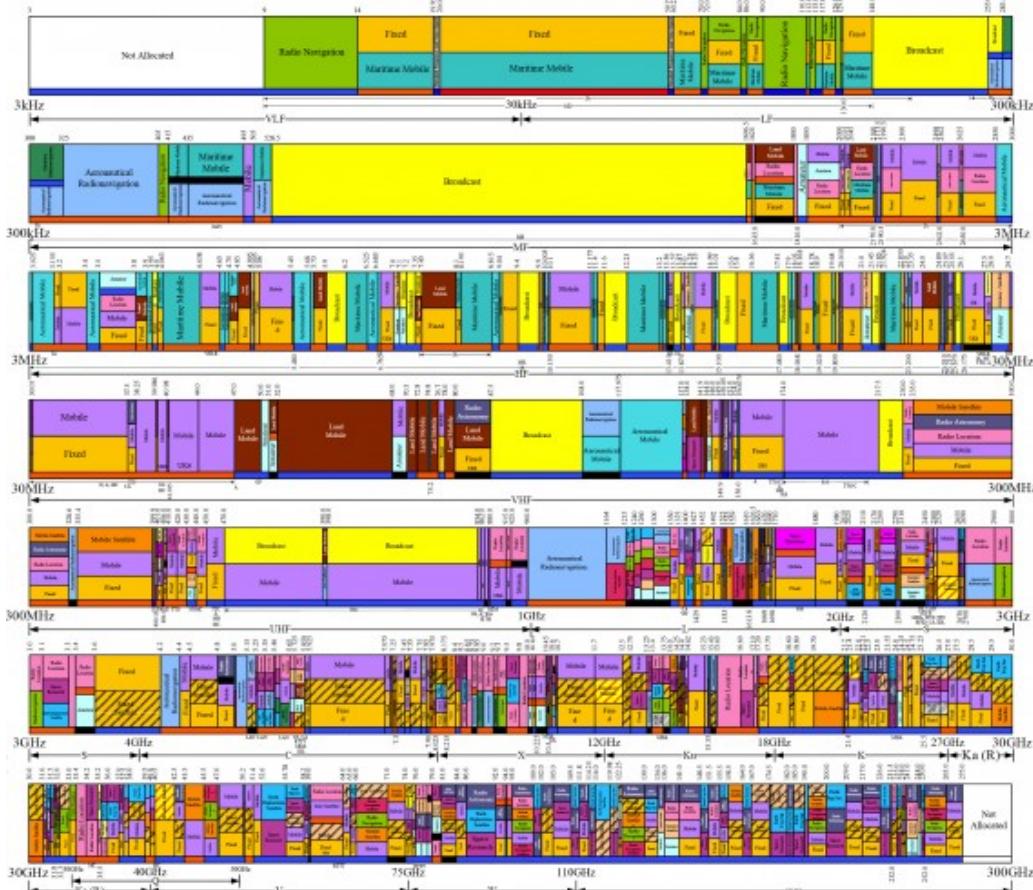
Electromagnetic Spectrum



- For radio signals the antenna transmits a sinusoidal signal (“carrier”) that radiates in air/space.



Radio/Microwave Spectrum (3KHz-300GHz)

