

# Local Area Networks (LAN)

-

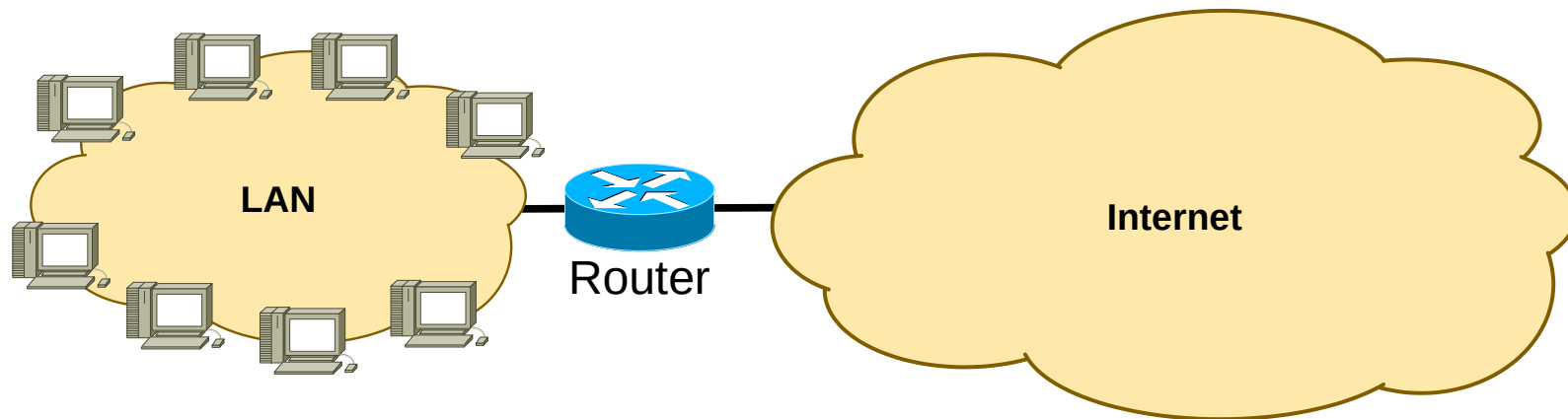
Introduction to Switching, IPv4 and Routing

**Fundamentos de Redes**

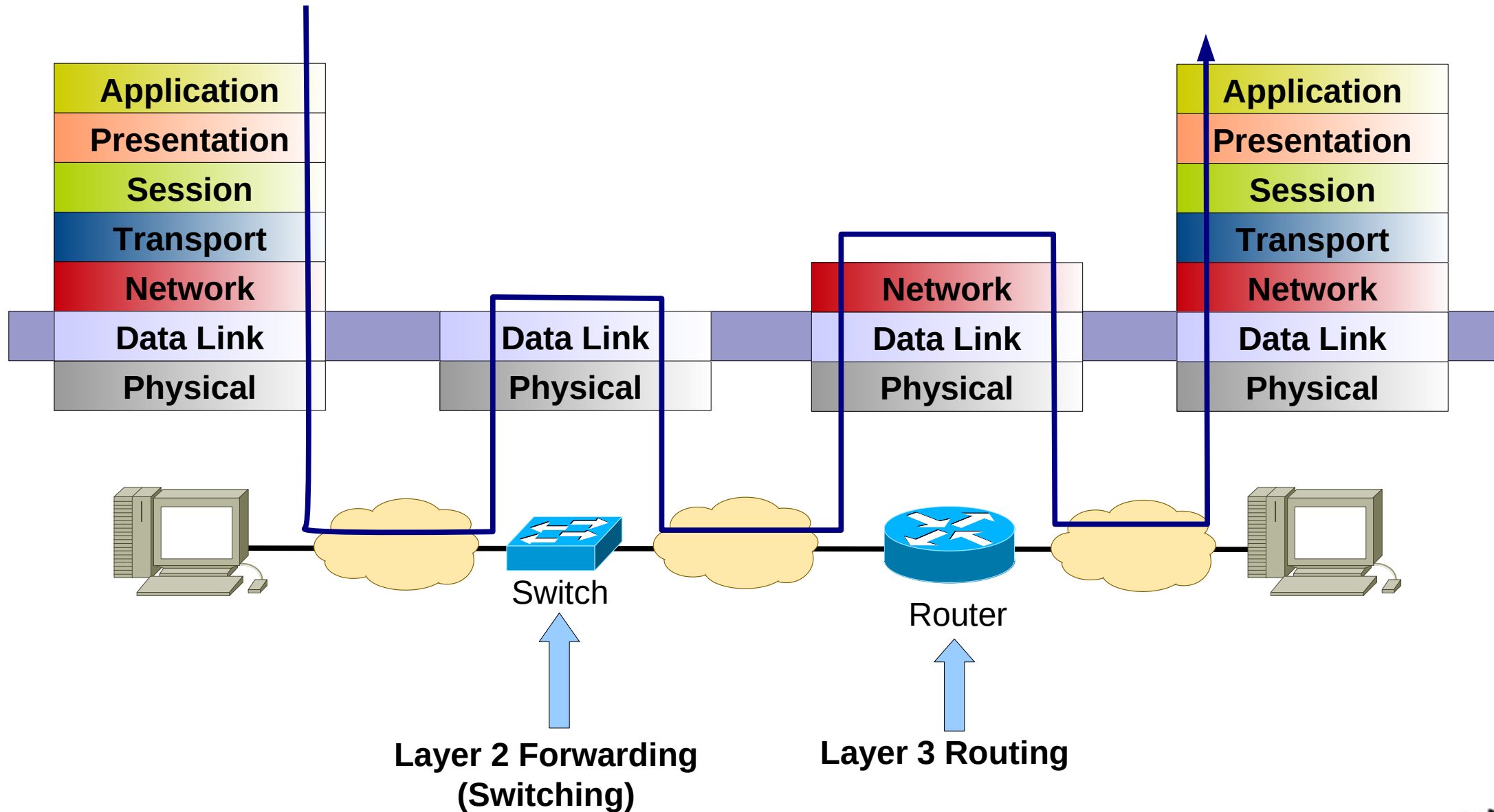
**Mestrado Integrado em  
Engenharia de Computadores e Telemática  
DETI-UA**

# 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, ...



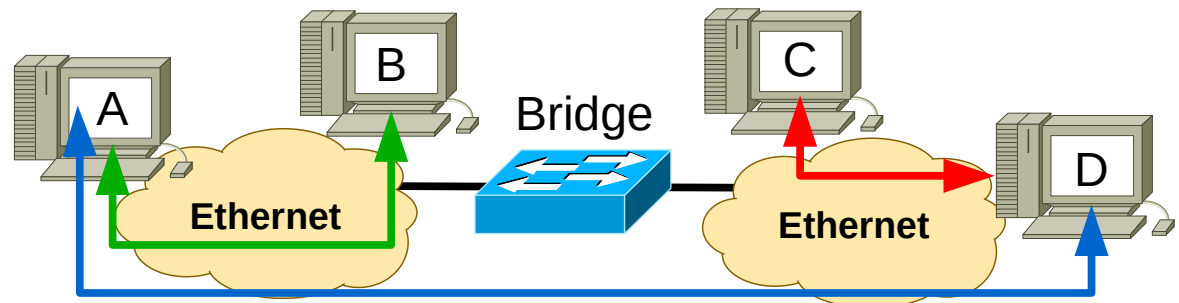
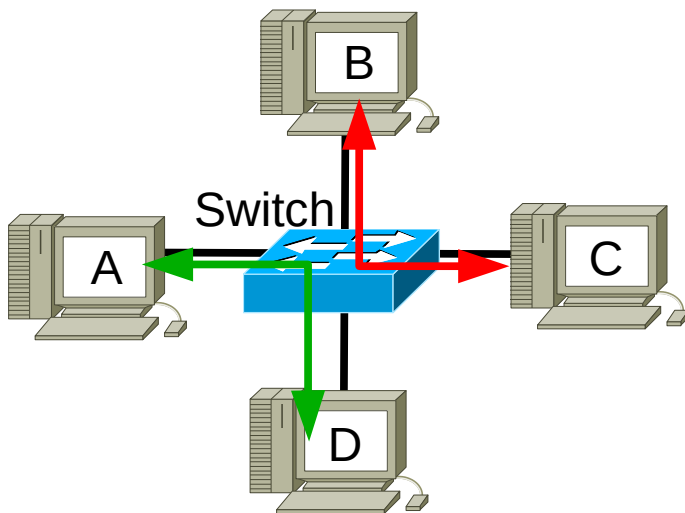
# Local Area Network (LAN)



