Seneca | SCHOOL OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

Lab Exercise 05 – Using Wireshark to Examine the Transport Layer

Objectives

Part 1: Use Wireshark to familiarize yourself with the TCP Protocol.

Part 2: Use Wireshark to familiarize yourself with the UDP Protocol.

Background / Scenario

To complete this Lab Exercise you must download the sample Wireshark Capture files from Blackboard. The filenames are `http_witp_jpegs.cap` and `dns.cap`. For your reference, these are sample capture files provided through the Wireshark Wiki: https://wiki.wireshark.org/SampleCaptures where many more interesting sample capture files are available.

These sample captures will illustrate the functionality of the Transport Layer and how the information in the header is used to move information between the Application Layer and the lower layers of the OSI Model.

Required Resources

- 1 PC (Windows 7, 8, or 10 with internet access with Wireshark installed)
- Sample Capture Files.

Part 1: The TCP Protocol

In Part 1, you will examine the header fields and content in a TCP Segment (A Layer 4 PDU is called a segment). A Wireshark capture will be used to examine the contents in those fields.

The contents of this file have been captured using Wireshark running on the client PC. The network traffic has been filtered so that it only contains the one type of traffic we want to inspect.

Step 1: Open the capture file `http_witp_jpegs.cap` in Wireshark

The screen is split in 3. We will focus on the top section (it should be colour-coded right now). Using your knowledge of the Transport Layer and with reference to this capture file, answer the following questions.

Using the Numbering on the left side, which segments contain the three-way handshake (only refer to the first time you encounter the three-way handshake)?

_The segments 1,2 and 3 make three-way handshake. In the first segment the request is sent to server Where seq=0, the connection initiates and then in second segment the server sends acknowledgement back with seq=0 and ack=1, and requests a session with client; at the last in the third segment the client acknowledges a communication session with the server, where seq=1 and ack=1.The structure of the segments were as follows:

1 0.000000 10.1.1.101 10.1.1.1 TCP 62 3177 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM=1

2 0.000651 10.1.1.1 10.1.1.101 TCP 62 80 → 3177 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1

3 0.000697 10.1.1.101 10.1.1.1 TCP 54 3177 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0_____

What is/are the source port(s) (list all that you find)?

_The list of the source ports which found in the capture is as follows:3177,3179,3183,3184,3185,3187,3188,3189,3190,3191,3192,3193,3194,3195,3196,3197,3198,3199 and 3200.

What is/are the destination port(s) (list all that you find)? Which one appears most frequently?

_unlike source ports, I found only one destination port of number 80. This was the only and only port to which all the clients requests were sent. This is the port number of http, so all the requests were sent to http server.

What Application Layer protocol is associated with the most frequent destination port number (the official list of port numbers is here https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt)? As I mentioned in the above answer 'http' is the application layer protocol which is associated with the most frequent destination port number i.e; 80.

What RFC(s) is/are associated with this Application Layer Protocol (there are several RFCs that apply here, list one)? __The are many RFCs associated with http, RFC 7231 is one of them.

Step 2: From the Statistics Menu, select Conversations. When the Conversations window opens, select the TCP Tab.

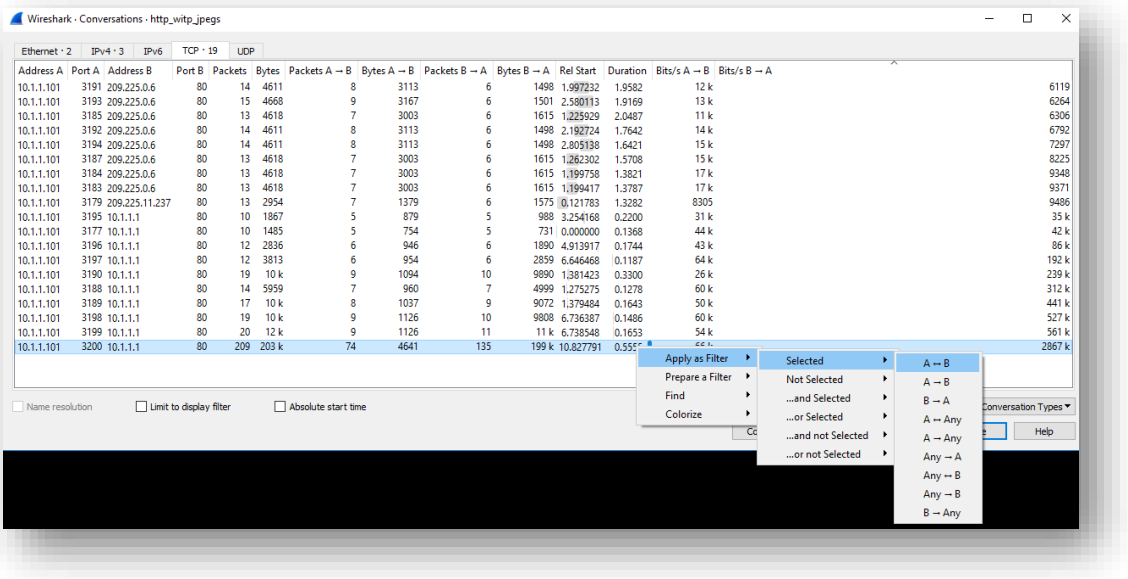
How many Transport Layer Conversations/Sessions are there? _There were 19 conversation sessions.

