ECE356 Wireshark Lab

Andrew Shim

19 September 2012

Part 1 - Getting Started

1.1

List up to 10 different protocols:

1. HTTP
2. TCP
3. GLBP
4. STP
5. LOOP
6. LLMNR
7. DHCPv6
8. ICMP
9. BROWSER
10. ARP

1.2

Looking at the red boxes in Figures 1 and 2 on the next page we see that the HTTP GET was sent at 20:01:13.271848 and the HTTP OK reply was received at 20:01:13.296157. Thus the time it took for the reply was:

$(0.296157 - 0.271858)$ seconds = **0.024309 seconds**

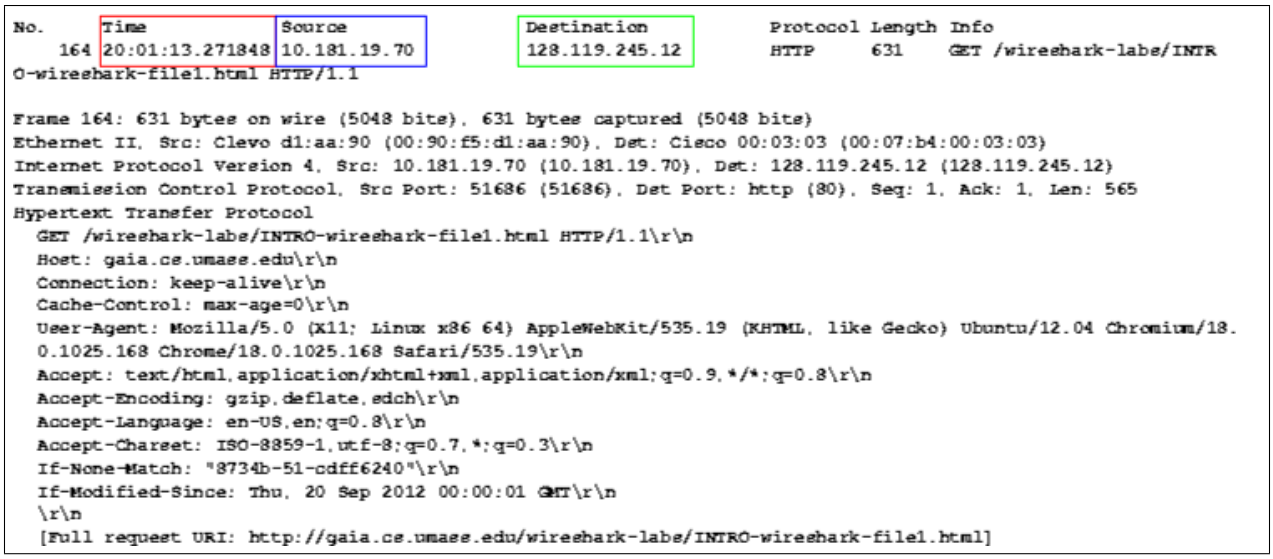

Figure 1: HTTP GET

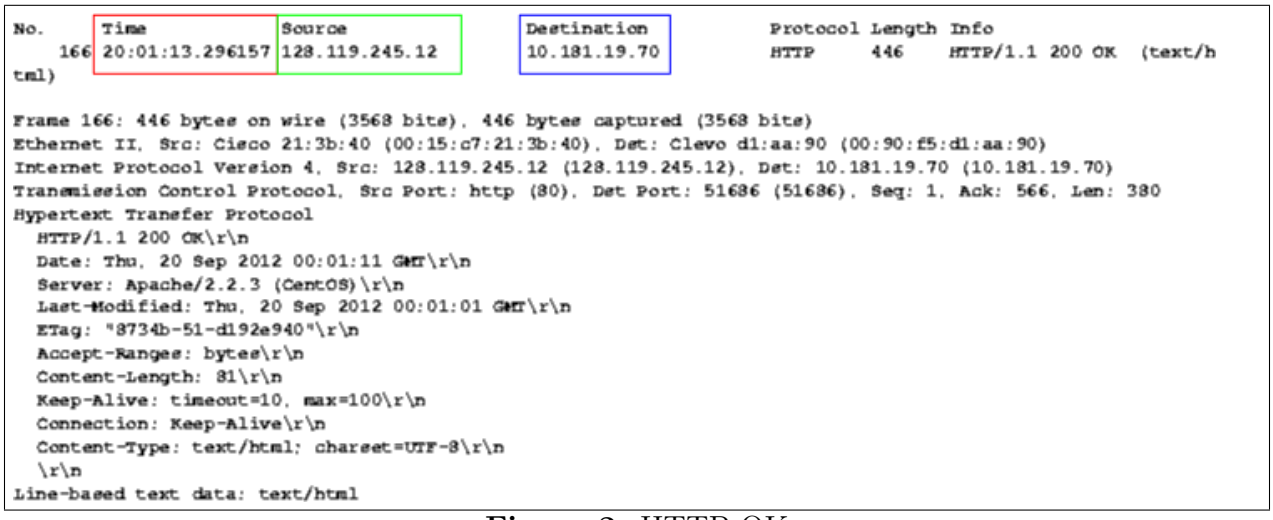

Figure 2: HTTP OK

1.3

Looking again at Figures 1 and 2 we see that the Internet address of gaia.cs.umass.edu is 128.119.245.12 (green boxes) and the Internet address of my host is 10.181.19.70 (blue boxes).

1.4

HTTP GET message: See Figure 1.
HTTP OK message: See Figure 2.

Part 2 - HTTP

2.1

If I just enter "http" in the filter then one other protocol, SSDP, gets additionally listed. Deducing from the semantics of the filtering criteria, I can conclude that the "http && (!udp.port == 1900)" criteria filters out any packets sent through UDP to the destination port 1900. Also, the "http" criteria allows the SSDP packets to appear in the output because those packets do use some HTTP-related methods although the primary underlying protocol is UDP.

2.2

IP address of my computer: 10.181.19.70
IP address of the server: 152.3.67.47

2.3

The status code returned from the server to my browser was 200. There are many other HTTP status codes ranging from the 100s to the 500s. The codes that I have seen most often are:

400 - Bad Request
403 - Forbidden
404 - Not Found
500 - Internal Server Error

2.4

Figure 3: Flow Graph Output

Examining the figure above we can see that every GET request is accompanied by an OK response. In a more general view, the client sends HTTP requests to the server and the server replies to the client through HTTP responses. Depending on the status of the server or a variety of other conditions, the server will send a particular HTTP response to the client (e.g. if the client is not allowed to access the data requested in the GET the server would reply with a 403 Forbidden response).