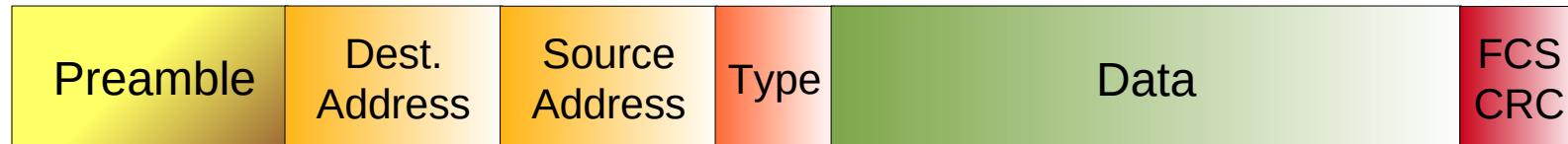
# Switching

- 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



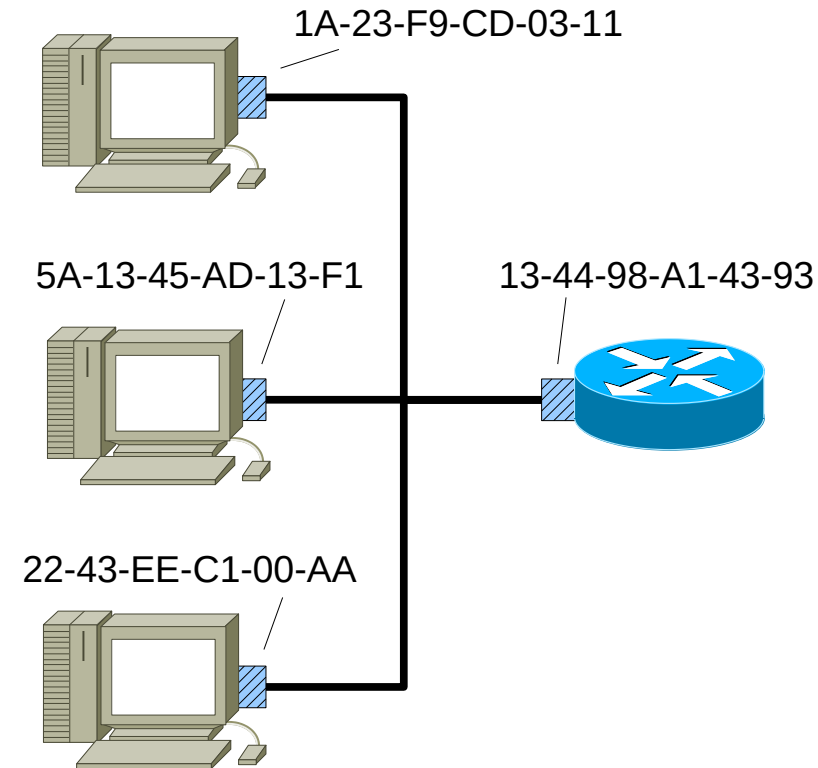
- The sender's network card encapsulates an IP datagrama (or any other network protocol) in an Ethernet frame.
- Preamble:
  - 7 bytes with pattern 10101010 followed by one byte with pattern 10101011.
  - Used to synchronize the sending and receiving clocks.
- Destination and Source addresses: 6 bytes 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





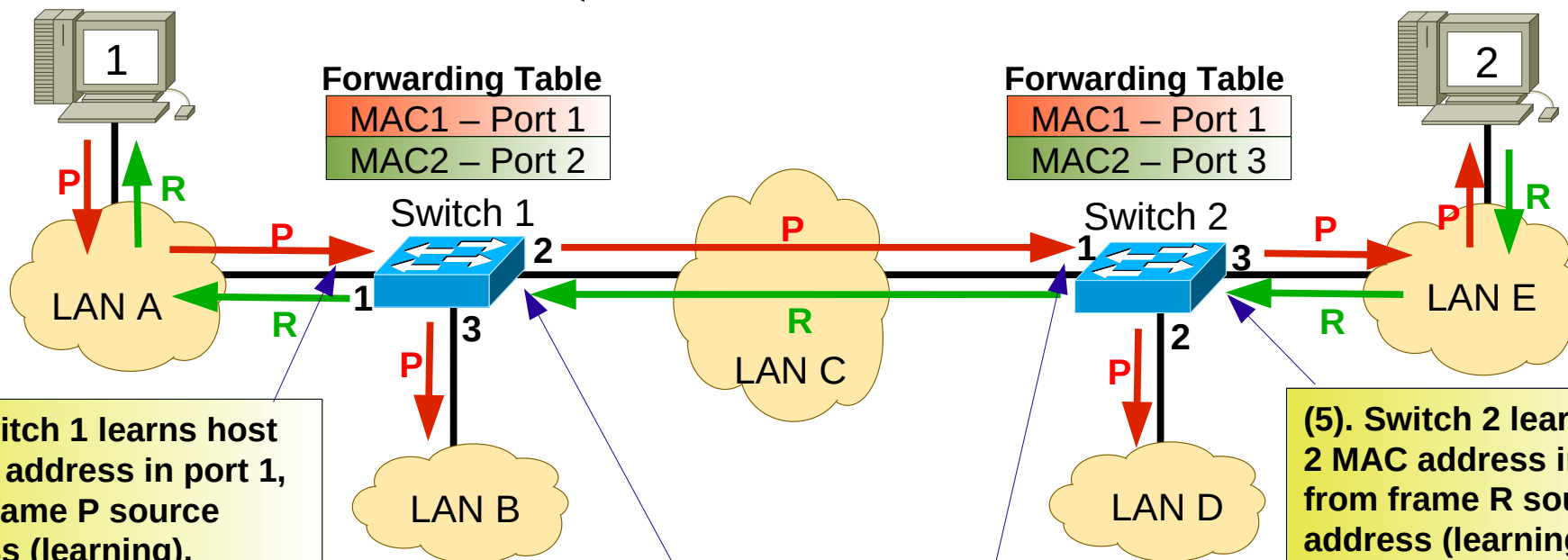
# Learning, Flooding and Forwarding

Frame P

Dest. = MAC2	Source = MAC1
--------------	---------------

Frame R (Answer to P)

Dest. = MAC1	Source = MAC2
--------------	---------------



Forwarding Table

MAC1 – Port 1
MAC2 – Port 2

Forwarding Table

MAC1 – Port 1
MAC2 – Port 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).