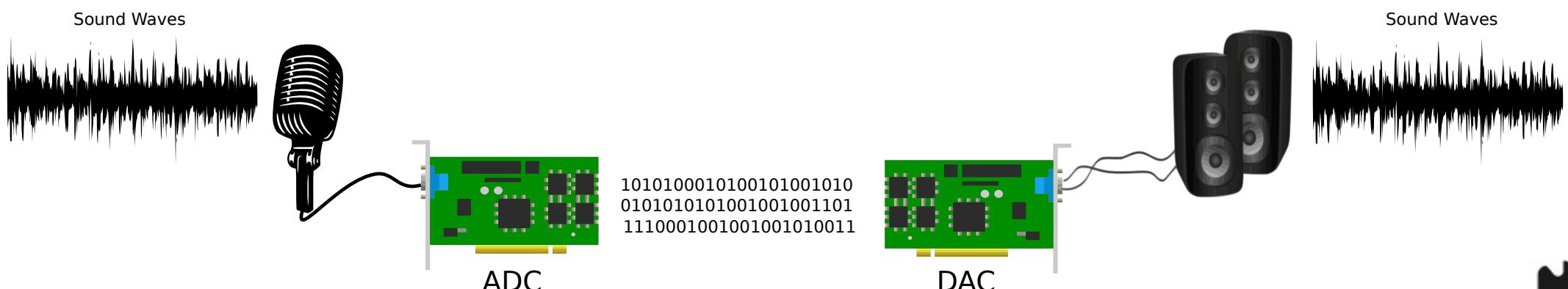


- Portugal (ANACOM)
 - <https://www.anacom.pt/render.jsp?categoryId=150422>
- UK (OFCOM)
 - <https://www.ofcom.org.uk/spectrum/information/uk-fat>
- USA (FCC)
 - <https://www.fcc.gov/engineering-technology/policy-and-rules-division/general/radio-spectrum-allocation>
- ...



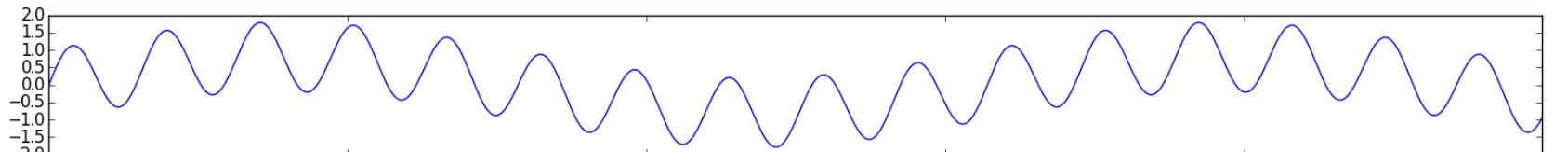
Analogue-Digital Conversion

- The digital transmission of analogue signals requires:
 - An ADC in the source, and
 - A DAC in the destination.
- ADC (Analogue to Digital Conversion)
 - Sampling
 - Quantization and Encoding
- DAC (Digital to Analogue Conversion)
 - Signal reconstruction

