# Lab #6: The ARP protocol.

Previous reading: Kurose2017, section 6.4.1

Videos: https://www.youtube.com/watch?v=2XdAXD3uS8c

https://media.upv.es/#/portal/video/8e5cbcf2-0f19-8740-988f-536217d4442a

## *1. Introduction.*

Each of the hosts and routers that are connected to the Internet is identified by a network layer address: IP address. But, in addition, each network adapter installed in each node will have a link layer address: MAC or physical address.

In this lab we will study the addressing at the link layer and the Ethernet Address Resolution Protocol (ARP). **ARP provides the link layer address (MAC address) for a given IP address to any sender in the same subnet as the destination.**

## *2. Physical addresses*

Link layer addresses are assigned to network adapters and are called MAC address (also known as physical address or LAN address.)

The MAC address is 48 bits (6 bytes) long, which gives us $2^{48}$ possible MAC addresses. These addresses are usually expressed in hexadecimal notation, with each byte indicated as a pair of hexadecimal numbers. We can see it in the following figure:

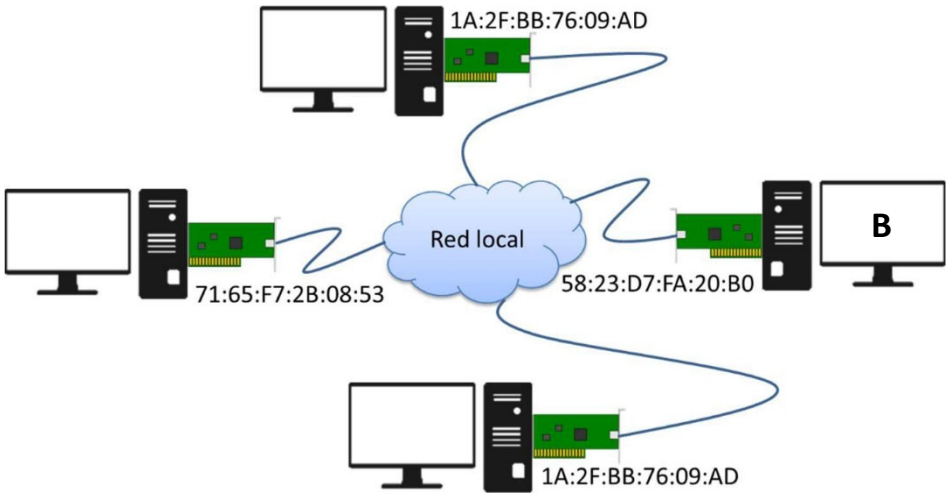



Figure 1. Local Area Network (LAN)

The physical or MAC address of an adapter has a flat structure (not hierarchical like an IP address) and will keep its value, even if the node moves from one LAN to another.

IEEE is responsible for managing the physical address space, ensuring that these addresses are unique, regardless of manufacturer and network. When a company wants to manufacture adapters, it must buy a part of the address space consisting of $2^{24}$ addresses. IEEE allocates the $2^{24}$ address block by setting the first 24 bits of its physical addresses and letting the company design unique combinations of the last 24 bits for each adapter.

When two nodes belonging to the same network want to communicate, they will need to know not only the other's IP address but also their physical address. **A computer only needs to find out the physical address of another if they both share the same IP network.**

# 3. Ethernet ARP protocol (RFC 826)

For an IP datagram to travel over the local area network, it must be encapsulated within a frame (Link layer transfer unit, Ethernet frame or Wireless frame). In this lab. we will study the case of Ethernet frames.

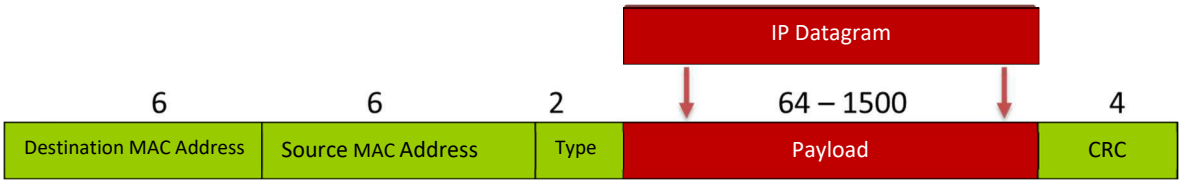



Figure 2. Ethernet frame

Ethernet frame contains the physical address of the next destination, the next hop in the way from Source host (source IP address) to final Destination host (destination IP address).