(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).

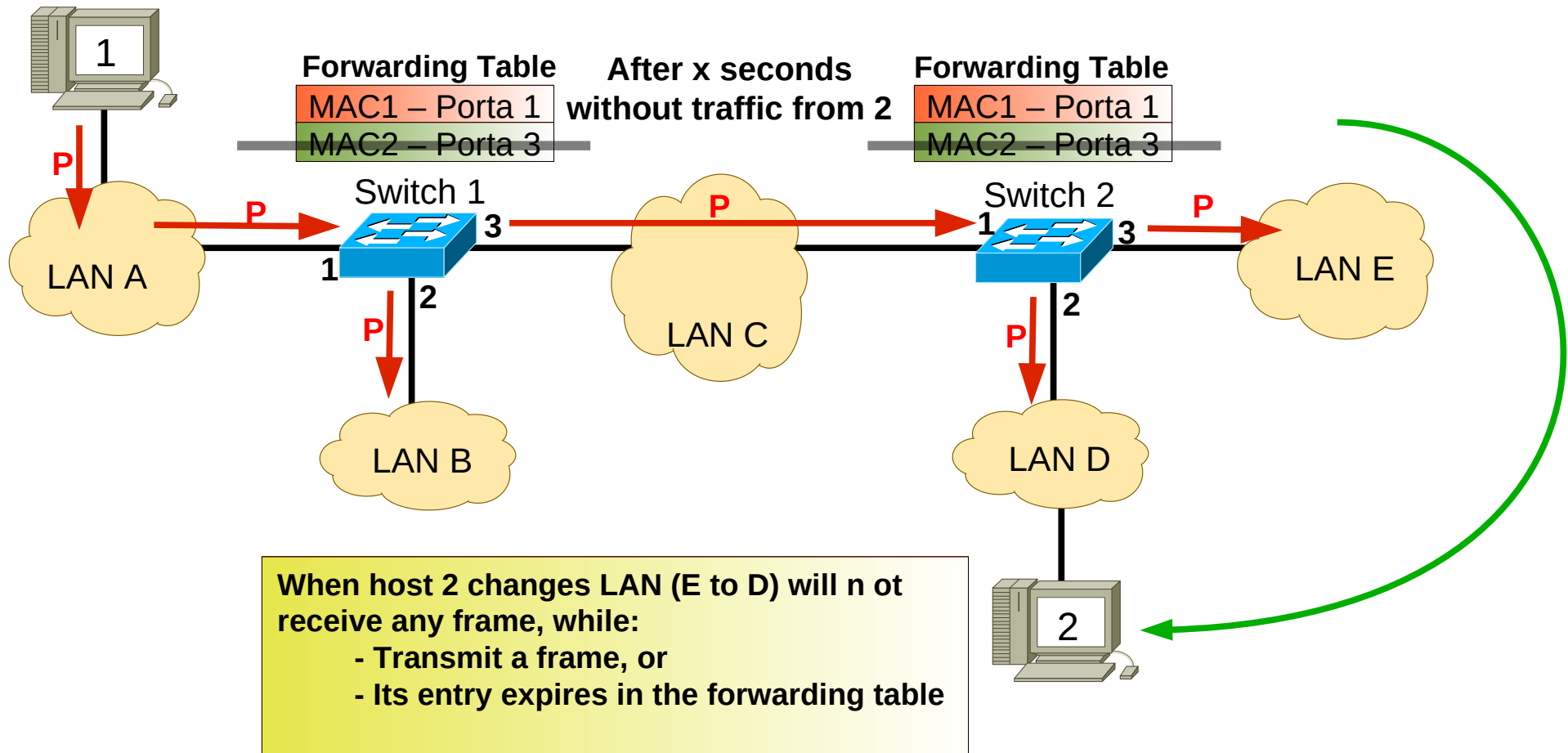
(5). Switch 2 learns host 2 MAC address in port 3, from frame R source address (learning).  
 (6). Switch 2 have frame's R destination (MAC 1) in the table, sends frame R to port port 1 (forwarding).



# Forwarding Table Aging Time

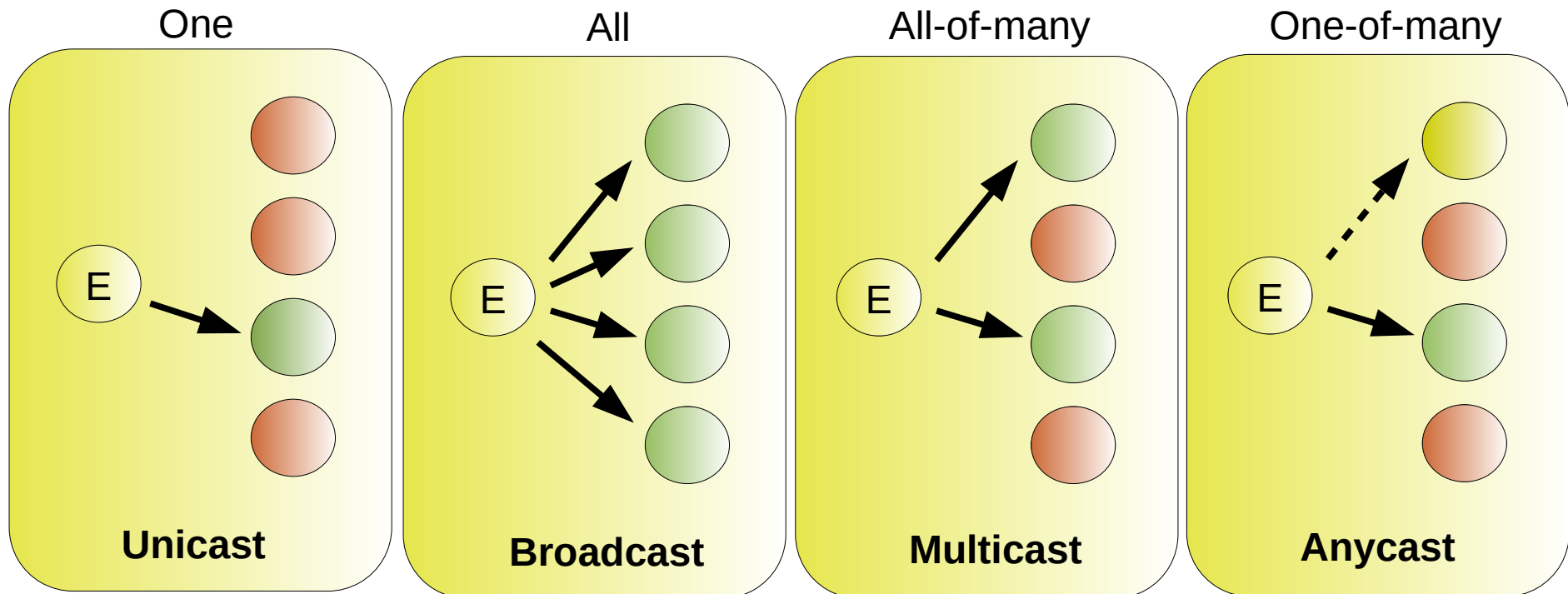
Frame P

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)

