IEEE 802.3 - Ethernet

COMPUTER NETWORKS A.A. 24/25

Leonardo Maccari, DAIS: Ca' Foscari University of Venice,
leonardo.maccari@unive.it

Venice, fall 2024

Sect. 1 The DataLink Layer

IEEE 802.3 - Ethernet

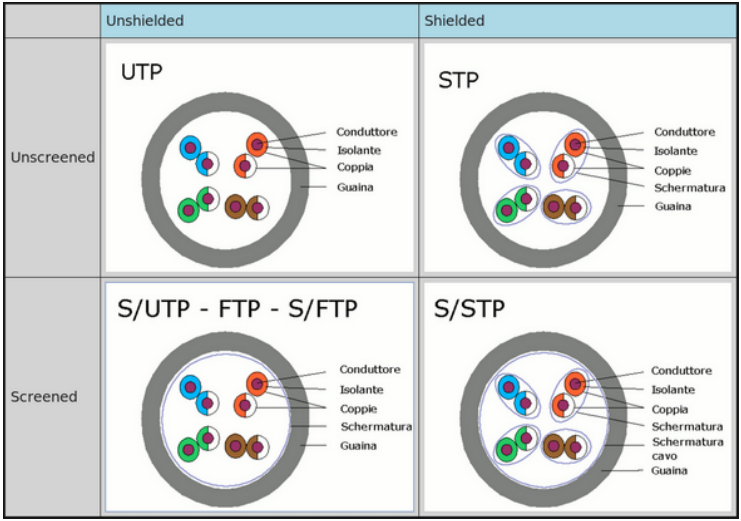
- What we call Ethernet is a family of standards designed starting from the early 70s in Palo Alto at Xerox
- The initial design is credited to Robert Metcalfe
- The IEEE 802.3 standards evolved greatly in the past 50 years, both in their physical layer and in their data link layer.
- Today Ethernet is mostly used on twisted copper cables and optic fiber.
- Some characteristics did not change, including the address used to identify a NIC and the Header format.

The Physical Layer

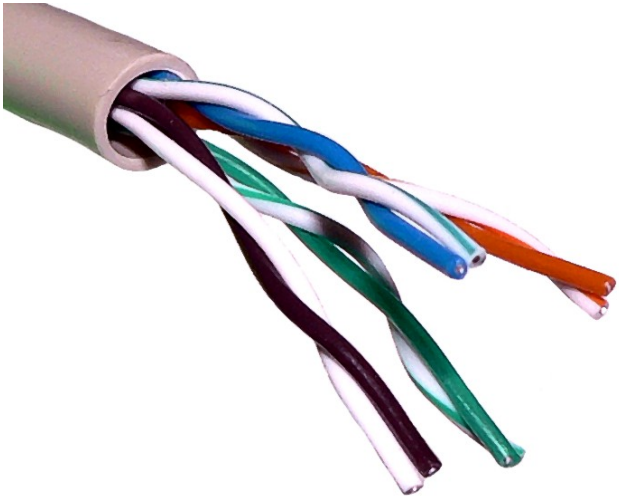
Twisted Pairs

- These are couple of copper cables twisted together, in each cable there is more than one pair (recent ones have 4 couples)
- There are several kinds of them with/without shields for external interference:
  - Unshielded: no shielding
  - Screened: there is a metal screen (kind of a metal *braid*)
  - Foiled: there is a metal *foil*
- So cables get their name out of the kind of protection [S/F]/[F]TP: (screened/foiled in the outer cable)/(foil for each inner pair) Twisted Pair.

