**Lab 4 Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**MCSE 1 Frame Addressing and ARP**

In this lab you will:

- Investigate how frame addressing works

- Learn how ARP works

- Use Wireshark to observe the 3 levels of addressing

**Procedure:**

\_\_\_ Start Ottawa and Hamilton. Log into the Administrator’s account on each computer. (We won’t use Halifax in this lab).

\_\_\_ Set Ottawa to 10.1.1.2/16 and Hamilton to 10.1.1.3/16. No preferred DNS server and no gateway.

\_\_\_ Make sure you can ping between Ottawa and Hamilton; both ways.

**Hamilton:**

\_\_\_ Obtain a 64-bit version of Wireshark from the instructor. Copy the Wireshark folder from the instructor’s flash drive to the root of C:.

\_\_\_ Run the Wireshark program from the Wireshark folder on Hamilton. Use the default settings during the installation.

\_\_\_ There is a compatibility issue with the WinPcap program. It is designed for Windows 7 and not Windows 8. Continue with the installation by clicking on the **Run the program without getting help** option.

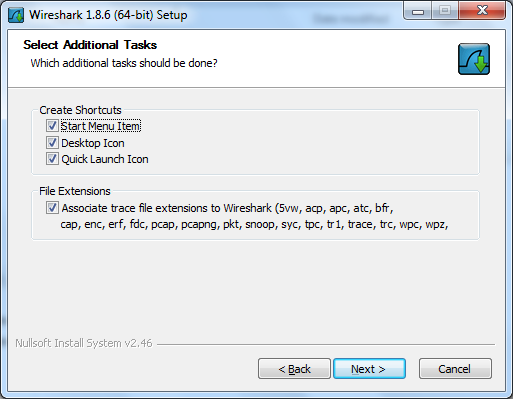
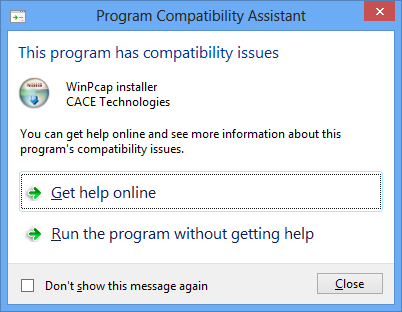


Fig. 1 Choose to place an icon on the Desktop

\_\_\_ When the installation is finished, press the Windows key to return to the **Start** window. There should be a new tile for Wireshark on the right side of the desktop.

\_\_\_ Click on the Wireshark tile to start the program. Click **Capture** on the main menu bar at the top. Select **Interfaces**. Figure 2 should be visible.

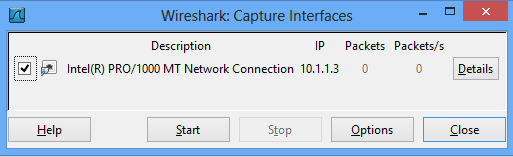


Fig. 2 Specify which interface to monitor.

\_\_\_ You must tell Wireshark which interface to monitor. Select the box opposite the interface on address 10.1.1.3. Click on **Start**.

\_\_\_ Make the selection shown in figure 3.