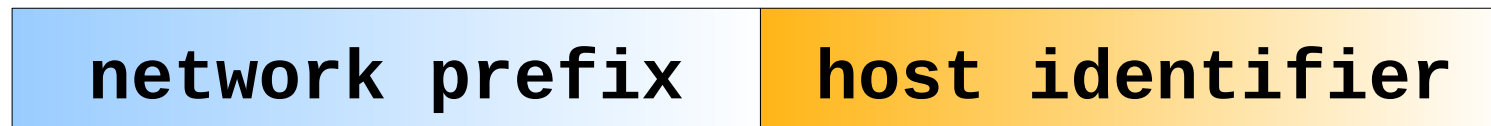


# IPv4 Addressing

- 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

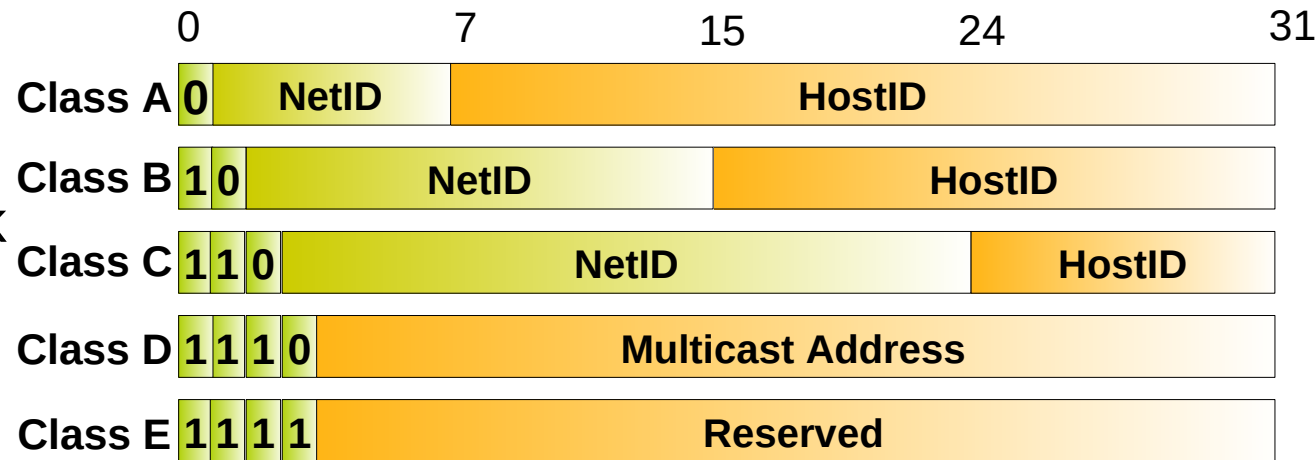
- The network prefix identifies a network and the host identifier identifies a specific host (actually, interface on the network).



- 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

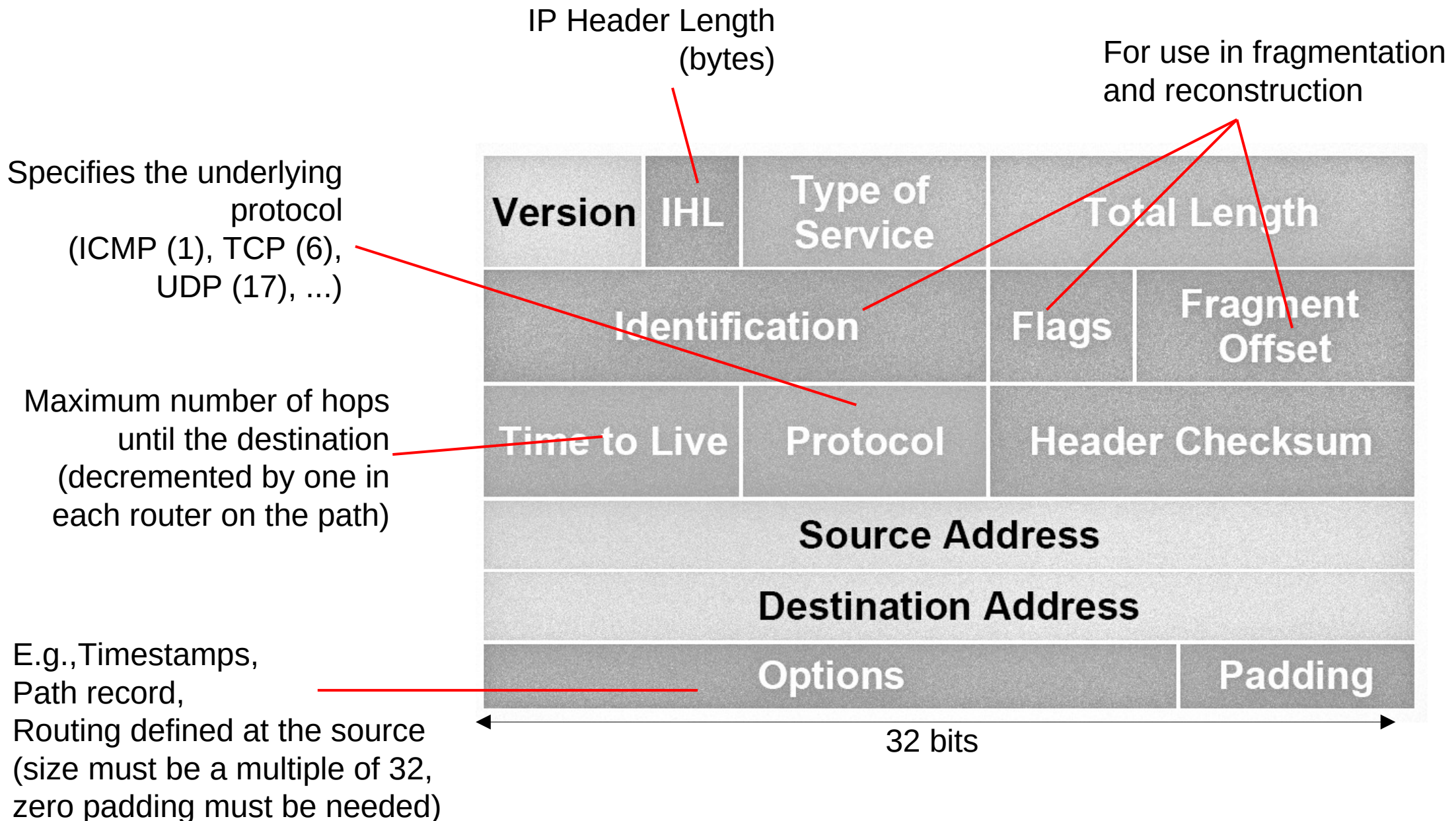
- There are two notations for IPv4 masks:
  - Decimal: 4 bytes separated by dots.
  - CIDR: A slash (/) a a number with the number of bits of the network prefix.
- Both notations still exist today.
  - CIDR starts to become prevalent.
  - IPv6 only supports CIDR.