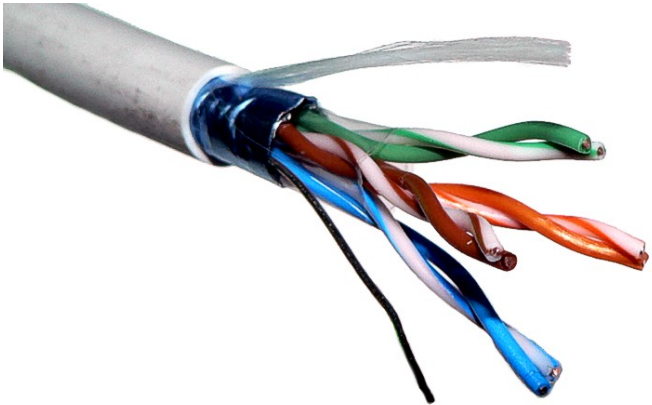
Shields and foils



Unshielded: UTP

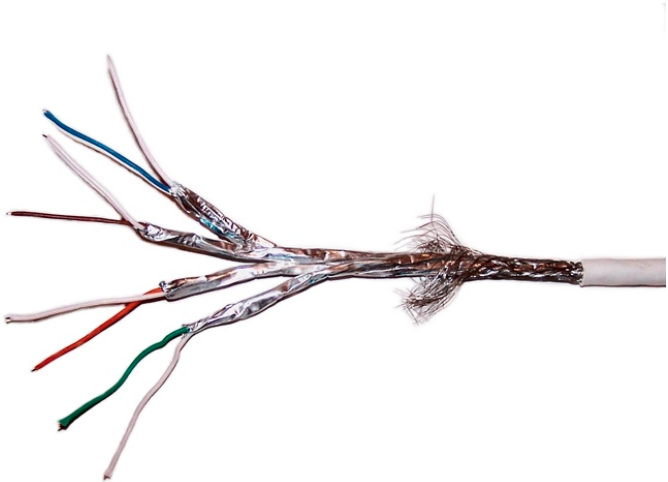


Foiled: F/UTP



The nylon string is pulled to break open the plastic shield in two.

Screened and Foiled: S/FTP



Cables and Speed

- Ethernet standards evolved a lot in the past 50 years, supporting higher and higher speeds
- And so did the cables, that are classified in Categories
- Each Category (cat 5/6/7/8...) has a different performance and can therefore be used with older or recent versions of Ethernet
- There is a nice Wikipedia table summarising this: `https://en.wikipedia.org/wiki/Ethernet_over_twisted_pair#Variants`
- Let's note one detail:

Back to Nyquist: 40GBASE-T

Name	Standard	Status	Speed (Mbit/s) [A]	Pairs required	Lanes per direction	Data rate efficiency (bit/s/Hz) [B]	Line code	Symbol rate per lane (MBd)	Bandwidth [C] (MHz)	Max distance (m)	Cable [D]	Cable rating (MHz)	Usage
40GBASE-T	802.3bq-2016 (CL113)	(not marketed)	40,000	4	4	6.25	PAM-16 RS-FEC (192, 186) LDPC	3,200	1,600	30	Cat 8	2,000	LAN, Data Center

- 40Gb/s total, 4 twisted pairs, each one using 1.600MHz.
- Modulation: PAM-16 (pulse-amplitude, 16 levels)
- Nyquist says: $C[Mb/s] = 4 \times 1600(2 \times log_2(16)) = 51.2Gb/s$
- However the real efficiency is said to be 6.2

$$C[Mb/s] = 4 \times 1600 \times 6.25 = 40Gb/s$$

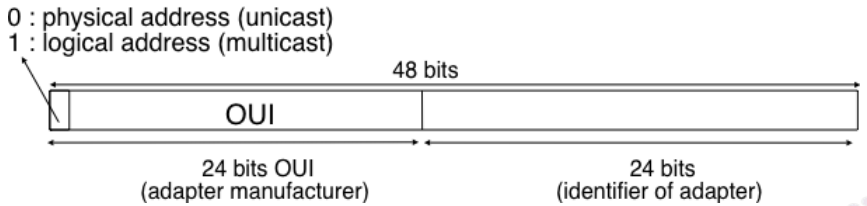
Ethernet Services

- Ethernet offers unicast, multicast, broadcast delivery of frames
- In a theoretically unreliable way, but modern Ethernets ensure a pretty reliable delivery of frames without reordering.