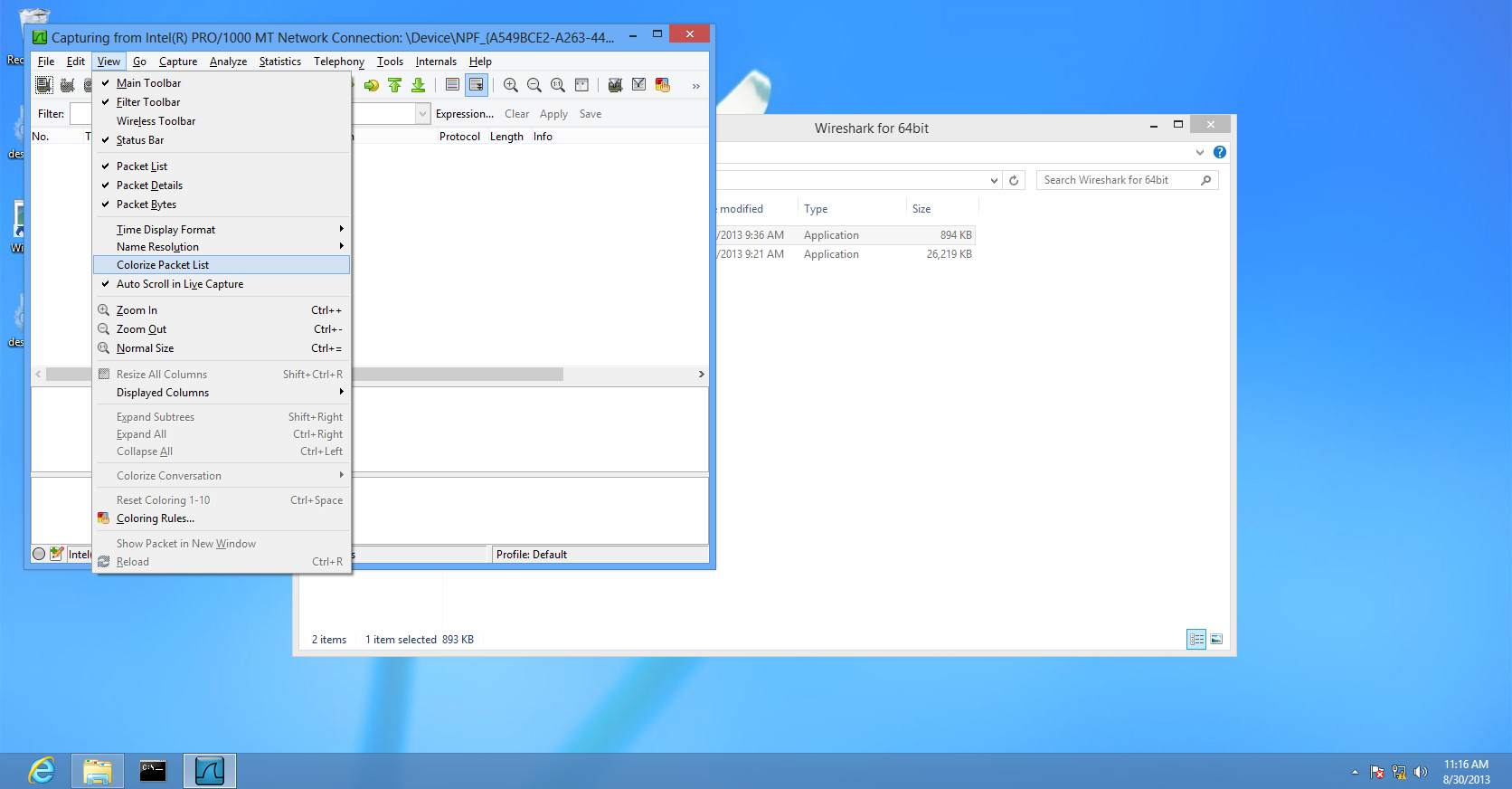
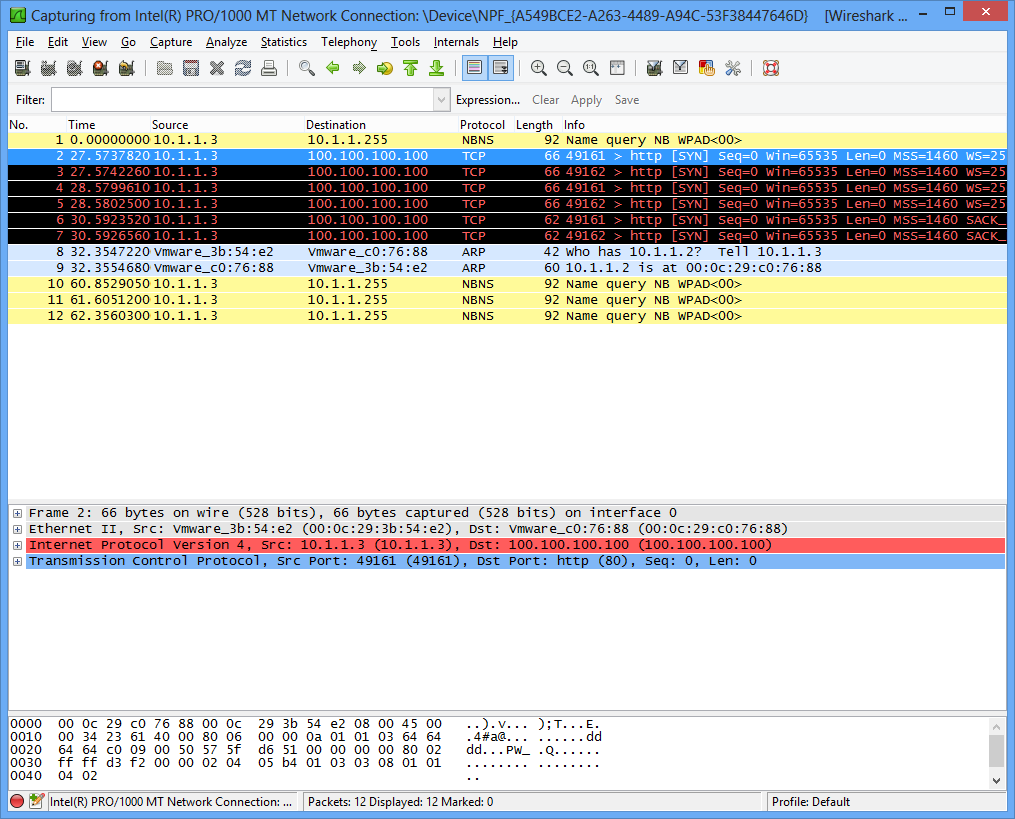
