

CSIS0234B Computer and Network Communications (Class B)

Tutorial 5

Understanding Ethernet address translation

In this tutorial, we see the steps that the computer performs to find the hardware address of a computer in the same network, using the frame capturing program `ethereal`. Work in groups of 3–4 and share 3 computers, all login as `root` at the initial login prompt so that you have the right to configure the network and capture frames.

1. Inspecting and changing the network configuration of a host

In the tutorial you will need to perform the following tasks that involve configuring the network and inspecting the configuration. Try them now so as to get familiar with your tools.

- a. `ifdown eth0` and `ifup eth0`: The Redhat way to stop and start the network. The `eth0` is called the **interface**, which refers to a particular network card. These commands are maintained by the Redhat distribution, to invoke various commands to start and stop the network, and also do other fancy things. You can also use the graphical interface from the **System Settings -> Network** menu to change the network parameters like the IP address.
- b. `ifconfig eth0`: Get information about the network interface. Among the things that you can find are the IP address (in form of a dotted quad like `123.45.67.89`), the hardware address (in form of 12 hexadecimal digits like `12:34:56:78:9a:bc`), some flags, and the number of frames and bytes being sent and received with that interface.
- c. `ifconfig eth0 down` and `ifconfig eth0 IP-addr`: The low-level interface to stop and start the interface. In principle you will then use the `route` command to configure the routing table, but since we haven't fully understood the routing yet, we will delay its discussion until a bit later. Until then, we will use the IP address you find in step (b) temporarily, and after testing we will restart the network using the Redhat tools.
- d. `ping host`: send "ICMP" request to *host* and wait for reply from the OS of *host*. The *host* can be a domain name or an IP address. This is usually used to test network connectivity. However, this is useful only if there is no firewall in the way which drops ICMP packets.

2. Familiarizing with ethereal

- a. Run "ethereal". In the "Capture" menu, select "Start". Besides the button labeled "**Filter:**", there is a box for entering a capture filter. Make sure it is empty, press "OK" to capture everything for 5 seconds, and press "Stop". It shows 3 "views": a **summary line** for each frame at the top, a **protocol tree view** showing a decoded frame in the middle, a **byte view** showing the bytes in a frame at the bottom. Click at different positions, including the expand buttons of the protocol tree view (on the left) and the hexadecimal numbers in the byte view, to see how `ethereal` assists you in understanding the frames.
- b. Most frames you captured in (a) are multicasts. Make a capture filter so that multicast frames are not captured. See if you can capture `ping` packets that are (1) going to and from your computer, and (2) being transmitted between the two other computers. If the answer to (2) is yes, it means the computers are connected by hubs rather than switches. If so, try again with **promiscuous mode** disabled when you start capturing.

- c. Now clear the filter that you have made, and capture for a few seconds again. What is the size of the smallest frame can you capture?
- d. Look at one such small frame. Note that there is a field called *Trailer* in the Ethernet layer. Compare the frame with the Ethernet frame layout as shown at the end of lecture 4, to find why it is smaller than the minimum Ethernet frame size of 64 bytes.

3. Address Resolution Protocol

- a. Restart the network with `ifdown` and `ifup` for all the 3 computers. Start capturing frames from any of them (using `ether src host MAC1` or `MAC2` or `MAC3`) for all computers. From one of the computers, `ping` one other for a couple of seconds **using its IP address**. Explain all frames generated. Also, note which frames are broadcasts. Try it again a few seconds later, and explain the difference given that there is an ARP **cache**. (We restart the network primarily to make sure the cache is empty.) Look at the cache with `arp`.
- b. Now restart the capturing, this time enable **update list of packets in real time** and **automatic scrolling in live capture**, and capture for two minute. See how long is the ARP cache valid (so that an ARP request is sent again).
- c. Restart the network, and perform the capturing again. This time `ping` using the host name rather than IP address. Explain each frame generated. (There is a `-n` option in `ping`, which might enlighten you about some of the frames generated. See the man page of `ping` to see the description of this option.)

4. ARP on Network configuration

In the following instructions we call the three computers A, B and C respectively.

- a. To see what will happen when two computers in the network use the same IP address, stop the network on A, and start it **using the low-level `ifconfig` tool** to use the IP address of B. Now start capturing frames as above from all three computers, and let A and B ping C's IP address. Continue capturing for a few minutes to see the abnormal behaviour.
- b. The Redhat network configuration tool uses ARP to avoid the problem. Start capturing from B, stop the network at A using the low-level `ifconfig` tool, and use the Redhat network configuration tool to configure the network to use the IP address of B. Now read the captured frames to see how it avoids the problem.
- c. Correct the IP address of A and do the above again. How it makes sure that an IP address clash does not occur? And what ARP packets are generated afterwards?
- d. Restart the network of A again, and look at the ARP cache at C immediately afterwards. Does it cache the MAC address of A now? Give one possible reason for the decision.
- e. Stop the network at A again, start capturing at B, and let B ping A (without success) for a minute. Try this again with one more `ping` process running. Does every ping attempt generate an ARP request? See how this is done by looking at the ARP cache at B.
- f. Start the network at A again and read the ARP cache now. What is the use of the ARP replies you found in step (c)?