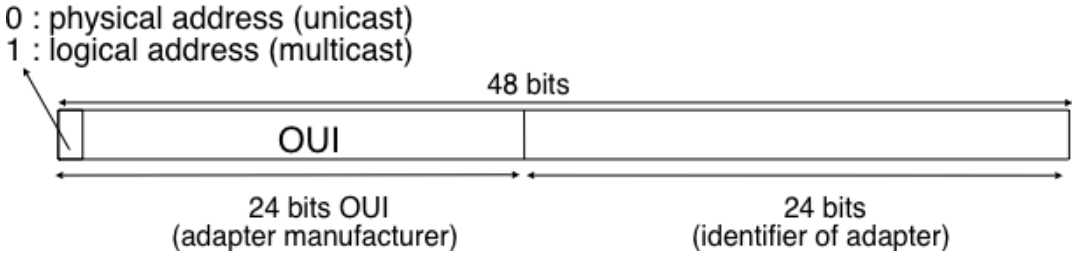
MAC Addresses

A MAC Address identifies a NIC at the data link layer. It is made of 48 bits of which:

- Un bit for the type ($0/1 \rightarrow$ unicast/broadcast)
- 24 bit for the Organisation Unique Identifier (OUI), unique per NIC manufacturer
- 24 bit unique for the NIC
- A MAC address should then be globally unique: `f8:75:a4:6a:97:33`
- They are pre-assigned to the NIC, you don't have to configure them.

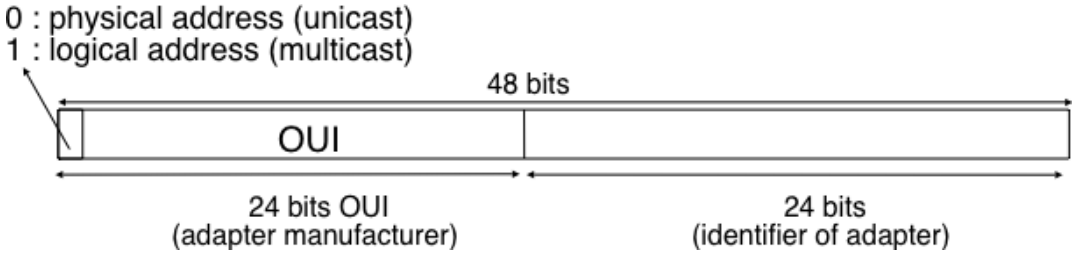


MAC Address Details



- The figure shows the unicast/broadcast bit as the first one, as they are transmitted on the cable.
- However, Ethernet encodes strings (an address is a string) reversing the bit order: least significant bit first
- This means 01:80:C2:00:00:08 is transmitted starting from 1000000...

MAC Address Details

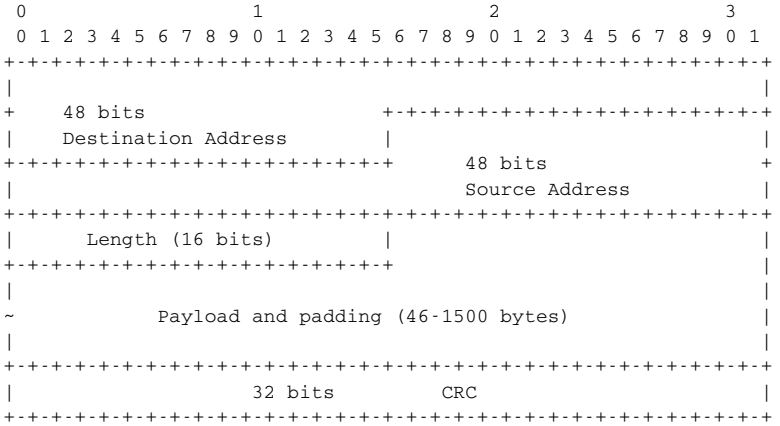


- Then 01:80:C2:00:00:08 is a multicast address
- A MAC address is unicast if its first Byte is even, the second nibble must be in (0,2,4,6,8,A,C,E)
- The OUI numbers assigned to manufacturers are all unicast `http://standards.ieee.org/develop/regauth/oui/oui.txt`

