

Networks & Operating Systems Essentials 2 (NOSE 2)

Lab 1: MAC addresses and the Address Resolution Protocol (ARP)

The aim of this lab is to familiarize you with command-line tools that can be used to examine part of the network state of your computer, related to the Network and Data link layers of the OSI reference model. Specifically, we will look at MAC addresses, the ARP protocol, and tools for viewing/managing the local ARP cache.

However, before you get started on the lab tasks, please make sure that you can/know how to:

1. Use the command line (i.e., know how to open a command prompt window, navigate around in the filesystem, execute a program, etc.),
2. Write Python code in an IDE of your choice (just write a simple Hello World script), and
3. Execute your Python scripts from the command line.

The above might seem mundane, but please don't underestimate them. This year the School switched to using CSCE images for our labs, meaning that programs might not be under the directories where you were used to find them. Also, the credentials used to log on are the University ones (guid, "mail" password). Let your tutor know in the first instance if you have any issues with the above 3 tasks.

MAC Addresses

When a frame is sent from one device to some other device on the same network segment, it must have a unique layer 2 target address. In the case of IEEE 802 network technologies (ethernet, WiFi), as discussed in the lecture, the layer 2 address is also known as the MAC address – a 48-bit number, written out as a string of 6 pairs of hex digits separated by colons or dashes – e.g., “01:23:45:67:89:ab” or “01-23-45-67-89-ab”. Every network interface, including physical (Ethernet, WLAN) and virtual (loopback, local VMs, etc.) interfaces, has its own MAC address.

As a reminder, when the layer 2 software on a device receives a frame, it checks whether the MAC address matches that of a local network interface; if it does, then the frame is processed and forwarded to the upper layers of the network stack, otherwise the frame is discarded. A special case to this rule is frames sent to the *broadcast* MAC address – that is, “ff:ff:ff:ff:ff:ff”; by definition this addresses “matches” all MAC addresses, and as such these frames are always processed. The MAC address of an interface is usually predefined and hardcoded by the manufacturer of the network hardware. You can examine, and in some cases even alter, the MAC address(es) on your device (although changing the MAC address requires superuser privileges and may not always be supported by the hardware/network stack; also, this practically allows one to perform ARP spoofing attacks¹, or to create an ARP proxy²).

¹ https://en.wikipedia.org/wiki/ARP_spoofing

² https://en.wikipedia.org/wiki/Proxy_ARP

Assume a program running on device A, with a layer 2 address of A_{MAC} and a layer 3 address of A_{IP} , wants to communicate with a program running on a remote device B, with a layer 2 address of B_{MAC} and a layer 3 address of B_{IP} . Also assume that these two devices have never communicated with each other in the past. In this case, device A knows its addresses (A_{MAC} , A_{IP}) and the layer 3 address of the remote device (B_{IP}) but not its layer 2 address (B_{MAC}). However, it's this latter piece that is absolutely necessary for the two devices to exchange network frames. How can device A then know or discover the layer 2 address of device B?

Address Resolution Protocol

One could advocate collecting all such addresses in advance and storing on every networked device a file, mapping layer 2 addresses to layer 3 addresses and vice versa. However, maintaining such a static list of mappings of all MAC addresses of all devices on a network to their IP (layer 3) address, is neither practical nor always feasible. For example, in cases where layer 3 addresses are allocated dynamically (e.g., WiFi, mobile phone networks, etc.), it would be impossible to know these mappings beforehand.

Our solution to this problem comes in the form of a protocol, known as the Address Resolution Protocol or ARP. In other words, the ARP is a protocol used to discover the link layer (layer 2) address of a remote device, in association to its network layer (layer 3) address (more on the latter in upcoming labs).

ARP packets contain, among other pieces of information, the sender's and target's layer 2 and layer 3 addresses. For address discovery, the ARP works as follows:

- Device A broadcasts an ARP request to the whole network segment. Specifically, the packet sent has:
 - Sender hardware address: A_{MAC}
 - Sender IP address: A_{IP}
 - Target hardware address: $ff:ff:ff:ff:ff:ff$
 - Target IP address: B_{IP}

and is often printed out in the form:

```
arp who-has  $B_{IP}$  tell  $A_{IP}$ 
```

- As this packet is sent to the broadcast address, it will be received and processed by the layer 2 software of all devices on the network segment.
- The device that has B_{IP} on one of its interfaces, will then send out a response on that same interface. The response packet has:
 - Sender hardware address: B_{MAC}
 - Sender IP address: B_{IP}
 - Target hardware address: A_{MAC}
 - Target hardware address: A_{IP}

and is often printed out in the form:

```
arp reply  $B_{IP}$  is-at  $B_{MAC}$ 
```


To avoid having to do several network roundtrips on every communication attempt, modern devices/operating systems maintain a cache of MAC addresses, also known as the ARP cache. This is a list of ARP responses, each with a timer attached to each entry. Every time the network stack processes a frame, it resets the timer for the entry

corresponding to the packet sender's MAC address. If a timer reaches a predefined threshold, then the entry is deemed stale and removed from the cache.