**The destination MAC address will belong to the final destination if source and destination belongs to the same LAN, on the other hand, if the final destination doesn't belong to the same LAN as the source, then the destination MAC address will belong to the router of the LAN, in charge to forward packets outside of the LAN.**

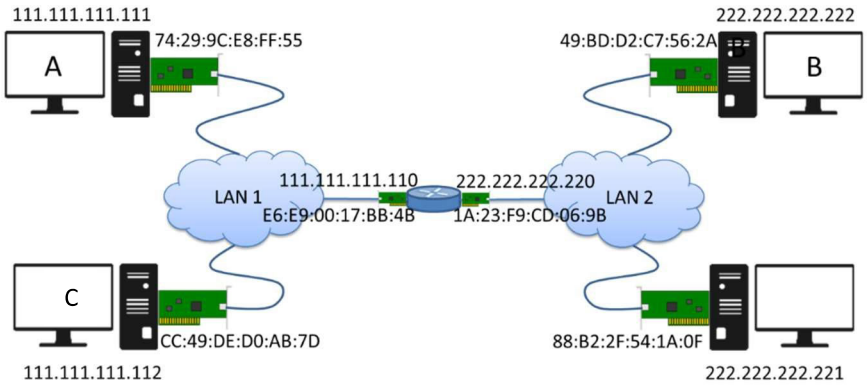



Figure 3. LAN1 and LAN2 Topology

Then the question is: How does the source host know if the destination is connected to same LAN as it?

Let's see it with two cases based in the topology of figure 3:

**Case 1: Source and Destination are in the same LAN**

1. Node A will generate a datagram with source IP address 111.111.111.111 and destination IP address 111.111.111.112 (Node C).
2. Node A checks its forwarding table:

| Destination Net. | Netmask | Gateway | Interface |
|---|---|---|---|
| 111.111.111.0 | /24 | 0.0.0.0 | 111.111.111.111 |
| 0.0.0.0 | 0.0.0.0 | 111.111.111.110 | 111.111.111.111 |

Making 111.111.111.112 AND 111.111.111.0 /24 A determines that the destination belong to the same network as it, so the next hop is the final destination, so it will have to obtain the MAC address mapped to the destination IP address 111.111.111.112.

**Case 2: Source and Destination are in different LANs**

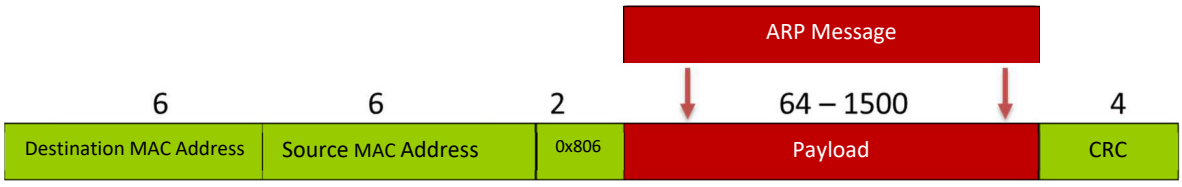
1. Node A will generate a datagram with source IP address 111.111.111.111 and destination IP address 222.222.222.222 (Node B).
2. Node A checks its forwarding table:

| Destination Net. | Netmask | Gateway | Interface |
|---|---|---|---|
| 111.111.111.0 | /24 | 0.0.0.0 | 111.111.111.111 |
| 0.0.0.0 | 0.0.0.0 | 111.111.111.110 | 111.111.111.111 |

Making 222.222.222.222 AND 111.111.111.0 /24 A determines that the destination doesn't belong to the same network as it, so it will choose the next entry, and the next hop is the router, so it will have to obtain the MAC address mapped to the router address 111.111.111.110.

**Then, the point now is how IP addresses and MAC addresses are related. Well, they are not. But given that MAC addresses are needed to send a datagram to the right destination, an IP address mapping to MAC address is needed. That is the purpose of the Address Resolution Protocol (or ARP). ARP provides the MAC address for a given IP address to any sender in the same subnet as the destination.**

ARP protocol messages belong to the link layer and are encapsulated in the data field of a frame. The Type field in the frame header will identify the message type: 0x806 identifies the ARP protocol in the case of Ethernet frames.



To find out a MAC address, the link layer will send an ARP query packet that will contain the source IP address, source MAC address, and destination IP address. This message will be addressed to all nodes in the LAN, using the broadcast MAC address: FF: FF: FF: FF: FF: FF as the destination address. **Let's see the frames generated in Case 1, when A (111.111.111.110) wants to send an IP datagram to C (111.111.111.112).**

| | | | | |
|---|---|---|---|---|
| FF:FF:FF:FF:FF:FF | A's MAC Address | 0x806 | ARP Query: Who has 111.111.111.112? | CRC |

This ARP query reaches all nodes on the LAN. Each node checks if the IP address included in the ARP queried belongs to it. The one whose IP address matches will reply to this by an ARP reply message in which they will add their MAC address. This message will no longer be sent by broadcast but will be sent to the node that made the query.

| | | | | |
|---|---|---|---|---|
| A's MAC Address | C's MAC Address | 0x806 | ARP Reply | CRC |

Each node has in its memory an ARP table, called ARP cache, where it temporarily stores the mappings between the IP addresses and MAC addresses it has used, and could reuse again. Each entry in the ARP cache has a time to live (TTL) value associated with it, indicating when it will be removed from the table. ARP is considered a plug-and-play protocol because the ARP cache is built automatically, it does not need to be configured by the system administrator.