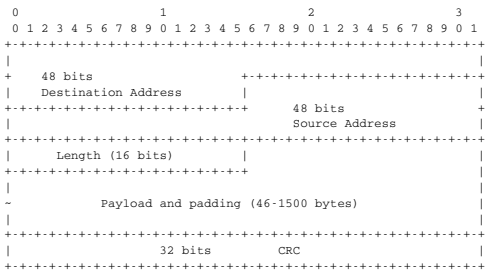
MAC Addresses and IPv6 Interface IDs

- We mentioned MAC addresses as a mean to assign an unique address to an IPV6 host.
- MAC addresses a part of a family of global identifiers that IEEE defines. Extended Unique Identifiers can be 48 or 64-bit long.
- A MAC address can be extended to become an EUI-64, so it can be used as the Interface ID in an IPv6 address.

Ethernet Frame Header



Ethernet Frame Header



- Destination and Source Addresses
- Length Field
- Payload
- CRC32 checksum

Frame Details

Some comments on the header organization

- Addresses are at the beginning of the frame, specifically, the destination address is the first one because a receiver NIC can read it first and decide if it is interested in decoding the rest of the frame, or just ignore it
- The CRC is at the trailer of the frame, contrarily to the IP headers, this is because the NIC can compute the CRC as it receives the data, and then compare it with the last 32 bits, without necessarily storing the data or the CRC.
- Ethernet is used mostly with the IP protocol but not necessarily. A MAC layer should include a field that specifies the upper protocol. In the initial Ethernet frame this field existed (the EtherType) and it replaced the Length field.
- However, there was no `length` field, as the frame had fixed minimum length and there must be an inter-packet gap (no transmission) at end a frame.

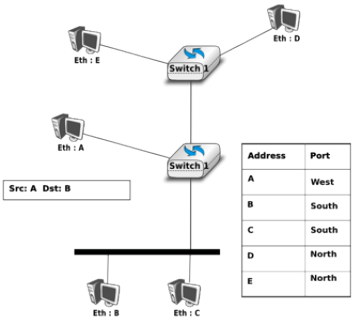
LLC Layer

- As a consequence, a new layer was introduced in the ISO/OSI model, so the Data Link layer is split in two: the MAC layer (media access control) and the LLC (logical link control).
- As the role of the second one is marginal, often times we use the MAC name as a synonym of Data Link layer.
- However, at some point the 16 EtherType bit format was modified to accomodate also the frame length.
- Then the LLC has lost most of its utility and it is rarely used.

The DataLink Layer

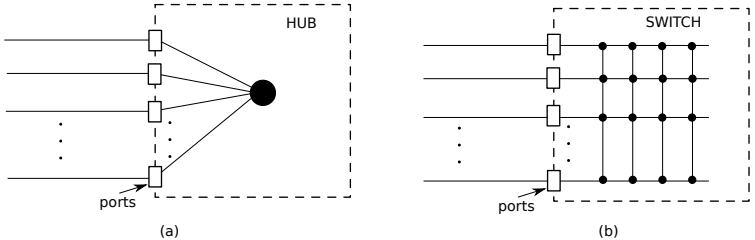
↳1.1 Ethernet Switches

Extending the Link



- A single link can be extended to create a whole network with devices that interconnect terminals one to the other
- This used to be done with Hubs, and now it is done with Ethernet Switches