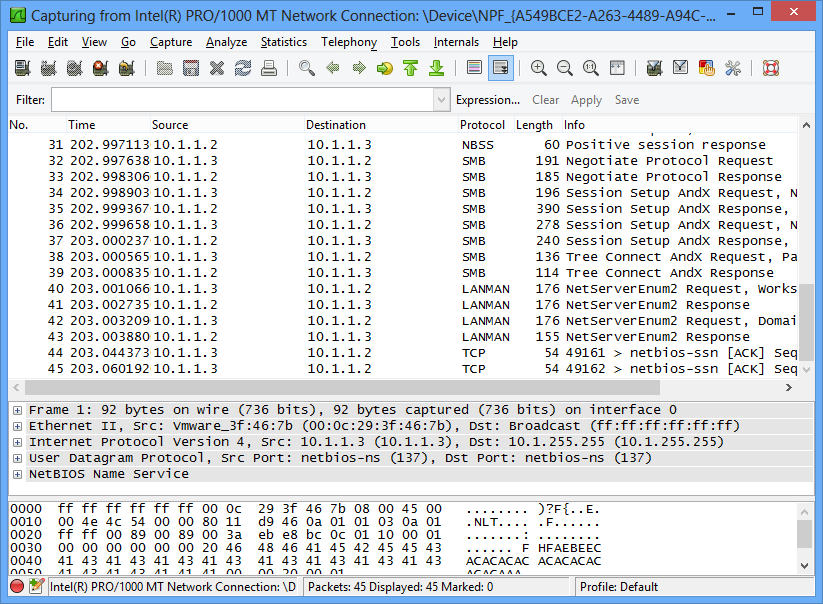