The information in this view can be sorted by clicking on the column header. Try clicking on the headers (Address A, Port A, Address B, Port B, etc.) to see how this works.

Which computer, Address A or Address B, do you think is the client? __According to me Address A is the client computer.

How Many different Servers is this client connecting to? This client is connecting to three different servers of addresses 10.1.1.1, 209.225.0.6 and 209.225.11.237.

Click on the Column Header "Bits/s B -> A", the largest value at the bottom (or top, depending on your sort direction) should be 2867k. Click somewhere on this line so that the entire line is highlighted. Right-click on this line and select "Apply As Filter" from the menu. Then select "Selected" and "A<->B" from the sub-menus. It should look like this:



When you have selected “A<->B”, click Close to close the Conversations Window. You should be back at the main Wireshark screen with only the Filtered conversation displayed. The numbers on the left side should start at 275 and end at 483.

The displayed traffic represents a single complete TCP “conversation” between two hosts: a client and a server. Note the three-way handshake before any application data is exchanged.

What is the source port for this conversation? _The source port for this conversation is 3200._____

What is the destination port for this conversation? The destination port for this conversation is 80._____

What is being requested by the client? The client requested for a website ‘/Websidan/2004-07-SeaWorld/fullsize/DSC07858.JPG HTTP/1.1’

_____

Reflection Question (no wrong answer, give it your best shot): Was the request successfully fulfilled? How might we know, based on this trace, if a problem has occurred?

__The request was successfully fulfilled because in the bottom segments, of the conversation, the server sent a status code 200, ok which means the request has been completed. We can identify the error if any occurs by recognizing this status code sent by server to client in the conversation. For example status code 400 means Bad Request.

_____

_____

Part 2: The UDP Protocol

In Part 2, you will examine the header fields and content in a UDP Segment (recall that a Layer 4 PDU is called a segment). A Wireshark capture will be used to examine the contents in those fields.

The contents of this file have been captured using Wireshark running on the client PC. The network traffic has been filtered so that it only contains the one type of traffic we want to inspect.

Step 1: Open the capture file `dns.cap` in Wireshark.

The screen is split in 3. We will focus on the top section (it should be colour-coded right now). Using your knowledge of the Transport Layer and with reference to this capture file, answer the following questions.

How do we begin communication between a client and a server when we use UDP?

_UDP does not form a connection with the server like TCP, but it just sends a datagram. First of all, a UDP socket is created, the socket is then bind to a server address. After that datagram packet is sent from the client and server processes the datagram packet and sends a reply. UDP packet has its own header information in packet along with the data. This data has source and destination ports on which communication occurs._____

_____

What is/are the source port(s) (list all that you find)?

All the source ports which I found in the capture are as follows: 32795, 32796, 32797, 1707, 1708, 1709, 1710, 1711.

_____

_____

_____

What is/are the destination port(s) (list all that you find)? Which appears most frequently?

_I only found one destination port in the capture of number 53.

_____

_____

_____

What Application Layer protocol is associated with the most frequent destination port number? File Transfer Protocol is the Application Layer protocol is associated with the most frequent destination port number

_____

What RFC(s) is/are associated with this Application Layer Protocol? _RFC 959 is associated with this Application Layer Protocol _____

Step 2: From the Statistics Menu, select Conversations. When the Conversations window opens, select the UDP Tab.

How many Transport Layer Conversations/Sessions are there? There were 8 conversations._____

In the context of UDP, what does a “session” mean (remember, UDP does not build a session before communicating, so what do these rows represent)? _UDP is a communication protocol that is used to make low-latency and loss-tolerating connections between two applications on the internet. Basically, UDP is connectionless protocol. UDP conversations are just combination of two end points. It does not have any three- way handshake between client and server. These rows just shows that data is sent to the destination.

