Lab 7: ARP (Address Resolution Protocol)

ATTENTION! You must use the user account **redlocal**

Pre-reading: Kurose2022, section 6.4.1

Pre-lab tasks to be completed before the lab session:

- Read the section: **Introduction.**
- Study the sections: **Physical Addresses** and **ARP Protocol.**

1. Introduction

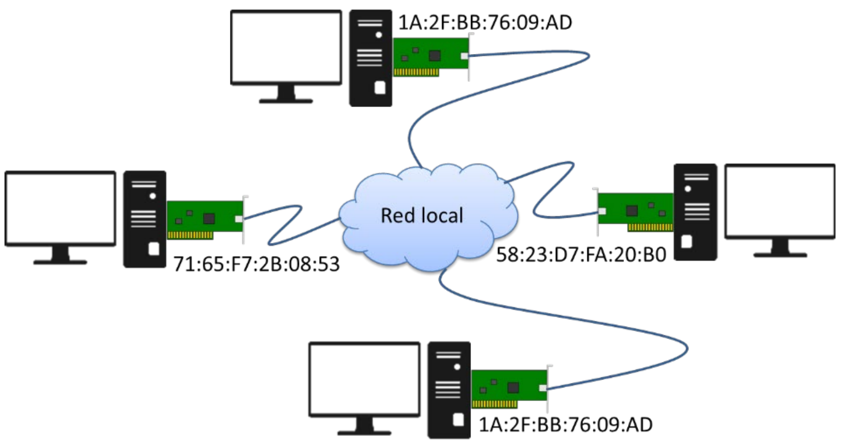
Each host and router connected to the Internet is identified by a network layer address: the IP address. In addition, every network adapter installed on a node has a link layer address: the physical address.

In this lab, we will study link-layer addressing and the Address Resolution Protocol (ARP), which is responsible for translating IP addresses into physical addresses.

2. Physical Addresses

Link layer addresses are assigned to network adapters and are referred to by several names: LAN address, physical address, or MAC address.

IEEE physical addresses are 48 bits long, giving us 2^{48} possible physical addresses. These addresses are typically expressed in hexadecimal notation, with each byte shown as a pair of hexadecimal digits. This is illustrated in the following figure:



The physical address of a network adapter has a flat structure (unlike IP, which is hierarchical) and does not change even if the node moves to a different network.

IEEE manages the physical address space to ensure global uniqueness, regardless of the manufacturer or network. When a company wants to produce network adapters, it must purchase a block of 2^{24} addresses. IEEE assigns the first 24 bits of the physical address block, and the company is responsible for ensuring uniqueness of the remaining 24 bits for each adapter.

When two nodes on the same network want to communicate, they need to know not only each other's IP address but also each other's physical address. **A crucial point is**

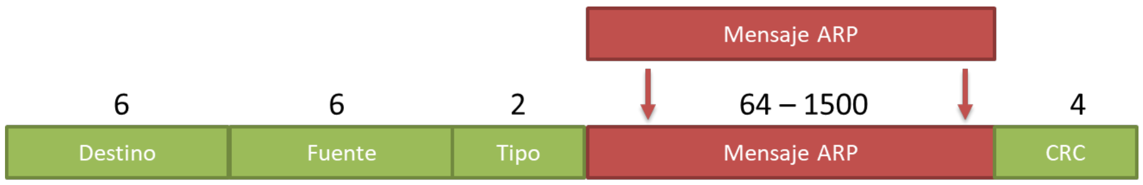
that a computer only needs to resolve another computer's physical address if both belong to the same IP network.

3. ARP Protocol

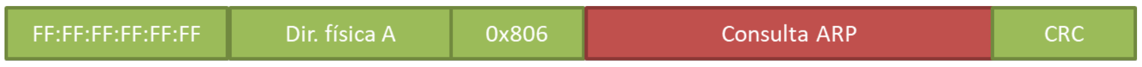
For an IP datagram to travel across a local area network, it must be encapsulated within a frame (Ethernet in our case). That Ethernet frame includes the physical address of the next hop, which may be the final destination (if it is on the same local network), or the router forwarding the packet outward (if the source and destination are on different networks).