Hubs and Switches



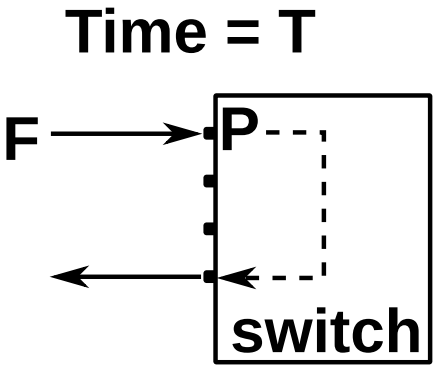
- A hub (a) is just a repeater, every frame that arrive to a port is sent to all other ports. It is now obsolete.
- A switch (b) understands the 802.3 standard
- It has a "backplane" that is a control logic that selectively connects one port with another and sends traffic only on the right port
- Thus it has to learn on which port a certain frame should be sent to reach the destination.

Backward Learning and Spanning Tree

- In the simplest situation the network is made of a single switch and many terminals, then the switch needs to learn where a certain terminal is. This is obtained with a *backward learning* algorithms
- In the more complex situation, there are more switch in cascade, and another algorithm must be implemented to avoid loops: the Spanning Tree (distributed algorithm and protocol)

Backward Learning (pseudocode)

```
# Arrival of frame F on port P at time T;
# Table : addr->[port, time]; Ports : list of ports
src=F.SourceAddress
dst=F.DestinationAddress
# update the table
Table[src] = (P, T)
if isUnicast(dst) :
 if dst in Table:
   ForwardFrame(F,Table[dst][0])
   return
# multicast, broadcast, or unknown destination
for o in Ports :
 if o!=P :
   ForwardFrame(F,o)
```



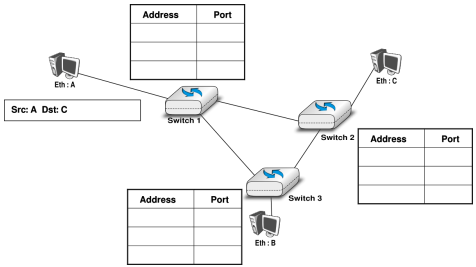
Periodic Cleaning of the Table

- Periodically, the switch checks the table and removes all stale addresses
- Stale addresses are those that have not been used in a certain time interval
- This is meant to clean up the table and make room for new MAC addresses

The DataLink Layer

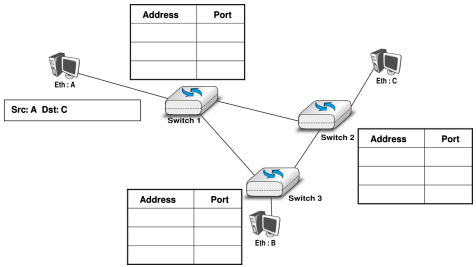
↳1.2 The Spanning Tree Protocol (STP)

Spanning Tree Protocol: switching loops



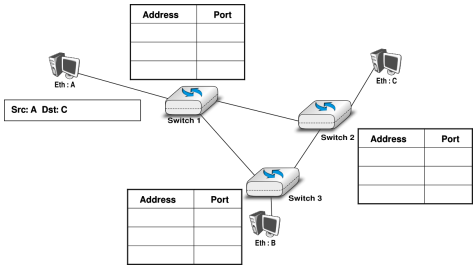
- What happens when this network boots from zero?
- All hash tables are empty
  - A sends a frame to C, switch 1 doesn't know where C is, then forwards on all ports

Spanning Tree Protocol: switching loops



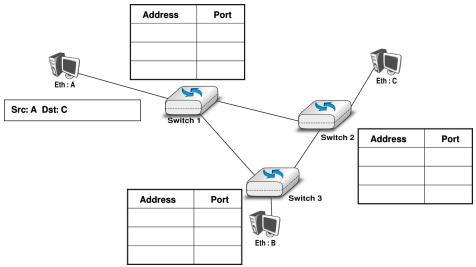
- What happens when this network boots from zero?
- All hash tables are empty
  - A sends a frame to C, switch 1 doesn't know where C is, then forwards on all ports
  - Switch 2 doesn't know C, so forwards on all ports

