CS39006: Computer Networks Lab

Assignment 1

# Using Wireshark For Analyzing Network Packet Traces

Report By:

* *Robin Babu Padamadan [17CS10045]*
* *Shivam Kumar Jha [17CS30033]*

Objective

The objective of this assignment is to understand the Wireshark tool and how you can analyse network packet traces.

Methodology Used

Running the following commands on the terminal was used to get the results:

* `iperf -c 10.5.18.163 -u -b 28000` for UDP measurements
* `wget --no-proxy [http://10.5.18.163:8000/1.jpg](about:blank)` for TCP measurements

The filters used while capturing data using wireshark are as follows:

* `ip.addr == 10.5.18.163` for filtering out results whose ip address is not 10.5.18.163.
* `tcp` or `udp` for filtering out results not following the specified protocol. Note that without the ip.addr filter, this filter shows other crucial protocols as well.

Observations

##### **Q1. List the different protocols that you observe in the packet trace, at application, transport and network layer for each of the UDP and TCP test cases.**

For UDP:

None at Application layer

User Datagram Protocol at Transport layer (along with ICMP protocol)

Internet Protocol Version 4 at Network Layer

For TCP:

Hypertext Transfer Protocol at Application

Transmission Control Protocol at Transport layer

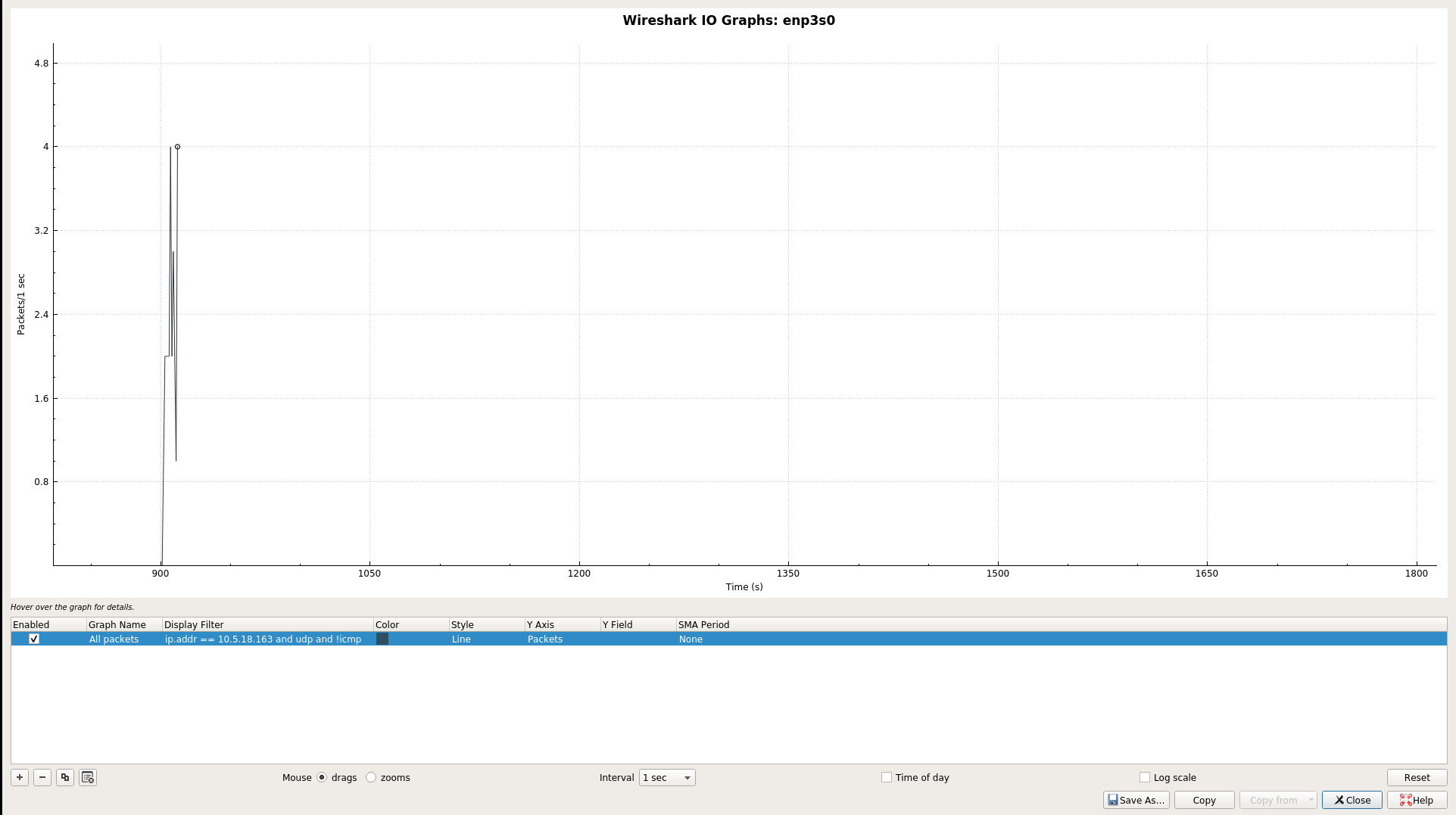
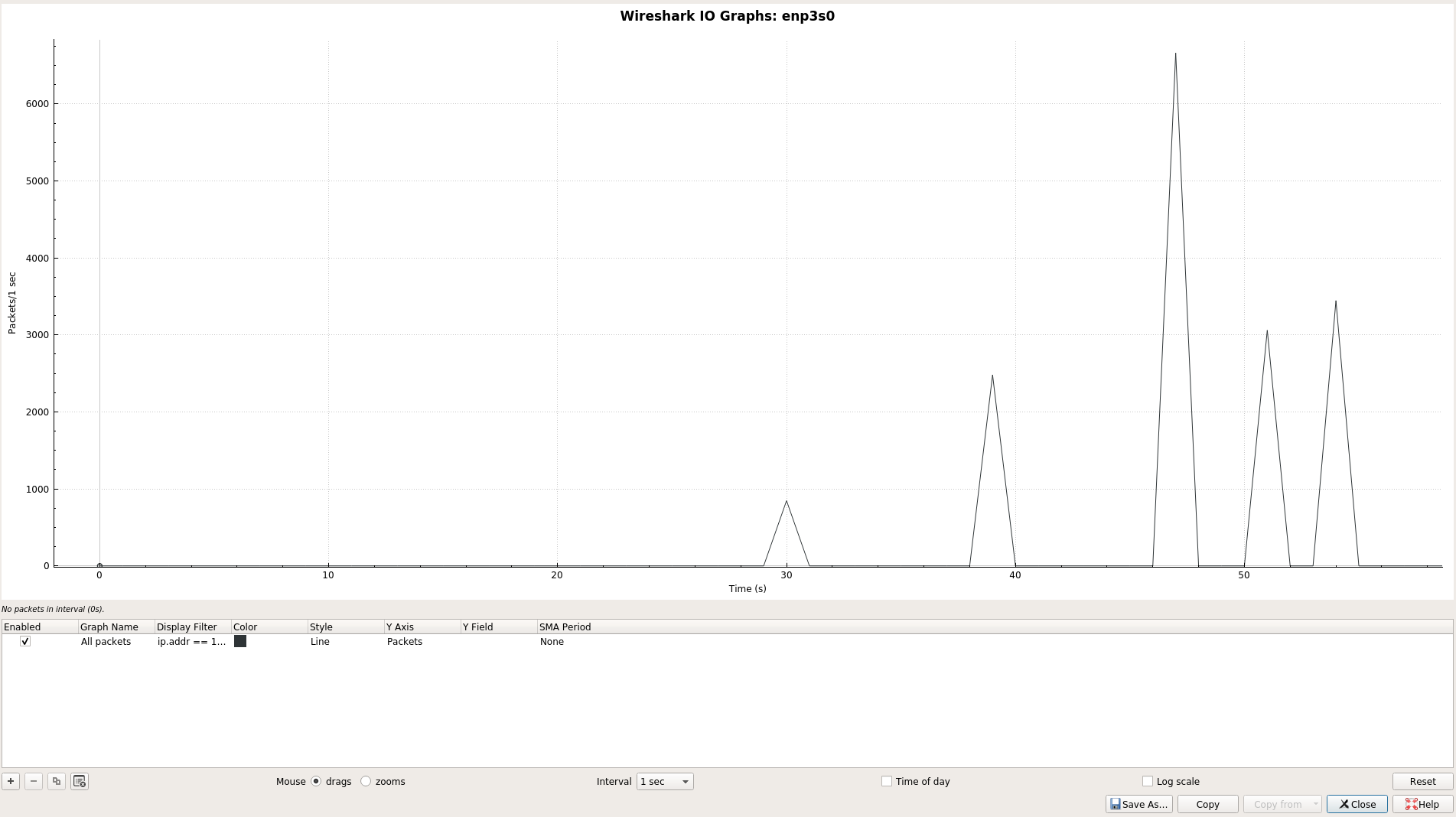
Internet Protocol Version 4 at Network Layer