TCP/IP uses a protocol to obtain physical addresses from IP addresses on a local area network. This protocol is known as ARP (Address Resolution Protocol), defined in RFC 826.

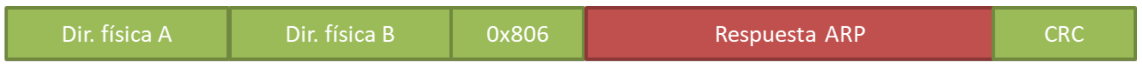
ARP messages operate at the link layer and are encapsulated in the data field of a frame. The **Type** field in the Ethernet header identifies ARP messages: 0x806 indicates an ARP message over Ethernet.



To resolve a physical address, the link layer sends an ARP request containing the sender's IP and MAC address, along with the target IP address. This message is broadcast to all nodes on the network using the broadcast physical address (FF:FF:FF:FF:FF:FF) as the destination. The source MAC address is that of the querying node. In our example, where the source node is "A", we have:



The ARP request reaches all nodes on the network, and each node checks whether the requested IP address matches its own. The one with the matching IP will respond with an ARP reply that includes its MAC address. This reply is sent via unicast directly to the requesting node.

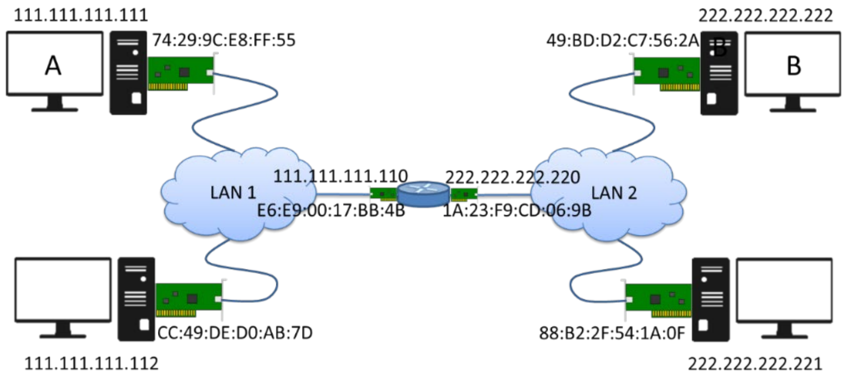


Each node maintains an ARP table in memory that temporarily stores the IP-to-MAC address mappings it has used, and might need again. Each entry has an associated Time-To-Live (TTL) after which it is removed. ARP is considered a plug-and-play protocol because the ARP table is built automatically, and so it does not require system administrator configuration.

When a node needs to map an IP address to a physical address, it first checks its ARP table. If the mapping is not found, it issues an ARP request on the network.

4. Sending an IP Datagram to a Node on Another Network

When two computers on the same network want to exchange an IP datagram, the source resolves the destination's MAC address and encapsulates the IP datagram in a frame addressed to that MAC. However, **if the destination is on another IP network, direct delivery is not possible**. Let's study this situation based on the following figure.

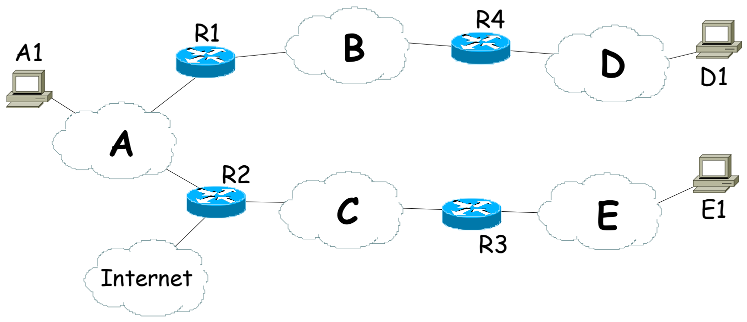


The diagram shows two networks: 111.111.111.0/24 and 222.222.222.0/24, connected via a router. The router has two adapters, each with its own IP and MAC address. Suppose node A wants to send an IP datagram to node B. The sequence of events will be:

1. Node A creates a datagram with source IP 111.111.111.111 and destination IP 222.222.222.222. After checking its routing table, it sees that B is not in its IP network, so it decides to send the datagram to its default gateway: the router.
2. Node A passes the IP datagram to the link layer, specifying that the datagram is to be delivered to IP 111.111.111.110 (the router). The link layer generates a frame with source MAC address equal to A's (74:29:9C:E8:FF:55) and destination MAC address equal to the router's adapter on the same subnet (E6:E9:00:17:BB:4B). To find this destination MAC, A checks its ARP cache or sends an ARP request. Once it has the router's MAC, it creates the frame and sends it.
3. When the frame arrives at the router, the IP datagram is passed to the network layer. The frame is discarded. The router consults its routing table to determine the interface needed to reach the destination, and whether direct delivery is possible. It determines that interface 222.222.222.220 should be used, and passes the datagram to its link layer.
4. The IP datagram is encapsulated in a new frame with source MAC address of the router (1A:23:F9:CD:06:9B) and destination MAC address of node B (49:BD:D2:C7:56:2A). The router may need to check its ARP cache or send a new ARP request to resolve B's MAC address.
5. Once the frame arrives at node B, the link layer extracts the IP datagram and passes it to the network layer for processing.

Exercise 1. The figure shows a set of Ethernet local area networks (A, B, C, D, and E) in a company interconnected 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 refer to the IP addresses of host D1 and Router 4's adapter connected to network D. Similarly, MAC(D1) and MAC(R4D) refer to the corresponding physical addresses. We will assume that all machines are properly configured and DNS resolutions are cached.

a) Assuming that the ARP caches of the adapters are initially empty, indicate how the ARP caches of all adapters will look after A1 sends a message to D1 and D1 replies to A1.
b) Then, if E1 sends a message to D1, how do the ARP caches look?



5. Traffic Analysis

In this section, we will use the protocol analyser Wireshark, which we already know, to analyse the ARP traffic generated on the network. To this end, we need to know a bit more about the machine we are working with.

Exercise 2. Using the command **ip a**, find out how many network adapters the computer you are working on has. Write down the physical address of each of them, and check if they are from the same manufacturer. Do all adapters have assigned IP addresses? What type of IP addresses are displayed?

Adapter	Physical Address	IP Address
lo	00:00:00:00:00:00	172.0.0.1
enp3s0	e8:ea:6a:09:6f:fb	172.0.0.1
eno1	58:11:22:b6:52:e1	--------

In previous lab practices, we focused on analysing information provided by Wireshark captures at the higher layers of the protocol stack: application, transport, and network. In this practice, however, we will study information related to the data link layer.

Exercise 3. Perform a capture with Wireshark while loading the website of the University of Valencia: http://147.156.200.249. Use the capture filter: "`tcp port 80 and host 147.156.200.249`" to isolate HTTP traffic. Select the first HTTP GET request and analyse the protocol headers in the middle pane.

- What type of addressing is used at the transport layer? And at the network layer? Transport: Ports | Network: IPs
- Expand the Ethernet header info. What are the source and destination physical addresses? Source: e8:ea:6a:09:6f | Destination: ac:99:29:95:16:82
- Based on the explanations in previous sections, who do you think owns the destination physical address? The UPV, it's probably a router
- Wireshark shows physical addresses in two formats. Which digits identify the manufacturer of the adapter? What vendor code(s) appear? The first 24 bits (Huawei and StarTech)
- The Ethernet header includes a Type field. What does it identify? What is its value in our case, and what does it represent? IPv4

To focus on lower layers (link and physical), go to **Analyse → Enabled Protocols** in Wireshark and disable IPv4. Observe the change in the main window. What values now appear in the Source, Destination, and Protocol columns? Re-enable IPv4 afterward for full TCP/IP analysis.

ARP maintains a cache of recently resolved IP-to-MAC mappings. The **ip n** command allows us to inspect and manipulate the ARP cache:

- **`ip n`**: displays the local ARP cache.
- **`ip n del <dir_IP> dev <device>`**: deletes a cache entry (requires root).
- **`ip n add <dir_IP> lladdr <dir_Eth> dev <device>`**: adds an entry (requires root).