Spanning Tree Protocol: switching loops



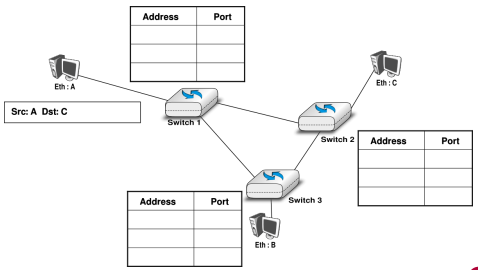
- What happens when this network boots from zero?
- All hash tables are empty
  - A sends a frame to C, switch 1 doesn't know where C is, then forwards on all ports
  - Switch 2 doesn't know C, so forwards on all ports
  - Switch 3 receives from 1 and 3, does not know C, then forwards on all ports

Spanning Tree Protocol: switching loops



- What happens when this network boots from zero?
- All hash tables are empty
  - A sends a frame to C, switch 1 doesn't know where C is, then forwards on all ports
  - Switch 2 doesn't know C, so forwards on all ports
  - Switch 3 receives from 1 and 3, does not know C, then forwards on all ports
  - Switch 1 receives frame 1 and 2 ...

Spanning Tree Protocol: switching loops



- What happens when this network boots from zero?
- All hash tables are empty
  - A sends a frame to C, switch 1 doesn't know where C is, then forwards on all ports
  - Switch 2 doesn't know C, so forwards on all ports
  - Switch 3 receives from 1 and 3, does not know C, then forwards on all ports
  - Switch 1 receives frame 1 and 2 ...

Ethernet does not provide a TTL field: frames are delivered but they may travel endlessly

Switching Loops

- A switching loop can saturate the capacity of the links, overload the CPU of the switches and kill your network
- It's not easy to block, you have to break the physical loop
- It's easy to create: just connect a switch to two ports on a network (don't do this at home)
- On the other hand, having a redundant network is important, because it resists failures

The Spanning Tree Protocol (STP), IEEE 802.1d

- It was proposed by Radia Perlman in 1985
- It is assumed that each switch has a unique 64-bit identifier.
- The first 16 bits (most significant) are determined by the administrator.
- The remaining 48 bits are a MAC address determined by the manufacturer.
- STP constructs a tree with the root being the switch with the lowest ID.
- The configurability of the first bits allows the administrator to place the root close to the border router for performance reasons.

A spanning tree is not a minimum cost tree. But it covers all the nodes of a graph and contains no loops.

Port State

- The ports of a switch that support STP can be in three states: *root*, *designated*, *blocked*
- STP is a distributed protocol. As long as all the ports are not set to some state, Initially ports do not forward traffic, till they reach some state.
- At convergence instead:
  - The root switch has all ports set to *designated*
  - All other switches have at least one port set to state *root*
  - All other ports are either *blocked* or *designated*

STP: Simplified Operation

- Each switch periodically sends a Bridge Protocol Data Unit (BPDU) frame on all its ports.
- The frame contains various data, including $< Root_{id}, Cost, Sender_{id}, p >$:
  - The ID of the currently designated root switch
  - the cost of the path to the root (weighted distance based on link capacity),
  - the ID of the node generating the BPDU,
  - p: the port number on which the BPDU was sent.
- BPDUs are sent to a specific multicast address, they are never forwarded and processed only by neighboring switches.
- Initially every switch uses its own ID as the root ID and cost set to zero

Comparing BPDUs

- BPDUs can be compared using lexicographic sorting (i.e. one filed is compared, if there is a tie, move to the next...).
- If $Root_{id}$ is different, the it is sufficient to break ties. So if we have:

$$BPDU =< Root_{id}, Cost, Sender_{id}, p >$$

$$BPDU' =< Root'_{id}, Cost', Sender'_{id}, p' >$$

then

$$\text{if } Root_{id} < Root'_{id} \text{ then } BPDU < BPDU'$$

if not, compare $Cost$, and so on...