2.5

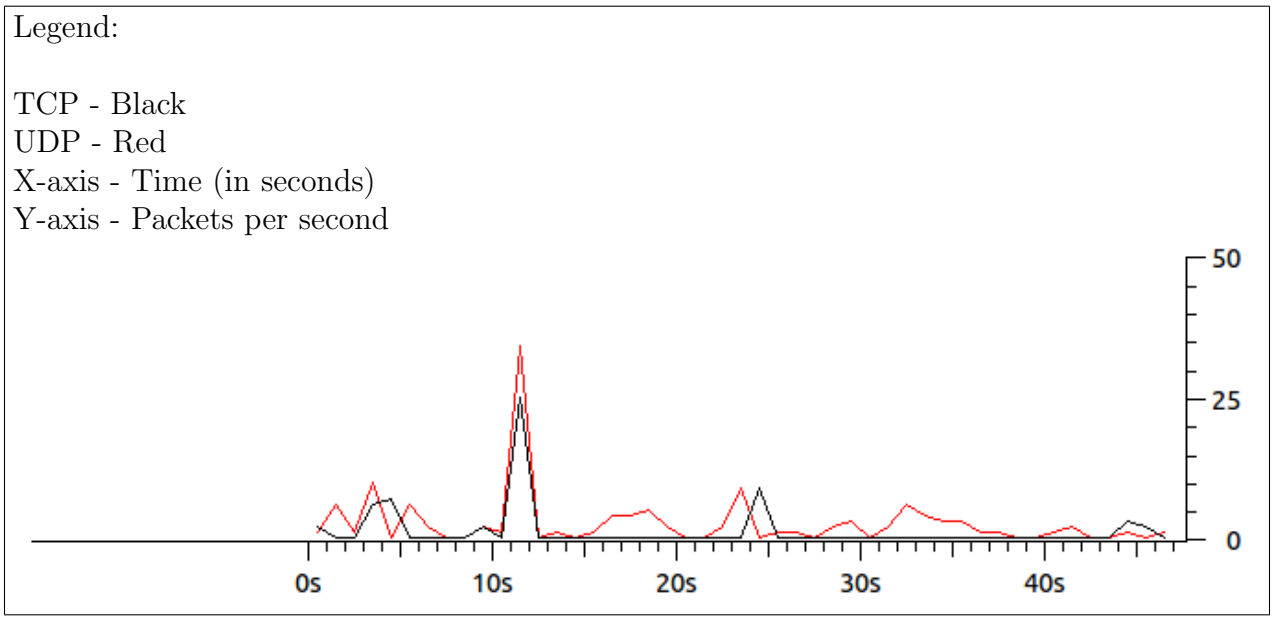

Figure 4: TCP vs. UDP I/O Graph

Looking at the I/O graph above we see that more UDP (red line) than TCP (black line) packets were sent. From taking a quick look at the Wireshark output, it looks as if most of the underlying communications used UDP. Basically, the communications that I (the user) do not necessarily need to worry about were primarily using UDP, whereas every communication that was a direct result of my actions (e.g. interaction with the browser) used TCP. Note the similarity of the UDP and TCP lines, where it appears as if the lines are overlayed. Assuming that this relationship is more than a coincidence, I would say that UDP packets are sent to pave the way for a TCP connection so that it can be properly established.

2.6

The HTTP message was sent over TCP. When you select a packet in Wireshark, Wireshark displays the details of the packet. The details section includes information about the frame, network connection, internet protocol, and which protocol it was sent through (e.g. TCP, UDP, etc.); from examining this section I was able to figure out that TCP was used.