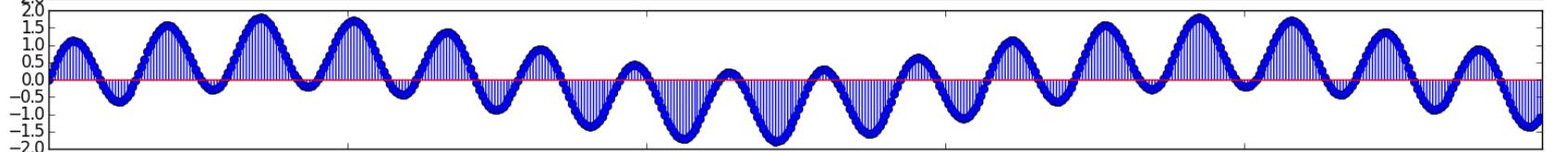


Sampling

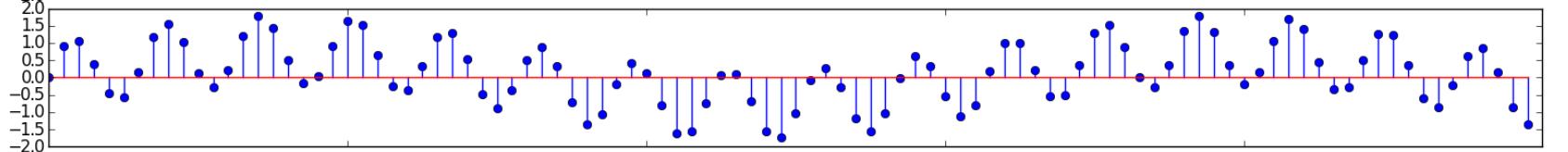
Analogue signal



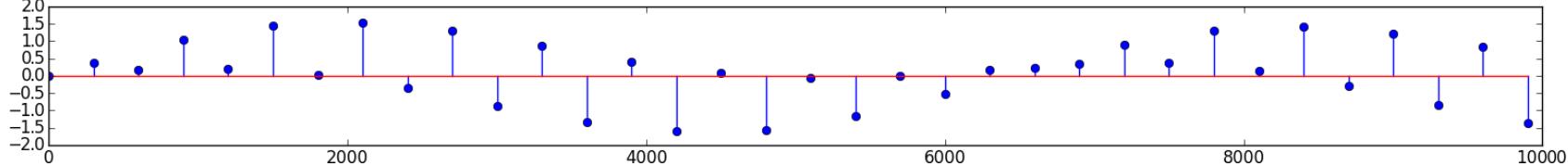
High sampling rate



Medium sampling rate



Low sampling rate

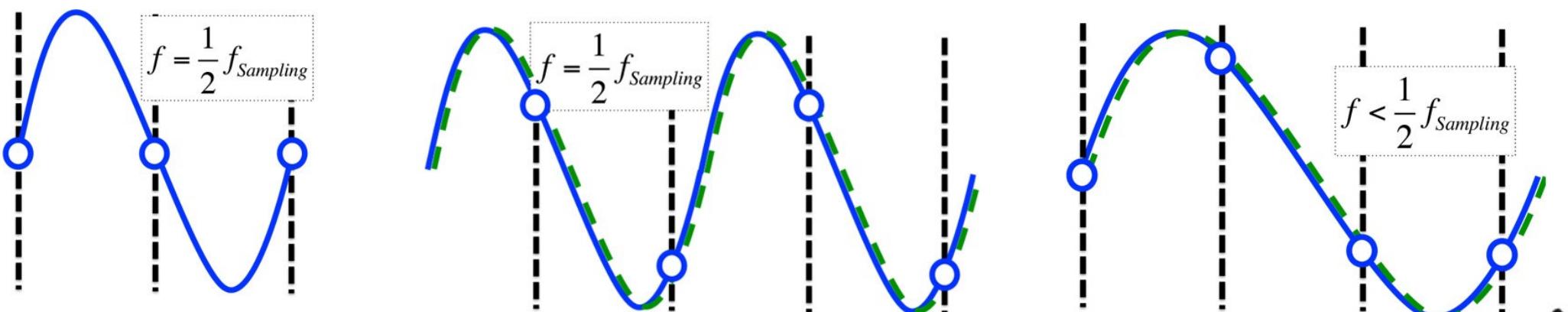


- The sampling process, measures and quantifies the analogue signal at equally space time intervals.
- The sampling process must be able to capture the main characteristics of the original analogue signal.
- The sampling rate determines the amount of information that its transferred to the digital signal.