STP Basics

- When switch A receives a BPDU from switch B, A calculates the cost c:

$$c = Cost + C_{AB}$$

and the *root priority vector*. That is stored for each port:

$$V_q =< Root_{id}, c, Sender_{id}, p, q >$$

where: C_{AB} is the cost of link A-B, q is the port over which the BPDU was received. The other parameters come from the BPDU received by A.

- In modern versions of STP the cost of a link is as follows:

Capacity	Cost
100 Mbps	200000
1 Gbps	20000
10 Gbps	2000
100 Gbps	200

STP Basics

- If V_q is lower than the its own BPDU (essentially, the received $Root_{id}$ is smaller or the cost is smaller), then:
  - A decides that B becomes the root switch, and will use the ID of B as the $Root_{id}$ in the BPDU it generates from now on
  - Port q in A becomes the *root* port. Now A is a child of B in the tree.

STP Basics

- Moreover, A will also compare the BPDU received on port q with its own BPDU (**Note well**: its own BPDU Vs the received BPDU, not Vs the priority vector)
  - if its own BPDU is smaller, then q takes the *designated* state: B is a child of A.
  - Otherwise, B is a potential sibling of A (so there could be a loop) and q becomes *blocked*.

STP Port States

- No data packets are sent/accepted from blocked ports.
- BPDUs from all ports are instead continuously processed (in case something changes).
- BPDUs are sent only to children switches.

Port state	Receives BPDUs	Sends BPDU	Handles data frames
Blocked	yes	no	no
Root	yes	no	yes
Designated	yes	yes	yes

STP Updates

- The process is constantly repeated, with the root switch computing its BPDU and sending it to its designated ports.
- Switches maintain an "age" of the last received BPDU, incrementing it every second.
- When this value exceeds a maximum, the switch understands that the topology has changed: some neighbor switch failed
- The STP protocol in the switch starts over: all ports stop forwarding traffic, a new root is searched.

Sect. 2 ARP and DHCP

Bridging the Network and Data Link Layer

- We now know how an Ethernet layer is built
- NIC A can send packets to NIC B on another host, based on the destination MAC address
- However, we know that all the traffic is carried by IP packets.
- So we have to ask ourselves one question: *Whenever a host with a certain IP wants to send a packet to another IP, which MAC address will it use as the destination?*
- There are two possible answers

A Local destination IP

- Assume the IP of A is 192.168.1.10/24, assume the destination IP is 192.168.1.11.
- Then A knows that host B is in the same physical network. The IP must be reachable without routing, just through switches.
- Then in this case A needs a way to discover the MAC address associated to the IP of B, and send an Ethernet frame with the correct IP and MAC destination addresses.
- This is solved using the Address Resolution Protocol (ARP)

A Remote IP

- Now A wants to send a packet to 1.2.3.4, that is not in the same subnet.
- The correct behaviour is to send the packet using:
  - In the IP header, the destination IP will be 1.2.3.4
  - In the Ethernet header, the destination MAC must be that of the router that is the network gateway
- How does A knows the MAC address of the router? This (among other issues) is taken care by the ARP and DHCP protocols.

ARP and DHCP

↳2.1 The ARP Protocol

ARP: Address Resolution Protocol

- Each host on the network maintains an *ARP table*
- The table associates an IP address with a corresponding MAC address
- So the question is, how is the ARP table updated?