CIDR	Decimal
/21	255.255.248.0
/20	255.255.240.0
/19	255.255.224.0
/18	255.255.192.0
/17	255.255.128.0
/16	255.255.0.0
/15	255.248.0.0
/14	255.240.0.0
13	255.224.0.0

CIDR	Decimal
/30	255.255.255.252
/29	255.255.255.248
/28	255.255.255.240
/27	255.255.255.224
/26	255.255.255.192
/25	255.255.255.128
/24	255.255.255.0
/23	255.255.254.0
/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 of 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 octets
  - 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)

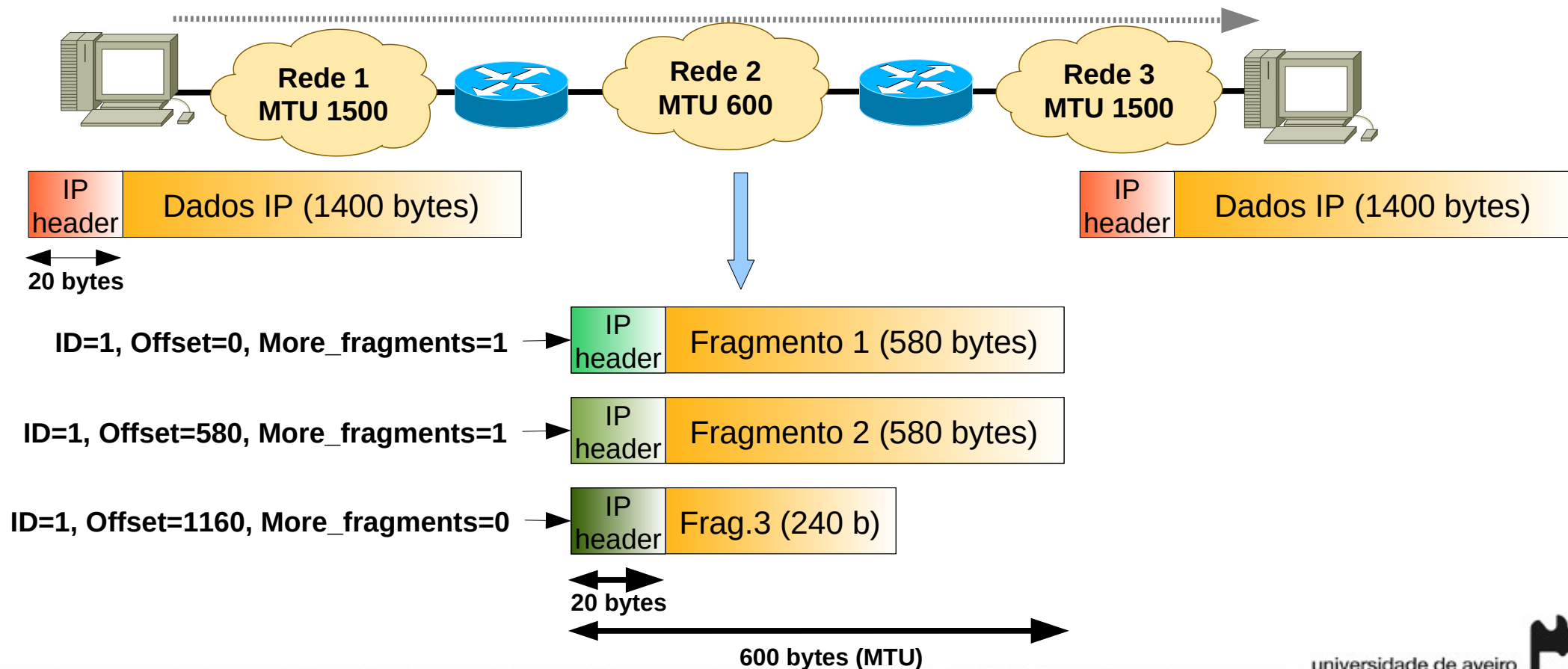
- Identification (2 bytes) – identifies fragments of the same original IPv4 packet.
- Flags (3 bits)
  - First bit for future use (always 0)  
Second bit is 0 if packet can be fragment, and 1 otherwise (do not fragment)
  - The third bit is 0 for the last fragment, and 1 otherwise (more fragments flag)
- 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)

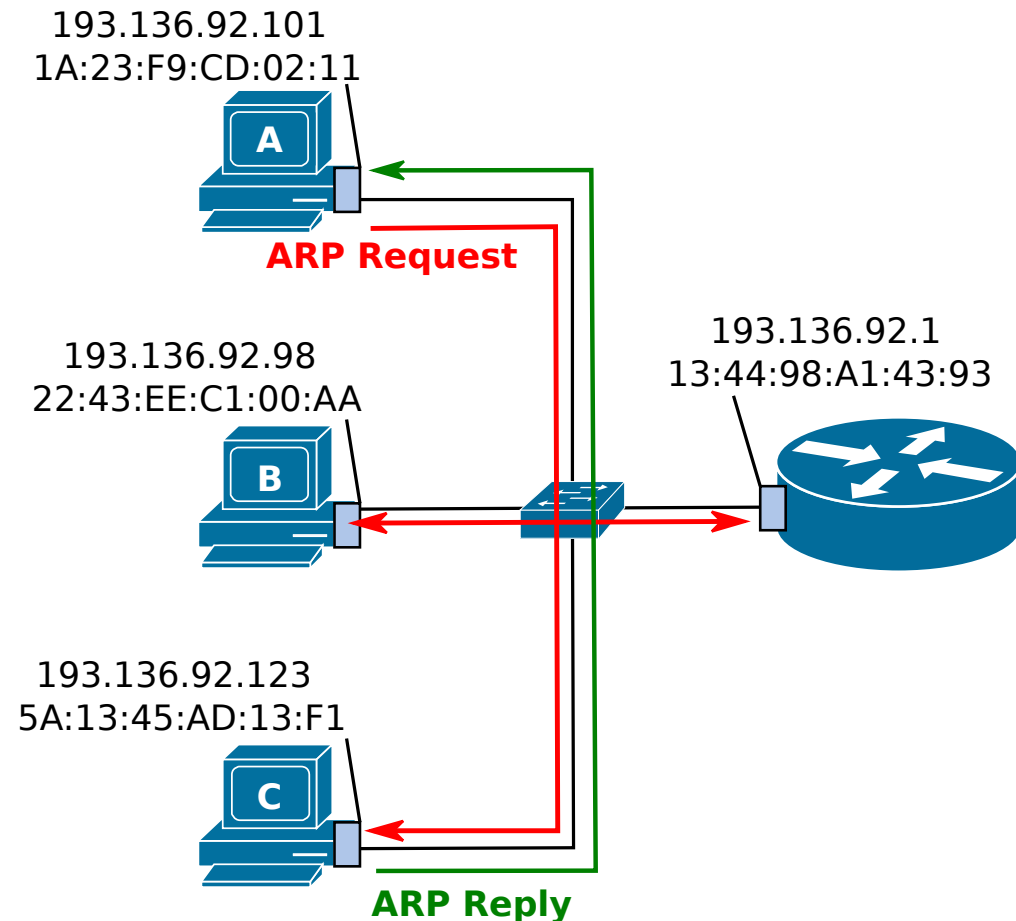
- Example:

- 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 it's on MAC address.