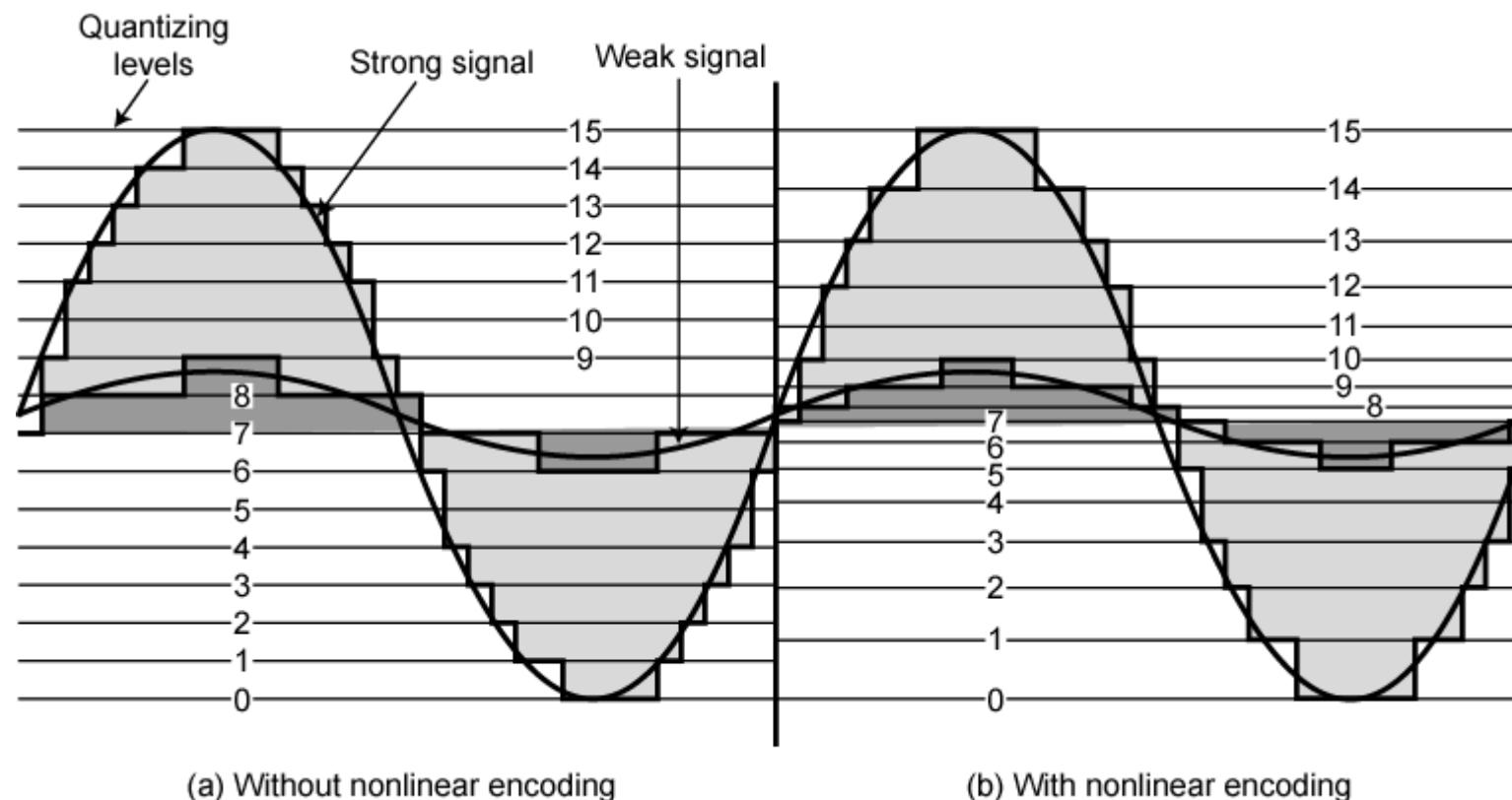
Sampling Theorem

- To reconstruct a signal from the samples, the sampling frequency must be high enough to capture the relevant signal information (frequency components).
 - Sampling frequency is the number of samples per second (f_s).
- For a signal where the highest (relevant) frequency is f_m , the sampling frequency (f_s) must be higher than two times f_m
 - $f_s > 2 * f_m \Leftrightarrow f_m < f_s / 2$
 - $f_s / 2$ is called the **Nyquist frequency**.
 - $2 * f_m$ is called the **Nyquist rate**.