ARP: Address Resolution Protocol

- When `192.168.1.20` needs to know the MAC address that corresponds to a given `192.168.1.11`, it broadcasts an ARP packet containing its MAC and IP address, and a request for the MAC of the destination
- The packet, being broadcast, is forwarded by all the switches in the network
- The ARP request arrives at the destination host which responds with a unicast packet directed to the source MAC

ARP Packets Fields

Hardware type (ethernet)
Protocol type (IPv4)
[...]
Sender hardware address
Sender protocol address
Receiver hardware address
Receiver protocol address

ARP Algorithm, upon packet arrival

```
?Do I have that hardware type ?
Yes: (almost definitely)
  ?Do I speak that protocol ?
  Yes:
    If the pair <protocol type, sender protocol address> is
        already in my translation table, update the sender
        hardware address field of the entry with the new
        information in the packet.
    ?Am I the target protocol address?
      Yes:
        [...]
```

- The ARP table is updated at each ARP packet received (both requests and replies)

ARP and DHCP

↳2.2 The DHCP Protocol

Booting a Host

- When a host boots, it needs several configuration parameters, among which:
  - An IP address that it can use: we know that every network has its own netid but the host doesn't know it
  - The IP address of the router that is the gateway of the network
  - The IP address of the DNS server

Dynamic Host Configuration Protocol (DHCP)

- DHCP is a management protocol that is used to configure the network hosts when they boot
- It is a conceptually simple protocol, that we describe here because it concerns the configuration of local networks and not the data exchange
- We will describe it functionally, not in the details

DHCP

- DHCP is a client-server protocol, the administrator sets-up a server that any host can contact to receive its own configuration
- DHCP uses UDP, on port 67
- The protocol is initiated by the client

DHCP Discover

- A newly added client switches on and sends a DHCP discover message. The destination IP of the IP packet is limited broadcast: 255.255.255.255 (or the equivalent for IPv6)
- The source IP address is set to 0.0.0.0
- The source MAC address is set to the NIC address MAC
- In the DHCP discover message the client also includes a client ID, normally the MAC address of its NIC
- The client specifies if it prefers to receive answers to the broadcast IP or the not-yet-assigned unicast IP[1]

[1] DHCP makes a "*creative*" use of IP, assuming the client receives unicast packets even before having an IP, see RFC2131.

DHCP Offer

- The server will answer with an Offer message, containing client ID (taken from the discover), the IP address that the server is offering, the subnet mask, the lease duration
- The offer contains also the IP of the DHCP server
- There can be more than one server in the network that may respond to the same offer, for redundancy reasons

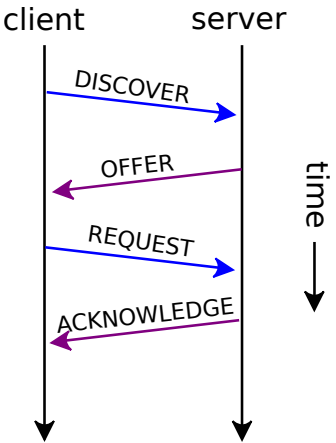
DHCP Request

- The client will then send a DHCP request message, that requests the IP that the server offered
- At the MAC and IP layer, the destination address is the broadcast address, but the message body contains the server IP address
- Any DHCP server receiving a request will check the server address included in the body of the packet and decide if to process or ignore the message.

DHCP Acknowledgment

- The server will reply finally with an acknowledgment message
- This is destined to the IP that the client requested
- It also contains the lease duration and any other configuration information: normally the default gateway and the DNS server

DHCP Summary[2]



[2]Image by Gelmo96, CC BY-SA 4.0, via Wikimedia Commons

ARP and DHCP

↳2.3 Communicating with the Internet

Putting all together

- When `192.168.1.1` needs to communicate with 1.2.3.4, it will follow this process:
  1. Check that 1.2.3.4 is not part of its network
  2. Then the destination MAC address of the frame must be the one of the gateway, this is was known at device configuration via `DHCP`
  3. If the MAC address of the gateway is not in the ARP table, then make an ARP resolution
- Once the MAC address of the gateway is known, send the frame to the MAC of the gateway using 1.2.3.4 as destination IP