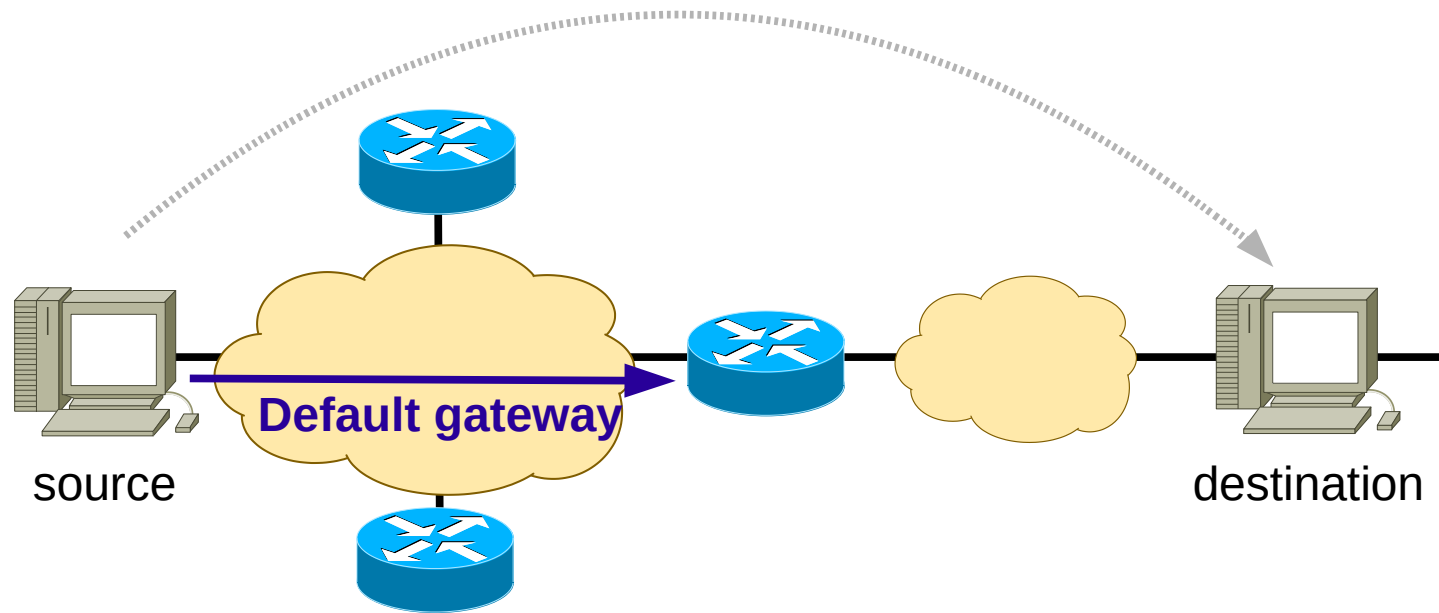
- MAC address resolution only happens in a the local network.

- ARP packets to not pass through routers.





# Routing to Another IP Network (1)

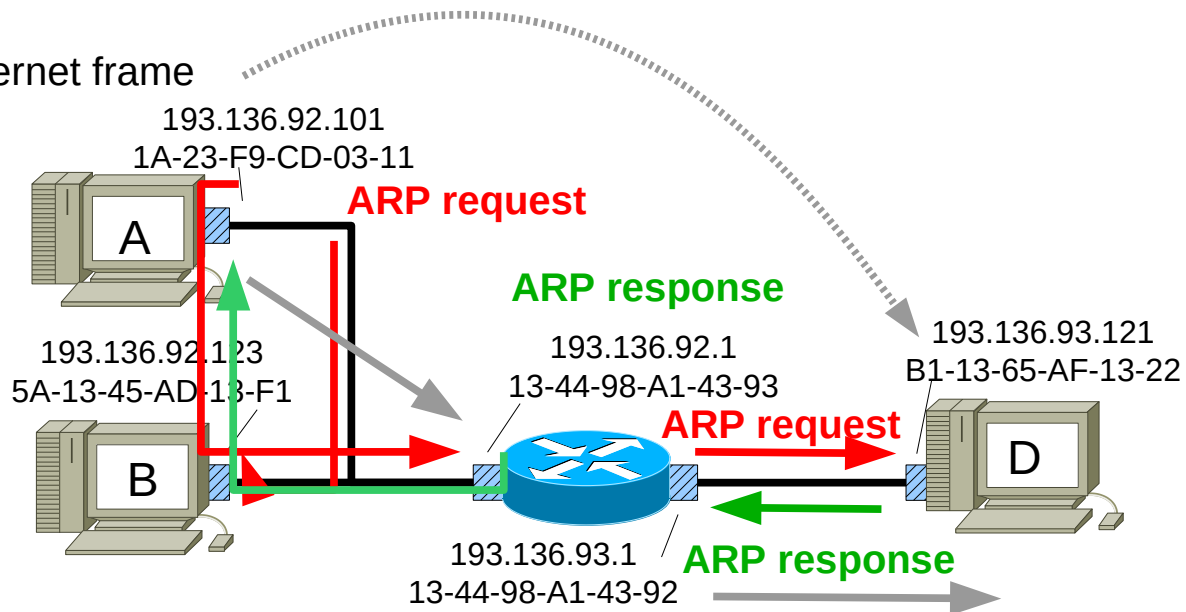


- When a host must send an IP packet to another IP, the packet must be sent to the **default gateway**.
- The **default gateway** must be provided at the same time than the IP address.
  - Manually or by self configuration.

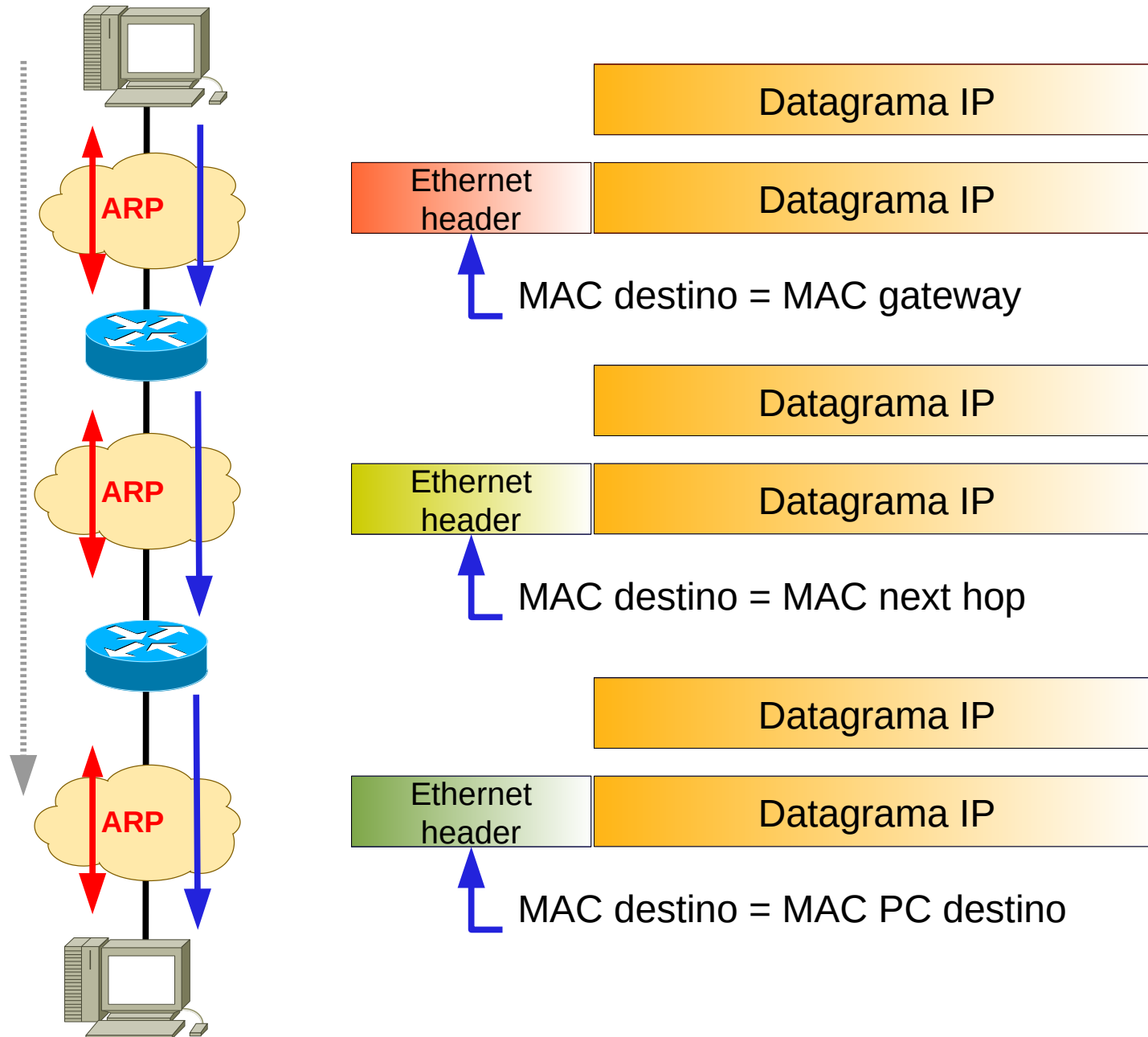
# Routing to Another IP Network (2)