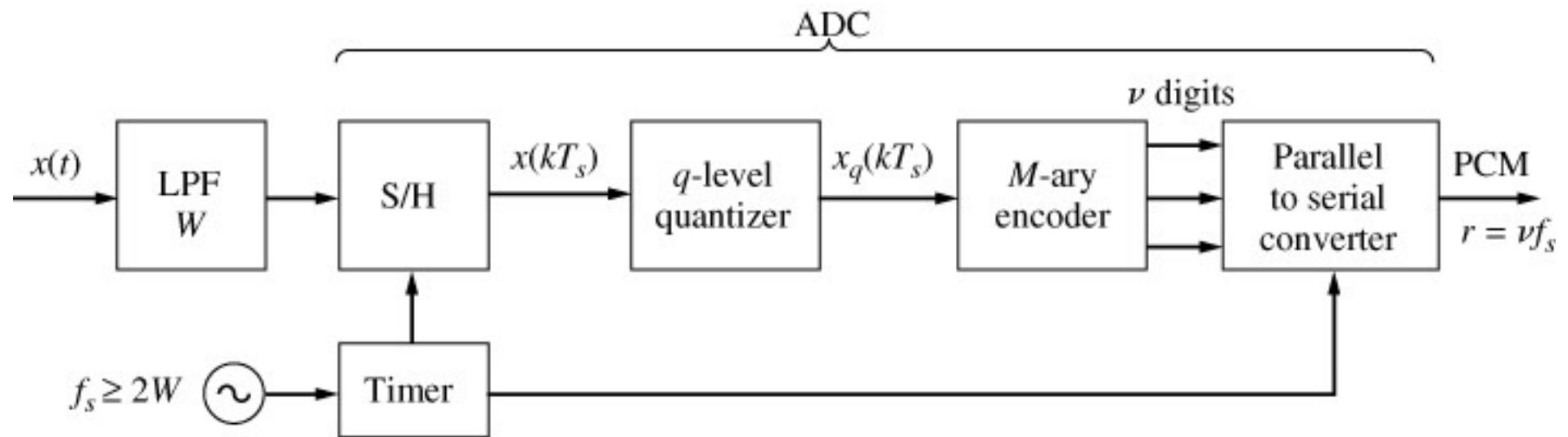
Signal Quantization and Encoding

- Each sampled value must be “rounded” to the nearest member of a set of discrete values.
- The resulting value is then encoded into a binary format.



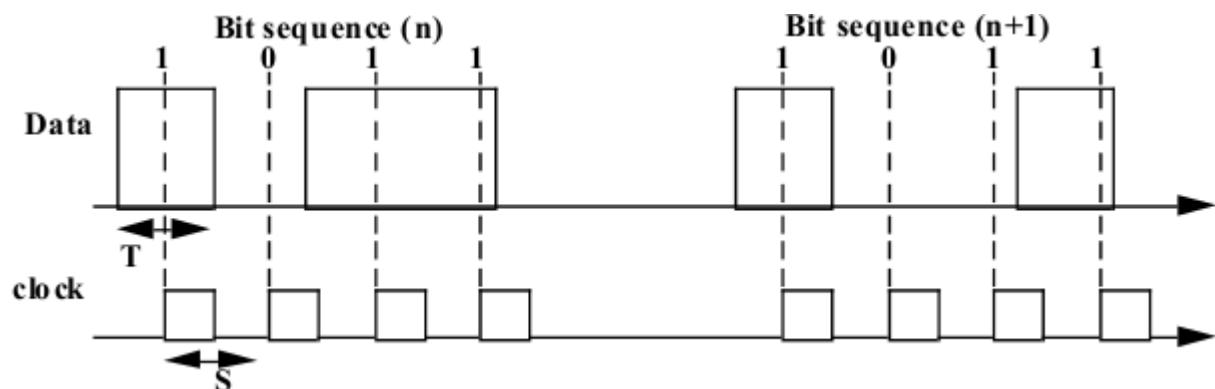
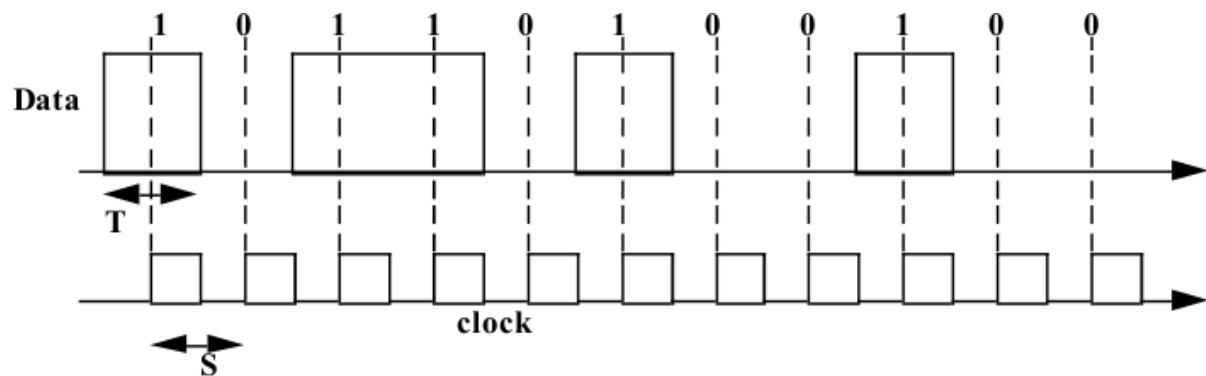
Pulse Code Modulation (PCM)

- All mechanisms of an ADC can be implemented using a PCM encoder.