Part 3: DNS

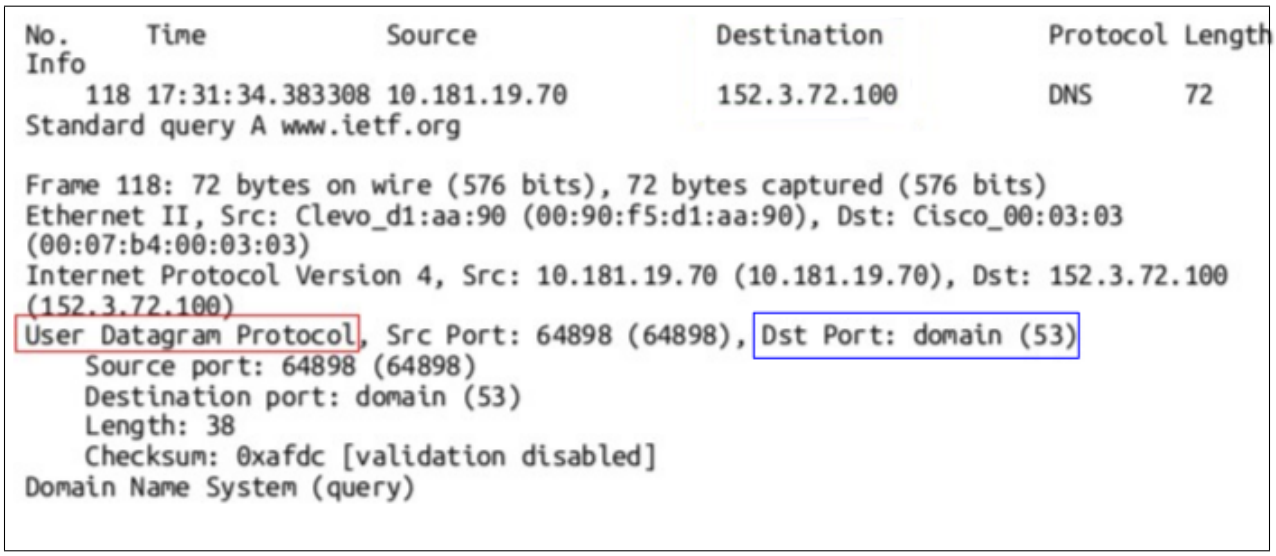

Figure 5: DNS Query Message

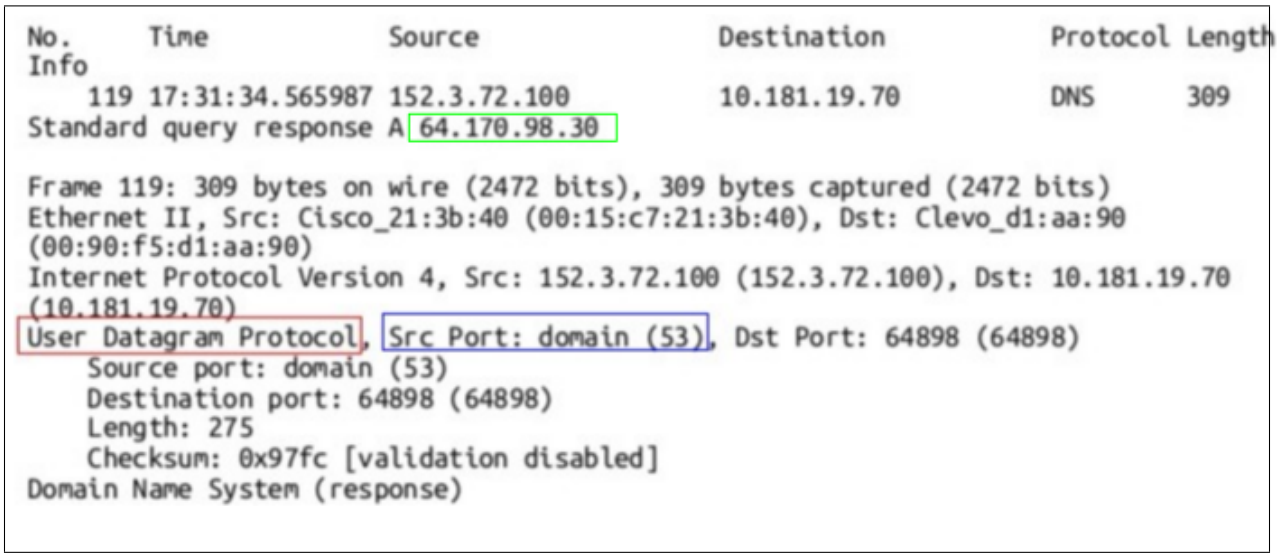

Figure 6: DNS Response Message

3.1

Looking at the red boxes in Figures 5 and 6 we can see that the DNS messages were sent over UDP.

3.2

Looking at the blue boxes in Figures 5 and 6 we can see that:

$$\text{Destination port for DNS query} = 53$$
$$\text{Source port for DNS response} = 53$$

3.3

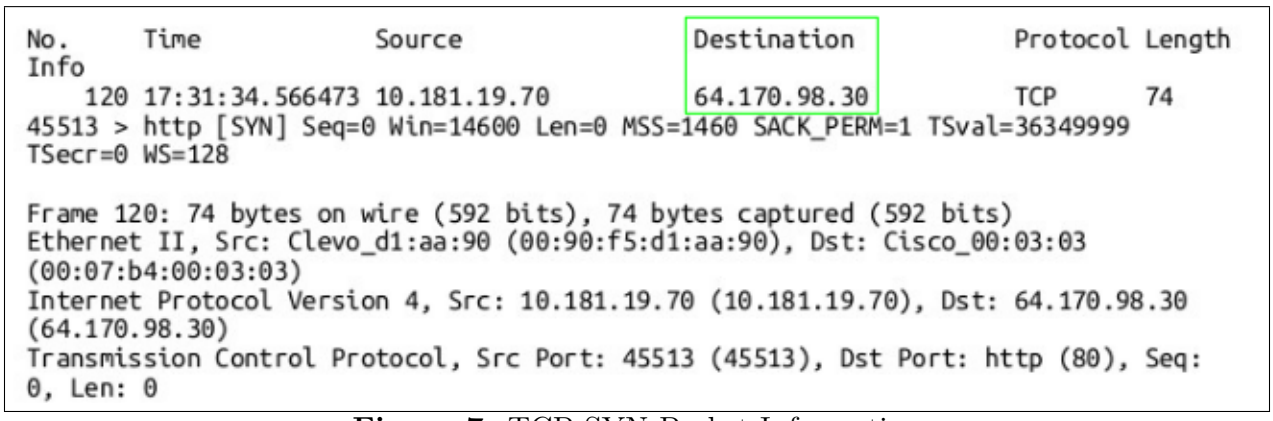

Figure 7: TCP SYN Packet Information

From Figure 7 we observe that the DNS response replied that the IP address of www.ietf.org is 64.170.98.30 (green box). Now looking back to Figure 6, we see that the TCP SYN packet was sent to that same IP address 64.170.98.30. So yes, the destination IP address of the TCP SYN packet corresponds to the IP address provided in the DNS response message.

3.4

My host did issue new DNS queries but, after looking at the HTML source of the web page, I do not believe that those queries were for the images because the images were provided by www.ietf.org. After the first DNS query for that name, my host cached the response IP address, so by the time it was retrieving the images it already knew which IP address it needed to get those images from. The new DNS queries had to do with the links on the web page that pointed to other servers for which my host did not know the IP addresses of (e.g. ioac.ietf.org, open-stand.org, rfc-editor.org).