Fig. 3 Uncheck the “**Colorize Packet List**” option

The last selection will ensure that the Wireshark display is all white instead of coloring certain types of packets a different color. Some of the color schemes that appear are very difficult to read. By making them all white, they are easy to read. See figure 4.



**Colored packets are hard to read**



**White packets are easy to read**

Fig. 4 Difference when packets are not colored

**Ottawa:**

\_\_\_ Type **ping 10.1.1.3**.This is Hamilton’s IP address.

**Hamilton:**

\_\_\_ When the pings are finished, click on **Capture** and **Stop**,in Wireshark.

This will stop the capture process.

Figure 5 should be visible. You should have four ARP packets and a number of ICMP packets. Each ping is made up of a request and a reply packet.

If there are no ARP packets, type **arp –d \*** at the command prompt on Ottawa and Hamilton. Then issue the ping command again. This time you should have ARP packets.

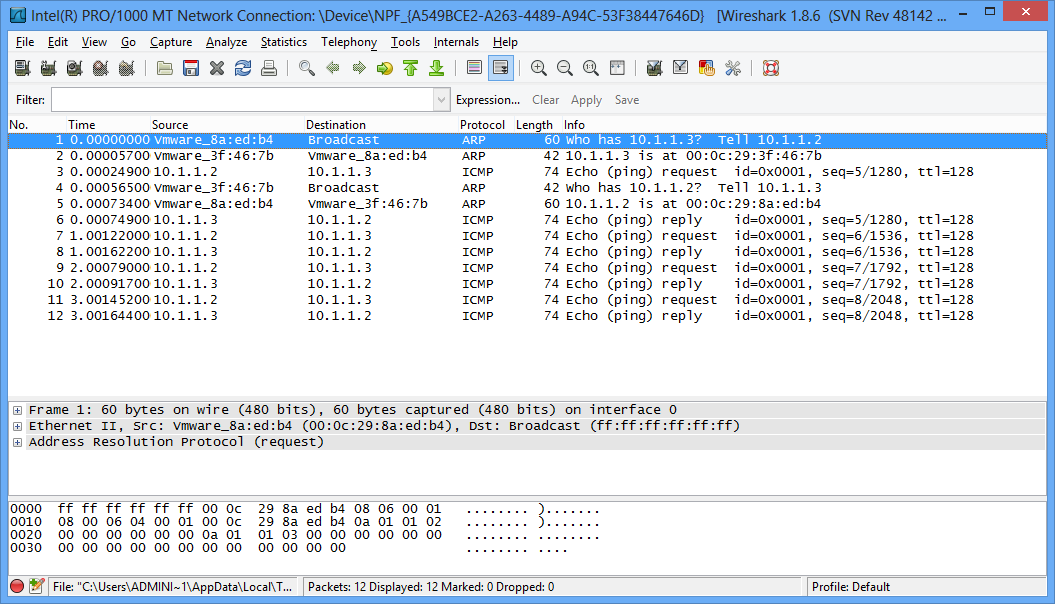


Fig. 5 Wireshark has captured the pings.

**1. Capture the view of Wireshark showing the 4 ARP packets and the ICMP packets.**

Here’s an explanation of what all the packets are used for:

When a host tries to reach another host on the same network the Ethernet frame must contain the source host IP and MAC addresses as well as the destination host’s IP and MAC addresses.

You can see in figure 5, the first two frames are ARP packets. When Ottawa pinged Hamilton, Ottawa had the IP address of Hamilton as evidenced by the ping command you typed on Ottawa:

**ping 10.1.1.3**

The ping cannot be delivered until Ottawa also obtains Hamilton’s MAC address. To get Hamilton`s MAC address, Ottawa sends an ARP request. This is a broadcast as seen in figure 5.

**2. Look at the “info” column in figure 5. Explain what the purpose of the 1st frame is in your own words. Don’t use the IP addresses in your explanation. (In other words, why was this frame sent?)**

The second frame in figure 5, is the ARP reply from Hamilton to Ottawa. This frame is telling Ottawa what Hamilton’s MAC address is so Ottawa can deliver the ping to Hamilton.

The addressing on the first frame was done at the Data Link layer (layer 2), using MAC addresses. Ottawa’s source MAC is **Vmware\_8a:ed:b4**.

MAC addresses consist of 12 hexadecimal digits. We only see the last 6 digits in Wireshark. The reason is that the first 6 digits have been assigned to VMWare so Wireshark decided it was more user friendly to replace the first 6 digits of the MAC address with the company’s name that owns this block of MAC addresses.

Notice the destination address in the 1st frame is a broadcast. In Hexadecimal format this would be FF-FF-FF-FF-FF-FF. In binary format it is all “1’s”.

The 2nd frame is Hamilton replying to Ottawa with its MAC address.

**3. Use the information in the 1st and 2nd frames to figure out what the first 6 hexadecimal digits are for “Vmware”. Record the first 6 digits in your answer file. Use the values in figure 5 not the values in your Wireshark display. This will result in everyone having the same answer so I can mark it.**

\_\_\_ Notice in frame 2, the frame was delivered using the source and destination MAC address of Hamilton and Ottawa, respectively.