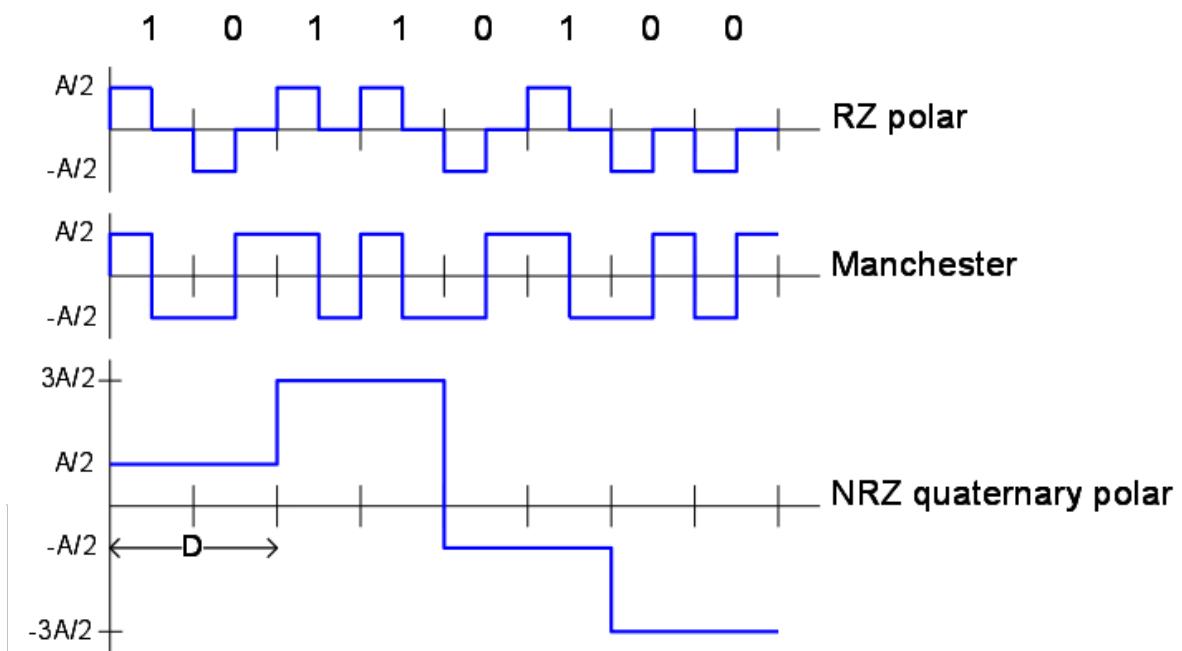
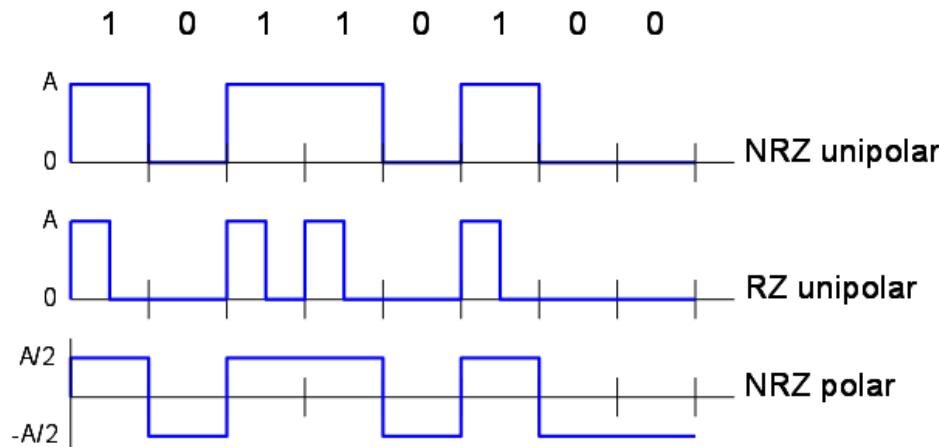
Digital Transmission

- Can be synchronous or asynchronous.
 - Synchronous Transmission data is transferred in the form of frames.
 - Asynchronous Transmission data is transmitted 1 bit or byte at a time.
- Synchronous Transmission requires a clock signal between the sender and receiver.
- Asynchronous Transmission sender and receiver does not require a clock signal, but data blocks must have a parity bit attached to it which indicates the start (start bit) of the new byte.
 - And, an optional stop bit.