When a node needs a correspondence between MAC and IP address, it will first check its ARP table, and in case it does not have the information, it will launch the ARP query to the LAN.

Notice that **in Case 2, when A (111.111.111.110) want to send an IP datagram to B (222.222.222.222),** it will happen the same, but the IP address in the ARP query will be the router IP address:

| | | | | |
|---|---|---|---|---|
| FF:FF:FF:FF:FF:FF | A's MAC Address | 0x806 | ARP Query: Who has 111.111.111.110? | CRC |

| | | | | |
|---|---|---|---|---|
| A's MAC Address | Router's MAC Address | 0x806 | ARP Reply | CRC |

Once A knows the next hop MAC address, that is the destination MAC address, it will encapsulate the IP datagram in a frame and it will be sent to the next hop in the path.

**In the Case 1** the frame that contains the IP datagram destinated to C will be:

| | | | | |
|---|---|---|---|---|
| C's MAC Address | A's MAC Address | 0x806 | IP Datagram from A's IP Ad. to C's IP Ad. | CRC |

**In the Case 2** the frame that contains the IP datagram destinated to B will be:

| | | | | |
|---|---|---|---|---|
| Router's MAC Address | A's MAC Address | 0x806 | IP Datagram from A's IP Ad. to B's IP Ad. | CRC |

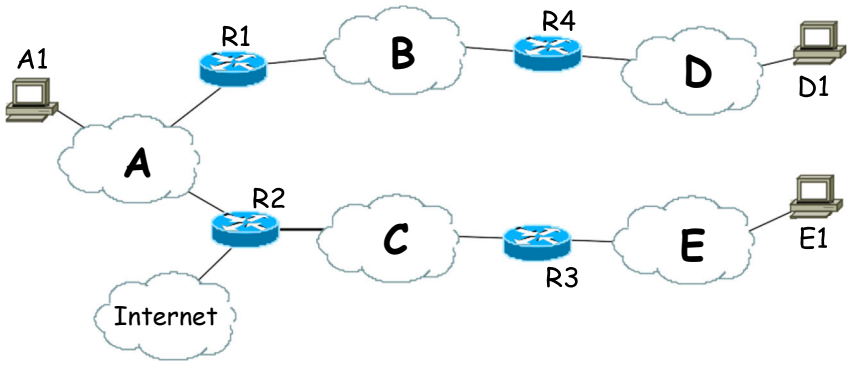
When the frame reaches router R, the router R will discard the frame and will pass up the IP datagram to the network layer. Router R will consult its forwarding table to determine which interface to use to reach the destination. It will be decided to forward the datagram using its 222.222.222.220 interface and will pass the datagram to its link layer. Router's link layer will encapsulate the IP datagram in a new frame:

| B's MAC Address | Router's MAC Address | 0x806 | IP Datagram from A's IP Ad. to B's IP Ad. | CRC |
|---|---|---|---|---|

Notice that IP addresses in IP datagram sent by A keeps the same, but MAC addresses in the new frame generated by the router have changed.

Once the frame reaches node B, the link layer will extract the datagram and pass it up to the network layer, which will process it conveniently.

**Exercise#1**.- The figure shows a set of local Ethernet networks (A, B, C, D and E) of a company connected to each other by four routers (R1, R2, R3 and R4). The network connects to the Internet through router R2. We will use the notation IP (D1) and IP (R4D) to denote the IP addresses of host D1 and Router 4, adapter connected to network D. From the same node, MAC (D1) and MAC (R4D) refer to their MAC addresses respectively. We will assume all nodes are correctly configured and cached DNS resolutions.




a) If we assume that the ARP caches associated with the adapters are initially empty, write down the new entries that will appear in the ARP caches of all the adapters when A1 sends a IP datagram to D1 and D1 replies to A1.

| ARP cache of adapter: | IP Address | MAC Address |
|---|---|---|
| A1 | | |
| | | |
| | | |

b) If after it, E1 sends an IP datagram to D1, how will be the ARP caches left?

| ARP cache of adapter: | IP Address | MAC Address |
|---|---|---|
| E1 | | |
| | | |
| | | |
| | | |
| | | |

# 4. Traffic analysis

Wireshark software will be used for gathering ARP traffic so it can be studied in detail. But first lets have a look at our computer's configuration details.