As you learned in the lecture, most frames contain 3 levels of addressing:

* Layer 4 port numbers
* Layer 3 IP addresses
* Layer 2 Ethernet MAC addresses

ARP frames are generated at the Data Link layer, layer 2. Therefore, they do not contain layer 4 (port numbers) addressing. They do contain the layer 3, IP address, and the layer 2, MAC address.

The wireshark display contains 3 sections. Observe the middle section for the next step.

\_\_\_ Expand the **Address Resolution Protocol (request)** portion of the first ARP packet.

**4. Capture the expanded view of the “Address Resolution Protocol (request)” portion of the 1st ARP packet. Make sure the Sender and Target IP and the MAC addresses are visible in your capture.**

\_\_\_ On Hamilton, at the command prompt, display the MAC table, also called the ARP cache. To do this type **arp –a**.

**5. Capture the contents of the arp cache.**

In figure 5, once Ottawa received the MAC address from Hamilton (in frame 2), it sent the first ping request. Then two more ARP’s appear in figure 5.

With XP machines, the entries in the MAC table are cleared out after a few minutes. With later versions of the client operating systems, the entries remain until you clear the arp table or turn off the computer.

**6. Why is it important for a computer to store the IP to MAC mappings of other computers?**

Notice there is another set of ARP frames; frames 4 and 5 in the figure 5.

**7. Explain in your own words what the purpose of frames 4 and 5 are.**

Once the MAC addresses are known, the pings continue uninterrupted.

**Hamilton:**

\_\_\_ Start a new capture in Wireshark. (Capture > Start > Continue without saving).

**Ottawa:**

\_\_\_ Ping Hamilton’s IP address.

**Hamilton:**

\_\_\_ Stop the Wireshark capture.

Notice the ARP frames appear even though the MAC table still contains the IP to MAC mappings. This seems counter-productive. Why ask for the MAC address when it is in the MAC table.

With XP, if the MAC address appeared in the MAC table an ARP was not sent. Which seems like a really good idea because it saves bandwidth and CPU cycles.

WI-FI connections using laptops and hand held devices is extremely common today. These devices obtain different IP addresses as they drift from network to network.

One wireless device that has the IP address of 10.1.1.45 may move on to another wireless network. Within 40 seconds a second host might join the network and be given the IP address 10.1.1.45. If a 3rd host was in communications with first host, it will have the wrong MAC address in its ARP table. When it tries to communicate with 10.1.1.45 it will be using the MAC address of the host that has moved to another network instead of the new host that has 10.1.1.45.

If the entries are older than 30 seconds, the new operating systems starting with Vista, double check the entries in their MAC table by sending an ARP to verify the entries.

**Hamilton:**

\_\_\_ Start a new capture in Wireshark.

**Ottawa:**

\_\_\_ Ping Hamilton’s IP address.

\_\_\_ Ping it again, right away.

**Hamilton:**

\_\_\_ Stop the Wireshark capture.

Notice the second set of pings did not generate an ARP request because the MAC table entries were considered still fresh enough to use.

**8. Capture the wireshark display showing the second set of pings did not generate ARP requests.**

\_\_\_ Start a new capture in Wireshark on Hamilton.

\_\_\_ On Hamilton, open the Internet Explorer and type the IP address 100.100.100.100 in the URL field and click on the GO button.

Notice there is no output generated in Wireshark. Why is that? (You may have spurious output unrelated to connecting to the 100.100.100.100 address. Just ignore that).

The reason is Hamilton wants to reach 100.100.100.100 which is not on the same network. The host applies the subnet mask (255.255.0.0), and finds the target is on network 100.100.0.0. The host applies the subnet mask to its own IP address and gets a network of 10.1.0.0. When the host compares the two networks it becomes evident they are not on the same network. Now the host must send the packet to the Gateway. We don’t have a Gateway configured on Hamilton so the host just throws the packet away and no frames are placed on the network for Wireshark to capture.