Line Coding (1)

- Line Coding converts a binary sequence into a digital signal
- Sender then uses the digital signal to modulate transmitting signal in a way that the receiver can recognize.
- Line Coding can be done bit a bit, or in block of several bits (symbol).
- There are several (bit a bit) Line Codes:



Line Coding (2)

- mB/nB Encoding

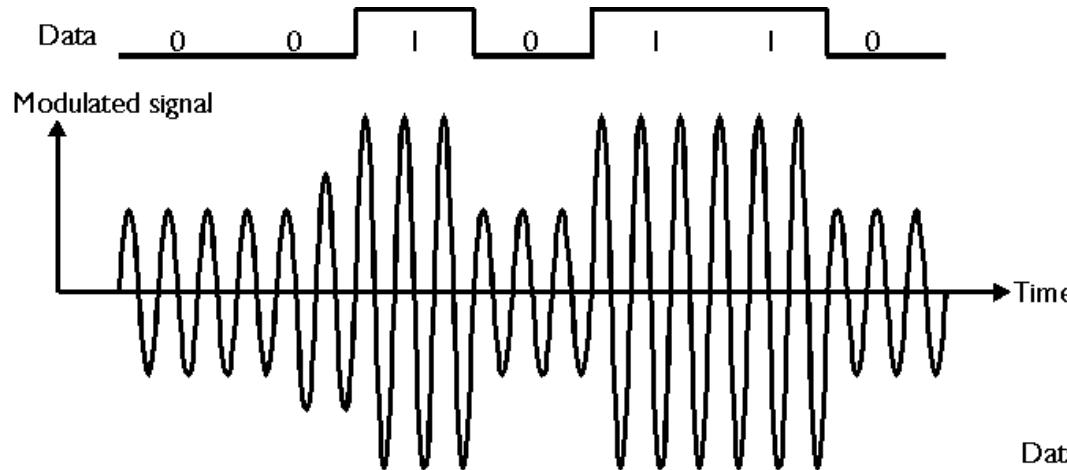
- Symbols of m bits are coded as line symbols of n bits.
- Each valid line symbol has at least two 1s.

| 4B/5B Code | | | |
|------------|--------|-------|--------|
| Bits | Symbol | Bits | Symbol |
| 0000 | 11110 | IDLE | 11111 |
| 0001 | 01001 | J | 11000 |
| 0010 | 10100 | K | 10001 |
| 0011 | 10101 | T | 01101 |
| 0100 | 01010 | R | 00111 |
| 0101 | 01011 | S | 11001 |
| 0110 | 01110 | QUIET | 00000 |
| 0111 | 01111 | HALT | 00100 |
| 1000 | 10010 | | |
| 1001 | 10011 | | |
| 1010 | 10110 | | |
| 1011 | 10111 | | |
| 1100 | 11010 | | |
| 1101 | 11011 | | |
| 1110 | 11100 | | |
| 1111 | 11101 | | |

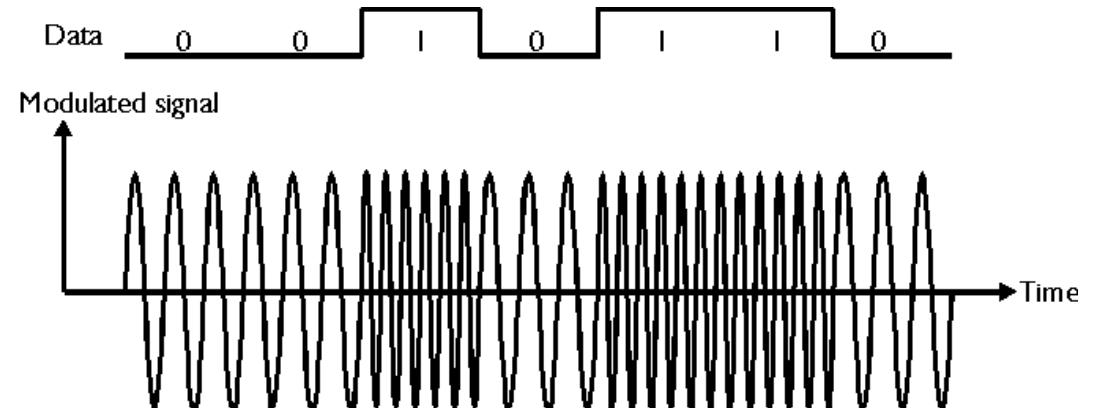


Modulation (1)