__ _____

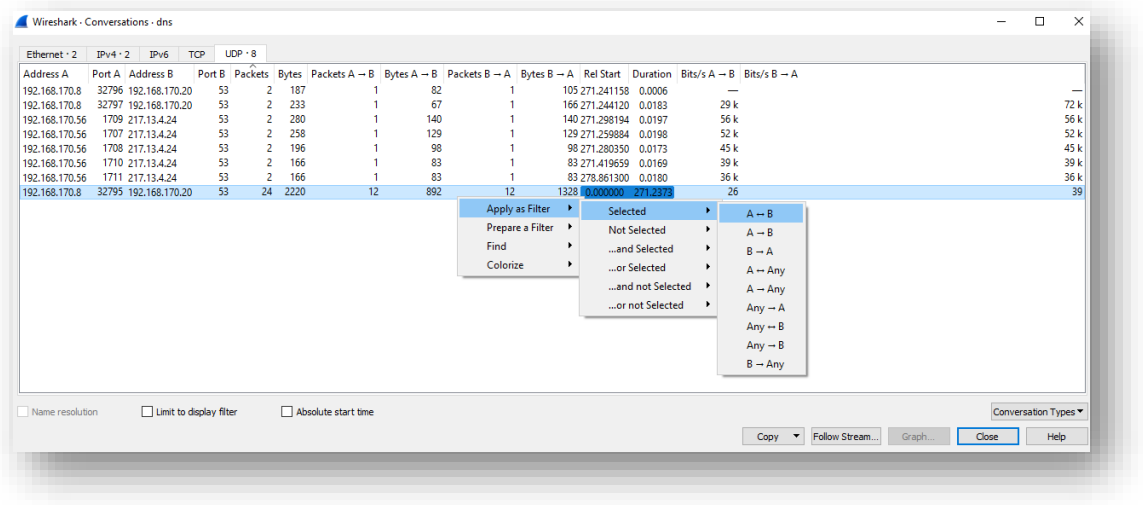
_____

Which computer, Address A or Address B, do you think is the client? _According to me, address A is client._____

How many different client addresses are there in this capture? There are two different client addresses in the capture, which are: 192.168.170.8, 192.168.170.56._____

How many different server addresses are there in this capture? The are two server addresses in the capture, which are: 192.168.170.20, 217.13.4.24._____

Click on the Column Header “Packets”, the largest value at the bottom (or top, depending on your sort direction) should be 24. Click somewhere on this line so that the entire line is highlighted. Right-click on this line and select “Apply As Filter” from the menu. Then select “Selected” and “A<->B” from the sub-menus. It should look like this:



When you have selected “A<->B”, click Close to close the Conversations Window. You should be back at the main Wireshark screen with only the Filtered conversation displayed. The numbers on the left side should start at 1 and end at 24.

What is the source port for this conversation? The source port for this conversation was 32795.

_____

What is the destination port for this conversation? _The destination port for this conversation was 53._

_____

Although UDP does not establish a session and maintain a connection like TCP does, we view this as a “conversation” in Wireshark because the application is using consistent source and destination numbers. How might this be useful when managing or troubleshooting the application or our network connectivity?

_Although UDP does not have anything similar to TCP sequence number and acknowledgement number to track the transmission of data. But still in the conversation of wireshark capture we can predict that how many packets of data have been sent to from the client to server and vice versa. In UDP we cannot make sure that packets are delivered in ordered form but can check if all the packets have been delivered or not, no matter in what order. This will help in checking packet loss if any occurs.

_____

_____

Reflection Question (no wrong answer, give it your best shot): What other information available in this view might be useful for managing or troubleshooting applications?

_In the screen capture of wireshark we can also see the duration of transferring the packets from client to server or from server to client. Along with that we can also check how many bytes have been transferred. These all things might help in troubleshooting applications if any packet loss occurs.

_____

_____

Reflection

The middle section of the three sections in Wireshark presents an analysis of each protocol layer. Select any row in the top section of wireshark and then view the information at each layer of the OSI model in the middle section. What does this analysis tell you about how the layers of the OSI model inter-relate with each other?

From middle section we can see all the information related to frame which is to be transferred. We can see that the connection is on Ethernet 2, what kind of IP address is this, is it IPv4 or IPv6, what are IP addresses of source and destination. Along with that we can also see the port numbers of both source and destination; from the domain name we can also check is it query from client to server or a response from server to client. This all information shows headers of all layers which are connected to the data during the transmission of data from one layer to another layer. So, all this information shows how all the layers of OSI model are connected to each other.