\_\_\_ Open the properties page of the NIC on Hamilton. Configure the Gateway

address of 10.1.1.2. We’re using Ottawa as our Gateway so we can get a

response.

\_\_\_ Start a fresh capture in Wireshark.

\_\_\_ In the browser, try to reach 100.100.100.100 again.

\_\_\_ Stop the capture when you get an output similar to figure 6.

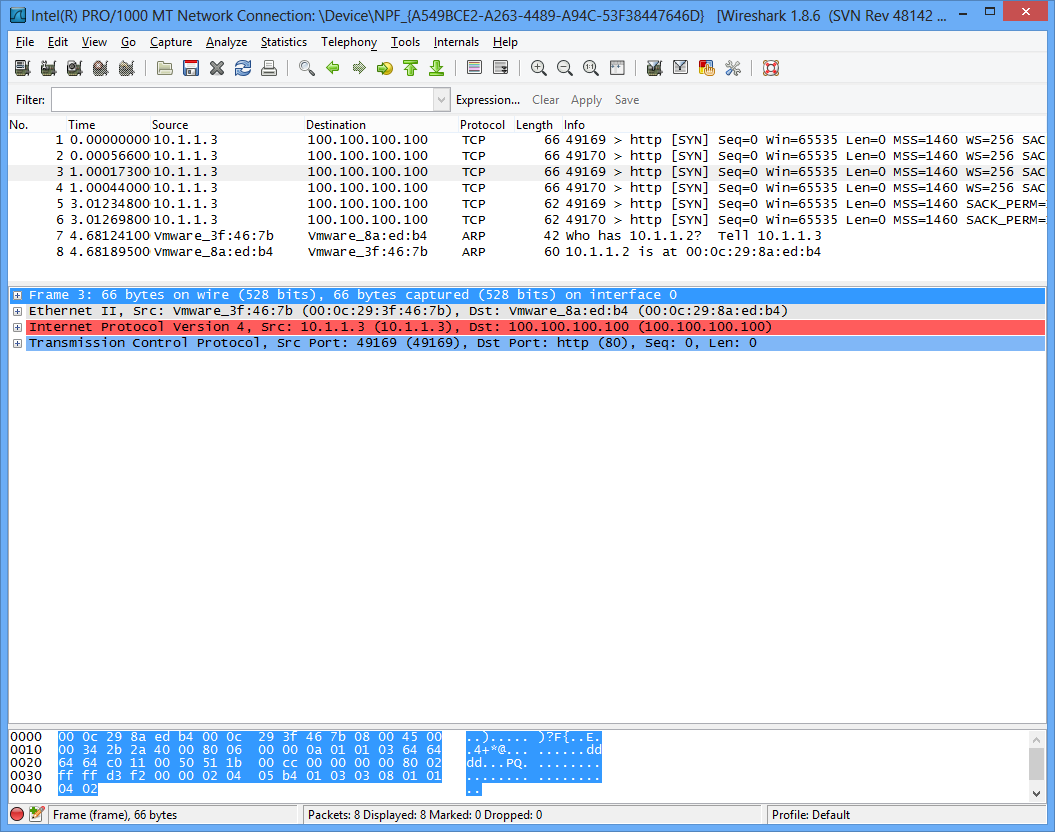
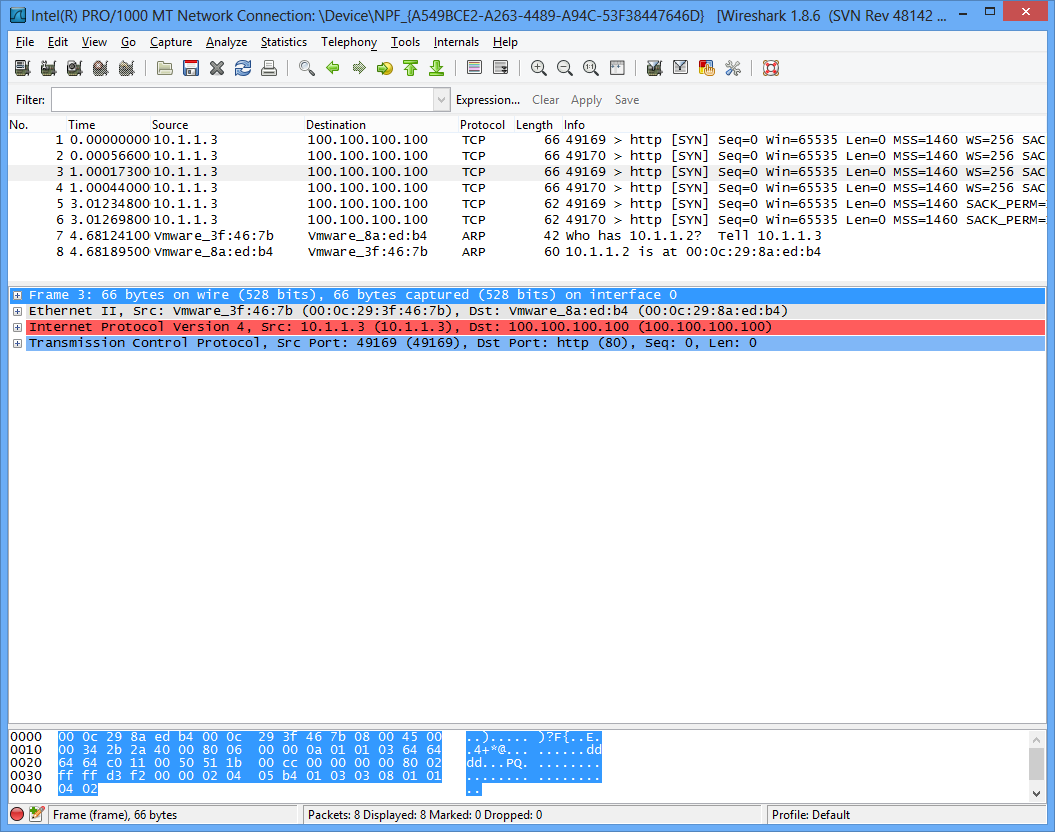


