CSCD58: Computer Networks **April 6, 2014**

Assignment 3

By: Pirave Eahalaivan *998152136 (eahalaiv)*

Question 1 : Poking Around the UofT Network

The following commands were used to answer questions 1.1 - 1.5.

```
eahalaiv@mathlab:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:84:73:98
          inet addr:142.1.96.164  Bcast:142.1.96.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2214116469 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2300178299 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1109160854898 (1.1 TB)  TX bytes:1214182444290 (1.2 TB)
eahalaiv@mathlab:~$ traceroute google.ca
traceroute to google.ca (74.125.226.151), 30 hops max, 60 byte packets
traceroute to google.ca (74.125.226.151), 30 hops max, 60 byte packets
 1  utsc-srv.gw.utsc.utoronto.ca (142.1.96.1)  0.495 ms  0.568 ms  0.612 ms
 2  mcl-utsc-gpb.gw.utoronto.ca (128.100.96.113)  10.210 ms  10.247 ms  10.277 ms
 3  lupus-gpb.gw.utoronto.ca (128.100.96.16)  1.806 ms  1.740 ms  1.677 ms
 4  ut-hub-utoronto1-if-re.gtanet.ca (205.211.94.233)  2.007 ms  2.287 ms  2.219 ms
 5  ORION-GTANET-RNE.DIST1-TORO.IP.orion.on.ca (66.97.23.57)  1.989 ms  2.030 ms  1.963 ms
 6  be201.gw01-toro.orion.on.ca (66.97.16.22)  1.849 ms  1.741 ms  1.762 ms
 7  74.125.48.230 (74.125.48.230)  1.690 ms  1.625 ms  1.608 ms
 8  209.85.255.232 (209.85.255.232)  1.611 ms  1.610 ms  1.600 ms
 9  209.85.250.7 (209.85.250.7)  2.253 ms  2.338 ms  2.429 ms
10  yyz08s14-in-f23.1e100.net (74.125.226.151)  1.666 ms  1.633 ms  1.694 ms
eahalaiv@mathlab:~$ arp 142.1.96.1
Address                  HWtype  HWaddress           Flags Mask            Iface
utsc-srv.gw.utsc.utoron  ether   40:55:39:27:da:c1   C                     eth0
eahalaiv@mathlab:~$ arp 205.211.94.129
205.211.94.129 (205.211.94.129) -- no entry
```

1. What are the IPv4/IPv6 addresses of the mathlab server?

   From the `ifconfig` command we can see that `mathlab` does not have an IPv6 address (`inet6` is not present). The IPv4 address is 142.1.96.164.

2. What is Mathlab's MAC address?

   From the `ifconfig` command we can see that the `HWaddr` (i.e. MAC address) is 00:50:56:84:73:98.

3. What are the IPv4 and MAC addresses of the router used by math lab to reach other subnets?

   From the `tracroute` command we can see that 142.1.96.1 is the IPv4 address of the router (`utsc-srv.gw.utsc.utoronto.ca`) used by `mathlab` to reach other subnets. To find it's MAC address 40:55:39:27:da:c1, the command `arp 142.1.96.1` was used.

4. What are the IPv4 and MAC addresses of the router used by U of T to reach the rest of the Internet (the router running BGP with external ASs)?

   From the `tracroute` command we can see that 128.100.96.16 is the IPv4 address of the router (`lupus-gpb.gw.utoronto.ca`) used by UofT to reach the rest of the Internet. The `arp 205.211.94.129` command does not produce any results for the MAC address of `lupus-gpb` because it is not part of our LAN.

5. How many IPv6 host addresses are currently allocated to the U of T AS?

   U of T has registered AS number 239, and it hosts one subnet `2606:fa00::`. The prefix `2606:FA00::/32` implies that $2^{128-32} = 2^{96}$ IPv6 host addresses are allocated for U of T AS239 (see `http://www.tcpiputils.com/browse/ipv6-address/2606:fa00::`).

Question 2 : Explain it to me

Gratuitous ARP

A gratuitous ARP request or reply is one that is not usually needed as per the regular ARP specifications (yet sometimes needed). A gratuitous ARP request is made from and sent to the same IP address, and the destination MAC address is `ff:ff:ff:ff:ff:ff` (no reply is made). A gratuitous ARP reply is one for which no request was issued.

For example, say node A and B share the same IP address 111.111.1.1 and have MAC addresses 01:01:01:01:01:01 and 05:05:05:05:05:05 respectively. Suppose the current active node A dies out, and node B is brought in to take it's place. Node B will now broadcast a gratuitous ARP reply so that all systems receiving this reply know to update their ARP table to map IP address 1 to the node B's MAC address.[1]

ARP Cache Poisoning

An ARP Cache Poisoning occurs when an attacker broadcasts and ARP reply, indicating that a specific IP address is associated with any other (false) MAC address. Now everyone on the LAN will update their ARP tables to map this IP address to the wrong MAC address and trust it is the right one.

An example of how ARP Cache Poising can affect a network is via denial of service. Say a hacker decided to deny everyone access to printing. All they need to do is send an ARP reply with the printer's IP address and a bogus MAC address.[2]

IPv6 tunnelling

RFC 4213 discusses two ways to use current IPv4 capable systems with the new IPv6 capable systems, and one such way is IPv6 tunnelling. In IPv6 tunnelling, IPv6 datagrams can be sent without loss of information over both IPv4 and IPv6 systems. The basic idea is to encapsulate the IPv6 datagram inside an IPv4 datagram's data/payload field at the start of the tunnel (consisting of IPv4 systems), and set the destination to the end of the tunnel.

In the diagram below, say node A has to send an IPv6 datagram to node F. Once the datagram reaches node B, it will be encapsulate it in an IPv4 datagram with source B and destination E before being passed through the tunnel (IPv4 nodes C & D). At node E the IPv6 datagram is extracted from the IPv4 one and the original IPv6 datagram is sent along to F, intact.

Token Ring

A token ring is a LAN that is setup in a circular/ring formation and the token-passing protocol (type of taking-turns protocol) is used to circulate a token (a three byte frame) around the network. This token ensures that only one node is active at a time and is the only node that is currently transmitting data. (It was an alternative to Ethernet)

Initially the token is set to 0 and is passed around the ring, once a node has a message to send, it'll switch the token to 1 and add the destination and message. The message travels along the ring until the destination, where it will be read and token is switched back 0 and passed down again. When the original sender receives the message back again, it will see the token is 0 and will empty out the message and destination.[3]

[1] `http://wiki.wireshark.org/Gratuitous_ARP`
[2] `http://www.watchguard.com/infocenter/editorial/135324.asp`
[3] `http://en.wikipedia.org/wiki/Token_ring`

Question 3 : Protocols