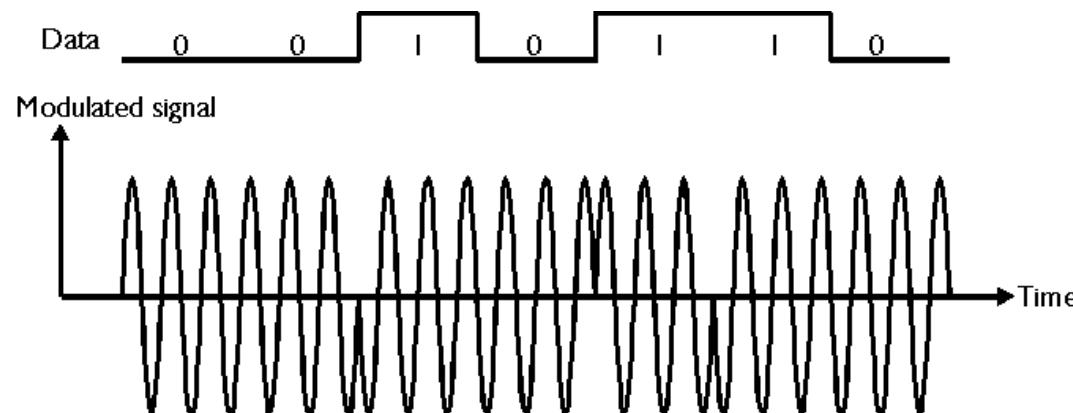
- Amplitude



- Frequency



- Phase

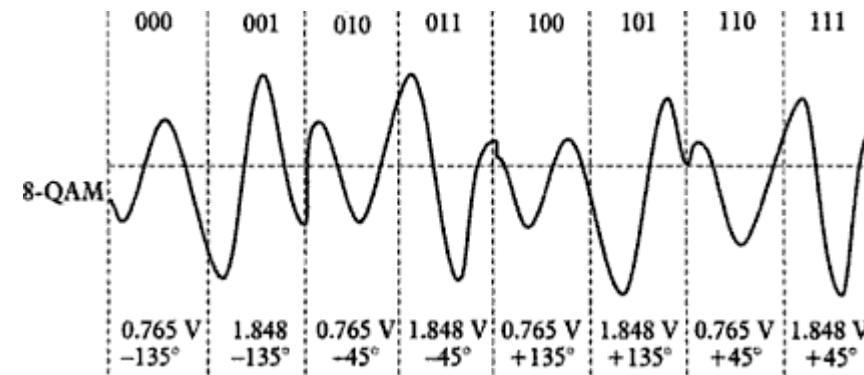
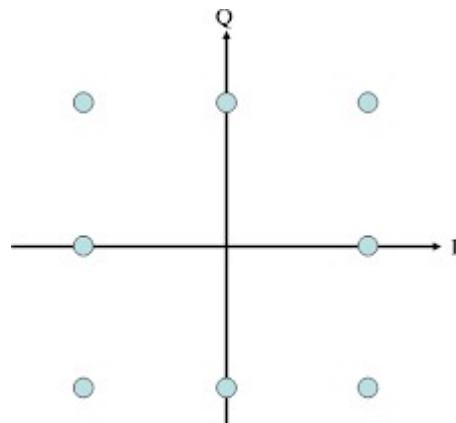


Modulation (2)

- Quadrature Amplitude Modulation (QAM)

- Uses 2-Dimensional signalling
 - Quadrature \leftarrow Sine wave + Cosine wave
 - $s(t) = I(t)\cos(2\pi f_0 t) - Q(t)\sin(2\pi f_0 t)$

- 8-QAM



- 16-QAM