Fig. 6 Hamilton (10.1.1.3) is trying to send frames to 100.100.100.100

In figure 6, it looks like Hamilton is sending the frames directly to 100.100.100.100. Actually, Hamilton is sending the frames to the Gateway, Ottawa.

Figure 7 shows the MAC address that the frames are being sent to is actually Ottawa. Remember from the lecture, the source and destination IP do not change as the packet travels from source to destination. Only the layer 2, MAC address changes.



Source and Destination ports

Source and Destination IP

Source and Destination MAC

Fig. 7 The ping is passed to the gateway

Now when we try to browse to 100.100.100.100 our host realizes it must send the frame to the Gateway and we obtain a TCP packet in Wireshark showing this.

Since our host does not get a response from the WEB server at 100.100.100.100 because it does not exist, our host tries 4 more times resulting in 5 attempts to reach the WEB server. Then it gives up.

You learned that each frame must have 3 different pairs of addresses:

1. a pair of source and destination MAC addresses for layer 2

2. a pair of source and destination IP addresses for layer 3

3. a pair of source and destination port numbers for layer 4

In figure 7, you can see the destination and source MAC addresses, source and destination IP addresses, and the source and destination port numbers.

**9. In figure 7, what is the destination port number?**

**10. Why is that port number chosen?**

**11. In figure 7, what is the source port number?**

**12. How was this port number chosen?**

\_\_\_ Start a new capture in Wireshark.

\_\_\_ On Hamilton, in the URL of the Internet Explorer, type: **ftp://100.100.100.100**

\_\_\_ Wait a few seconds and then stop the Wireshark capture.

\_\_\_ Highlight one of the frames that contains the letter “**ftp**”. View the addresses in the lower half of the display.

**13. What is the destination port number for ftp?**

**14. What is the source port number?**

\_\_\_ Start a new Wireshark capture.

\_\_\_ In the Internet Explorer, type [**ftp://100.100.100.100**](ftp://100.100.100.100), again.

\_\_\_ Stop the wireshark capture.

**15. What is the source port number?**

\_\_\_ Repeat the process. You should see that the Hamilton assigns a new (consecutive) source port number each time a new session is started.

That’s it…..

You’re outa here !!!!