For example, if the operating system identifies the Ethernet card with the device ID **enp3s0**, then to delete the ARP table entry for the IP address 1.2.3.4, the command would be:

```
ip n del 1.2.3.4 dev enp3s0.
```

Also, it is possible to add an entry to the ARP table:

```
ip n add 1.2.3.4 lladdr 00:11:22:33:44:55 dev enp3s0.
```

Throughout the lab, you may encounter situations where a physical address is already available in the ARP cache, and therefore the ARP messages you want to capture are not generated. In such cases, you can use the command **sudo ip n del <IP_addr> dev <device>** to delete the entry from the table, which forces the system to query the physical address again using ARP when the information is needed. Keep in mind that the first time you run this command, it will prompt you for the password of the **redlocal** user.

Exercise 4. From a terminal, run **ip n** to view the ARP cache contents. Note your results. Which machine(s) do they belong to? How is one of the addresses related to the previous exercise?

Then, perform a Wireshark capture using the filter: "**`(arp or icmp) and host 158.42.180.xx`**", replace xx with your lab workstation number. While capturing, run: **ping –c 1 zoltar.redes.upv.es**. What is zoltar's IP address? 158.42.180.62 Is zoltar on the same IP network as your workstation? Yes, we are in the same network

Check the ARP cache again. What is zoltar's MAC address? 00:16:3e:1d:09:01

In the capture, locate the ARP request and reply related to the ping. Use the display filter **arp**. Analyse the ARP request:

Network / link layer

- What layers of the TCP/IP stack are present in an ARP message? Why? Where are ARP messages encapsulated? In ethernet frames
- Examine the Ethernet header: what are the source and destination MACs? What does the Type field identify? Source: 00:16:3e:1d:09:01 | Destination: e8:ea:6a:09:6f:fb
- Observe the fields of the ARP message. What information does the ARP request provide to the other nodes on the network? In which field is the queried IP address indicated? What address appears as the physical address of zoltar.redes.upv.es?" 1) The physical address of the source node of the request

  00:00:00:00:00:00 (because it is asking for it)

Now select the ARP reply:

- What are the Ethernet source and destination addresses? Who owns the source MAC? Source: 158.42.181.10 Destination: 158:42.180.62 Source mac is of our computer
- What IP and MAC addresses appear in the ARP reply? Where is the info our host needed?
- Which field differentiates a request from a reply?

If we go back to the capture from Exercise 4, we can observe how, immediately after the ARP request, the ICMP messages generated by the execution of the ping command appear. That is, once the physical address of the destination has been obtained, our machine can already send the ICMP echo request message inside a frame addressed to that destination.

Remember that physical address resolution is only performed for destinations within your same IP network. Let's look at this in more detail in the next exercise.

Exercise 5. Run **ping www.upc.es** and check whether the IP address of the server www.upc.es has appeared in the ARP cache. Do you find the IP of www.upc.es in the cache? Why or why not? What is the physical address of www.upc.es?

As you know, before executing the ping command, a query was made to the DNS server to resolve the name www.upc.es. Why doesn't the address of the DNS server appear in the ARP cache?

To finish the lab, let’s see what happens if we insert incorrect information into the ARP table. For example, we can modify the ARP cache so that it includes, as the MAC address of our gateway, an incorrect MAC address.

Exercise 6. Using the commands **ip n del** and **ip n add**, delete the ARP cache entry corresponding to your gateway, and then insert a new entry to indicate that its MAC address is 00:11:22:33:44:55.

IMPORTANT: since the operating system may refresh the ARP cache entry corresponding to your gateway at the same time you're trying to delete it, after using the **ip n del** command, make sure, using the **ip n** command, that your router’s information has actually been removed. Once it has been deleted, you can use the ip n add command.

After you’ve inserted the incorrect information into the ARP cache, verify that everything went as expected using the **ip n** command.

Now that we have “misconfigured” the ARP cache, in Wireshark set the capture filter to (**udp port 53 or arp or icmp**) **and host 158.42.180.xx**, where xx is your station number. Capture the traffic generated when doing a ping www.umu.es (website of the University of Murcia). What happens? How do you explain what’s going on?

To restore the initial configuration and make the ping to www.umu.es work again, what should be done? Is it necessary to use both commands ip n del and ip n add, or would using just one of them be enough? In the latter case, which one should be used?