CS224 (m): Computer Networks (minor)
Tutorial 09, 28/30 Sep 2016

Concepts tested: ARP, ICMP, NAT, IPv6

1. Find out the IP address of your machine as well as that of your friend. For this, you can use the ifconfig command in linux, or ipconfig command in windows. Now, just by observing a wireshark trace on your machine, and without running any traffic between the two machines, can you find out the hardware (MAC) address of your friend's machine? You may need your friend's machine to generate some traffic, e.g. some browsing. And of course, both of them need to be on the same (extended) LAN. Does what you find matches what you observe when you type the ifconfig or ipconfig command?

2. In the same exercise as above, pay further attention to the various ARP queries which you can see on the LAN. What are the source and destination IP addresses of these? What are the source and destination MAC addresses?

3. Use the "arp" command to view the OS's ARP table (the cache of ARP mappings). You may have to use the "-n" option to prevent reverse name lookup. The exact format of the "arp" command depends on whether you are running linux or cygwin. In cygwin, just typing "arp" gives the various modes of usage. In linux, type "man arp" to learn how to use it.

4. For this exercise, you need to run as superuser or administrator. Start a cygwin terminal as administrator in windows; in linux, you can just precede each command with a "sudo". Try deleting some entries from the OS's ARP table. After deleting, observe how the ARP table gets filled up again, by using wireshark and generating some traffic, e.g. browsing or ping.

5. Observe "ping" traffic on wireshark, by pinging your friend's machine.

6. Now "ping" a non-existent IP address. What ICMP message do you observe? From which machine?

7. **[15 HP]** Setup your laptop as a NAT router for your friend's machine. You can search google on how to do this (I've set this up only on linux, but it should be straightforward on windows too). DO NOT run a DHCP server; instead just use static IP addresses for the NAT interface on the NAT router, as well as the friend's machine. Observe the NAT traffic using wireshark, first on your friend's machine, then on the NAT router machine. Note: you can assign more than one IP address for an interface in Linux, by using "eth:0", "eth0:1", etc. In such a case, you may just need one interface to setup the NATted system.

8. Assuming there are no options to be carried as part of the IP datagram, which version of the IP protocol provides more payload capacity (to the higher layer)?

9. Earlier in IPv4, if you want the source to fragment the datagram (not routers), then source can discover the MTU size along the path by sending a datagram (with MTU size equal to the network it is connected to) with DF fragment set. At the router where the MTU size is exceeded, the router will send back an ICMP error message conveying the right MTU size. The source then uses the new indicated size and this process repeats till the lowest MTU along the path is found.

   In IPv6, how can a source discover the correct MTU size along a path? Note that unlike in IPv4, there is no 'DF' flag as part of IPv6 since source itself does the fragmentation. Routers don't even look at the fragmentation extension header carried as part of the datagram.