- 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



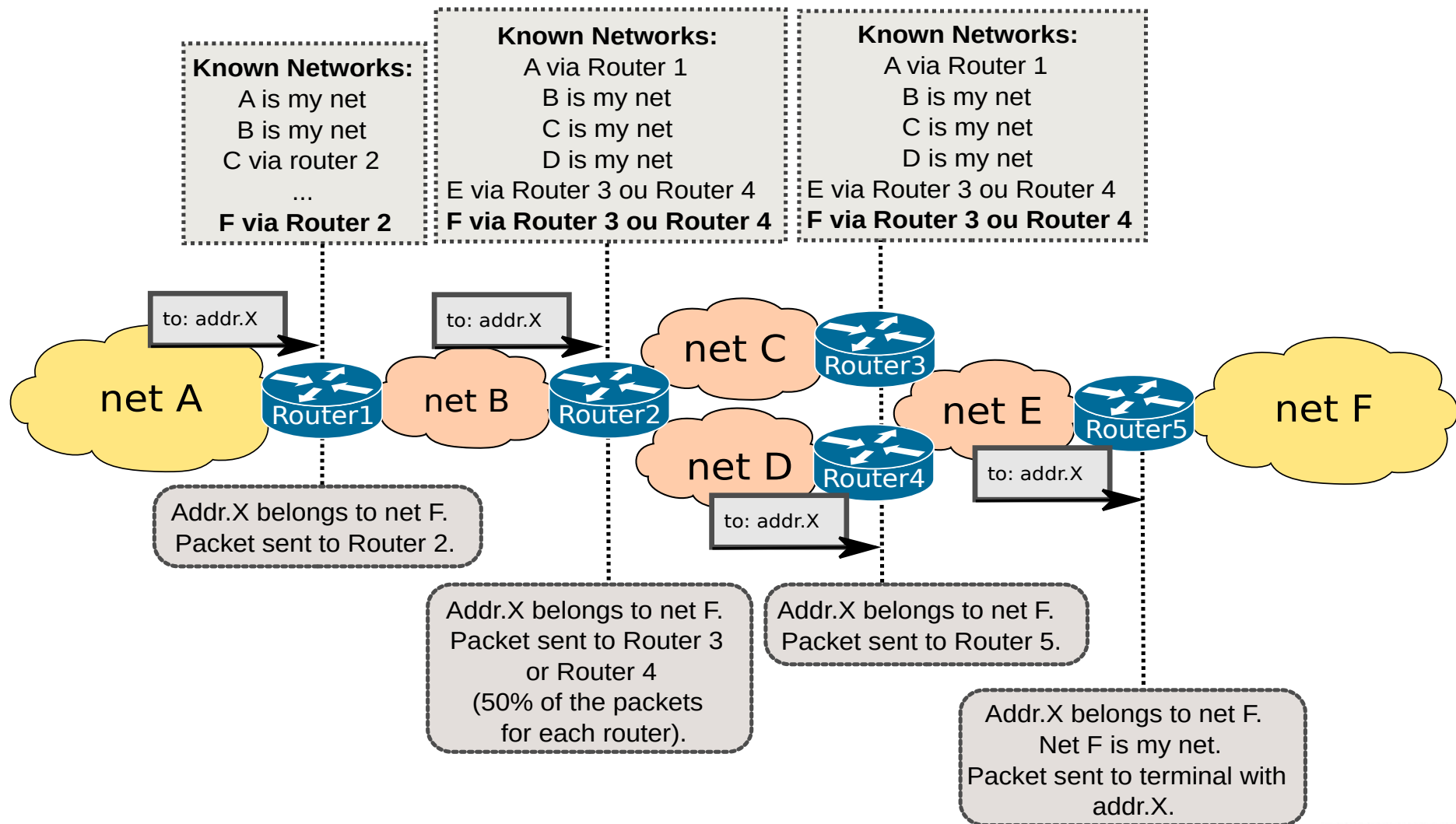
# IP Routing Overview (1)

- 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.



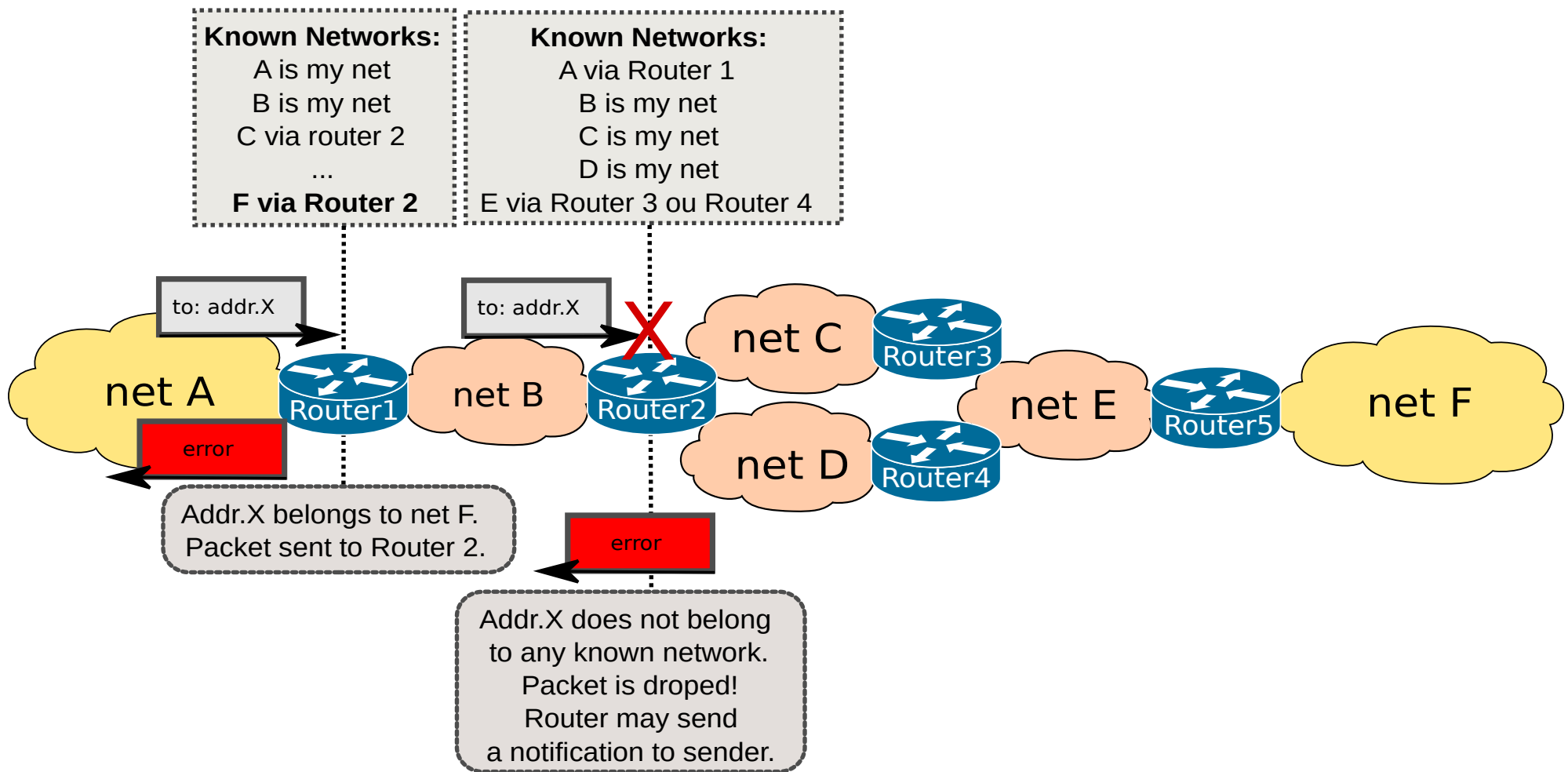
# IP Routing Overview (2)