**Exercise#2.** Using `ip a` in linux or `ipconfig /all` in Windows, find out how many network interface cards are installed in your computer.

1. Write down each card's MAC address and the corresponding IP address too.
2. Find out what type of technology is used for each network adapter.

| Adapter | MAC Address | IP Address |
|---|---|---|
| | | |
| | | |
| | | |
| | | |

For the next exercise the link layer information is where attention will be focused. Please note that ARP is not related to IP protocol at all. Though it is related to IP addresses, the information exchange it is purely based on the exchange of two link layer frames with an specific data format.

**Exercise#3.-** Start a Wireshark capture (http-traffic capture filter this time: "tcp port 80 and host www.upv.es") while you load http://www.upv.es main page on your browser. Select the HTTP request message that includes GET.

1. What type of transport layer addressing is used? (It means, what is the field in the transport header that identifies whom has to receive the data in the payload field?)

2. What type of network layer addressing is used?

3. What type of link layer addressing is used? Whom is the destination MAC address?

4. What does Type field mean (on Ethernet header)? What is its value for captured traffic?

• Note that Wireshark expresses MAC addresses in two formats.

5.What digits identify the manufacturer of the adapter?

The Wireshark analyzer displays, by default, the highest layer information possible. However, it is possible to focus on the lower levels (link and physical layers). To do this, through the option Analyze-> Enabled Protocols, we will disable the IPv4 protocol.

Notice how the appearance of the Wireshark windows has changed, especially the top one. What values appear now in the Source, Destination, and Protocol columns? Re-enable the IPv4 protocol to get a complete overview of TCP/IP.

Our computer keeps a list of previously resolved MAC-to-IP mappings in memory. This way anew ARP is not needed for each IP datagram you want to send to a known destination. The `arp` command allows us to both view and manipulate the contents of our arp cache:

- `arp -a`: allows us to view the content of the local ARP cache.
- `arp -d <IP_dir>`: allows us to manually delete entries from the arp cache. For this, the user must have root permissions.
- `arp -s <IP_dir> <dir_Eth>`: allows us to manually add entries to the cache. We also need root permissions.

In order to get root permissions in Windows, you must run the "Command Prompt" as administrator. In Linux, you must use sudo command as "sudo arp -d <dirIP >"

**Exercise#4.** From a shell window run the command `arp -a` to list the content of the resolution cache. Write down the results. Which are the computers that appear on the list? How are they related with the previous exercise?

1. Now start a new Wireshark capture (with capture filter arp or icmp this time).

2. Now execute ping to another host in your LAN, if you don't have another laptop at home, you can ping to your mobile. Usually you can see your mobile's IP address in *"Ajustes" -> "Acerca del teléfono"->"Estado"*.

`ping` *`mobile's IP address`*

and check again the content of the ARP cache. What is the MAC address for that destination? You can stop the capture now.

3. Find the ARP request and reply within the capture (you can use display filter arp).
4. Select the ARP query message, and answer the following questions:
   - Which layers of the TCP/IP (transport (TCP or UDP?), network (IP?), link (Ethernet?)) protocol stack appear in the ARP query captured? Why? Where is ARP query encapsulated?
   - Evaluate the Ethernet header of the ARP query. Look at the values of the source and destination addresses, as well as the Type field. What identifies this last field?
   - Look at the fields in the ARP query. What information does the ARP query provide to the other nodes in the network? In which field does it indicate the IP address consulted? What is the MAC address of your router?
5. Now select the ARP reply message associated with the query made:
   - Look at the source and destination address of the Ethernet header, and identify whom the source MAC address? Verify that the Type field identifies the ARP protocol again.
   - What IP addresses appear inside the reply ARP message? Whom is that IP address? Where does the information that our computer had requested appear?
   - Which field allows you to differentiate an ARP query from a response?

If we return to the capture made, we can see how immediately after the ARP query, the ICMP messages generated due to the execution of the ping command appear. In other words, once the destination's physical address has been obtained, our machine can send the ICMP echo request message within a frame addressed to it. In this sense, it is similar to the DNS protocol.

**Exercise#5.**- Execute the ping www.upc.es command and check if the IP address of the www.upc.es server has appeared in the cache. Who owns the other address that appears?
As you know, before the execution of the ping command, a query was made to the DNS server to resolve the name www.upc.es. Does the DNS server address appear in the ARP cache? Why? (In Windows you can find out DNS IP address with the command ipconfig / all).

With ARP it is only possible to find out the physical addresses of computers that are on your same IP network.

**Exercise#6.-** Finally, try to find out the physical address of 3 devices that are connected to your network and that are running. To do this, use the ping command and then query the content of the ARP cache. Remember that ARP queries are performed before the execution of the ping, so the ARP result is independent of whether the destination machine answers the ping or not.