The ARP also finds other uses. For example, a host can use a so-called ARP probe message to test whether some other host on the network has a given IP address (e.g., to avoid an IP address collision); in this case, in the probe packet the sender hardware address is set as normal, the sender IP address is set to 0, the target hardware address is set to the broadcast address, and the target IP address to the probed address. Also, if a device's MAC or IP address changes, the host can use a so-called ARP announcement (aka gratuitous ARP) to inform all other devices on the network of this update.

Lab Tasks

In this lab, you are asked to perform the following tasks:

1. Open a command prompt window:
 - a. If you are on a **Windows machine**: you can press  + 'r' to bring up the "run command" windows, then type "cmd" (without the double quotes) and hit enter. Alternatively, you can search for the "**Command Prompt**" option through your Start Menu (usually located under Accessories).
 - b. If you are on a **Mac** click the Launchpad icon in the Dock, type **Terminal** in the search field, then click **Terminal**. Or in the Finder, open the /Applications/Utilities folder, then double-click **Terminal**.
 - c. If you are on a **GNU/Linux device** (in most distributions e.g., Ubuntu) you can find it using Ctrl + alt + t , or simply by clicking the terminal GUI on your desktop (if you are on GNU/Linux you should probably know this already!)
2. Run ipconfig command to identify the network interfaces, IP addresses and MAC address(es) on your computer:
 - a. If you are on a **Windows machine**: use the "ipconfig". Type "ipconfig /?" to get a list of options for this command. Note the interface name, IP address and MAC address of all connected interfaces. Why are there so many when there is only one network cable coming out of the computer (or there is no cable)? Discuss with your classmates/tutor. Do you identify differences between a wired with a wireless interface?
 - b. If you are on a **Mac/GNU-Linux machine**: use the "ifconfig" command and check available options by typing "man ifconfig". In some GNU/Linux distributions you may also use "ip a" or "ip addr". For more options run "man ip".
3. Use the "arp -a" command to examine the ARP cache on your computer. Comment on what you see. Are there any peculiarities/irregularities? Which are the devices you can see in your local network? Look them up on the web and discuss your findings with your classmates/tutors.
 - a. If you are on Mac/Gnu-Linux machine and want to have a leaner view of your interfaces try "networksetup -listallhardwareports" on Mac (and thank this useful BSD-based tool <https://www.manpagez.com/man/8/networksetup/>). You can also check

for more options through “man networksetup” (e.g., checking interfaces, network services offered etc.).

4. As mentioned, the MAC addresses are usually predefined and hardcoded by the manufacturer of the network hardware. To avoid collisions, each manufacturer is assigned a 24-bit prefix (i.e., 6 hex digits); that is, all network interfaces produced by a given manufacturer, will have MAC addresses sharing the same first 6 hex digits. The IEEE maintains a list of these prefixes (also known as Organizationally Unique Identifiers or OUIs) at <http://standards-oui.ieee.org/oui.txt>. Look up the MAC addresses of your computer; what do you see? Do the same for any peculiar/irregular entries you possibly identified in the previous task. **Keep in mind that many manufacturers expand the range of their MAC addresses (e.g., Apple devices) and you might not find your address. An alternative lookup could also be done in <https://www.whatsmyip.org/mac-address-lookup/>.**
5. How could you find the MAC address of another computer? That is, how could you force the ARP cache to have an entry for a computer, if you know its IP address? Try this with some other computer in your house.
6. (This would work only in the labs). The IP address of the web server of the School of Computing Science is 130.209.240.1. What is the matching MAC address? Discuss how you found it or possible reasons if you failed to do so with your classmates/tutor.
7. (This would work only in the labs) The IP address of the web server of the University is 130.209.16.90. What is the matching MAC address? Discuss how you found it or possible reasons if you failed to do so.

What to Submit

Nothing... 😊 You should discuss your questions and answers with your classmates and tutor at the lab; this will allow you to get direct feedback and may allow you to explore various options on the spot. Sample answers to the lab tasks adapted on my local PC configuration will be posted on Moodle at the end of the week. You should then review them and relate them to your answers and discuss with your classmates/tutor in the next lab.