- Hop-by-hop decision:
  - Based on the packets' **IP Destination Address**.
  - Rules listed on the **IP Routing Table**.



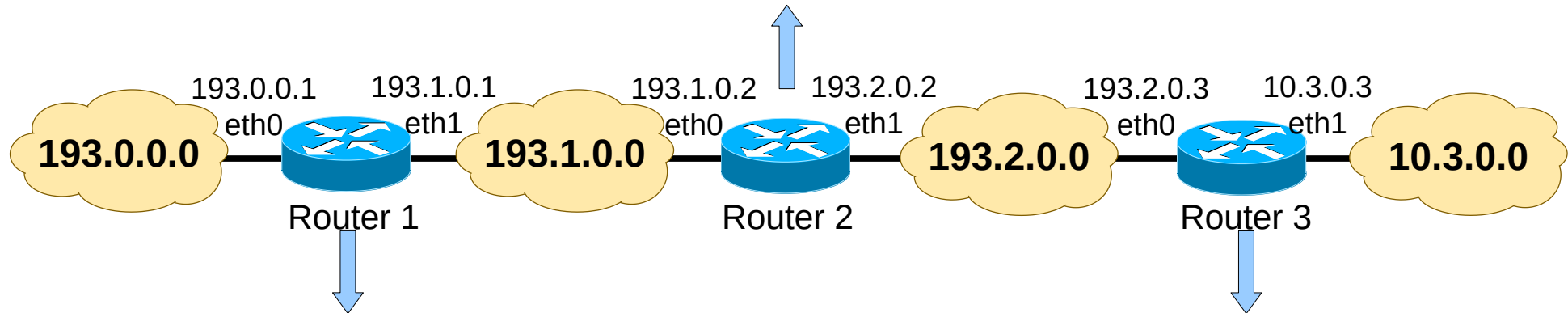
# IP Routing Overview (3)

- 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)

Dest.	Mask	Next hop	Interface	Metric
193.0.0.0	/24	193.1.0.1	eth0	1
193.1.0.0	/24	direct	eth0	-
193.2.0.0	/24	direct	eth1	-
10.3.0.0	/8	193.2.0.3	eth1	1



Dest.	Mask	Next hop	Interface	Metric
193.0.0.0	/24	direct	eth0	-
193.1.0.0	/24	direct	eth1	-
193.2.0.0	/24	193.1.0.2	eth1	1
10.3.0.0	/8	193.1.0.2	eth1	2

Dest.	Mask	Next hop	Interface	Metric
193.0.0.0	/24	193.2.0.2	eth0	2
193.1.0.0	/24	193.2.0.2	eth0	1
193.2.0.0	/24	direct	eth0	-
10.3.0.0	/8	direct	eth1	-



# 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

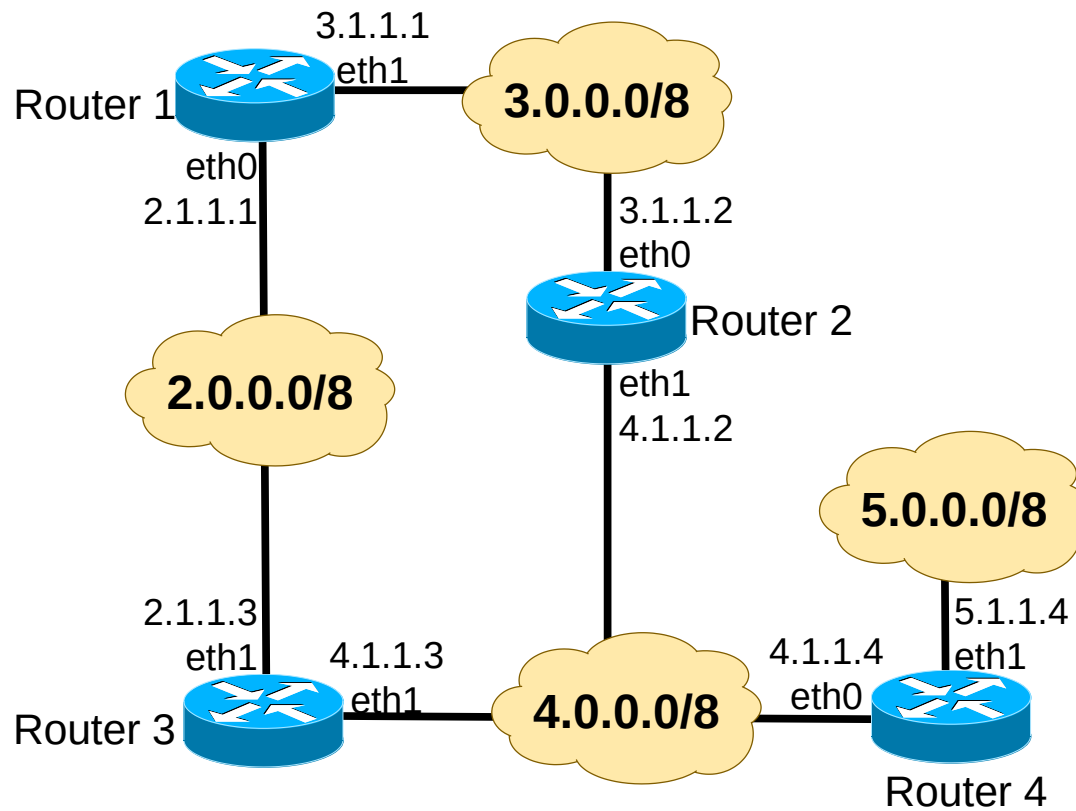
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	193.136.92.1	0.0.0.0	UG	100	0	0	enp5s0f1
169.254.0.0	0.0.0.0	255.255.0.0	U	1000	0	0	enp5s0f1
193.136.92.0	0.0.0.0	255.255.254.0	U	100	0	0	enp5s0f1

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