##### **Q2. Analyse the packet trace using Wireshark and compute the following:**

1. How many TCP packets are transferred for each case while accessing the files pic1.jpg to pic5.jpg? Are all the packets the same size? What are the different packet sizes you observe for each file access?  
     
   Pic1: 599 TCP Packets with first one of size 17 bytes, then all but last of 1448 bytes and end at 972 bytes.  
   Pic2: 2152 TCP Packets with first one of size 17 bytes, then all but last of 1448 bytes and end at 1685 bytes.  
   Pic3: 4724 TCP Packets with first one of size 17 bytes, then all but last of 1448 bytes and end at 1798 bytes.  
   Pic4: 2281 TCP Packets with first one of size 17 bytes, then all but last of 1448 bytes and end at 1898 bytes.  
   Pic5: 2186 TCP Packets with first one of size 17 bytes, then all but last of 1448 bytes and end at 1336 bytes.  
     
   NOTE: Some TCP packets in the middle of communication deviated from 1448 bytes and were of sizes 2896 bytes, 4377 bytes, 5792 bytes etc. These occurred only a few times among thousands of packets and hence are ignored above.
2. For the test case with UDP, are all the UDP packets of the same size? If not, what are the different UDP packet sizes you observe?  
     
   Yes, all packets are the same size.  
   UDP: A total of 25 UDP Packets are sent, each of size 1470 bytes.

Explanation:

UDP is an unreliable protocol at the transport layer, however it maintains all packets of uniform size and transmits them at a fast rate. It is this fast and uniform rate of transmission that comes at a cost, making it unreliable and thus may lead to packet losses in the middle. However, TCP is a reliable protocol, thus it always ensures that the packets get transmitted. This reliability comes at the cost of speed and uniformity. Thus TCP packets are of different sizes and are transmitted at a slower rate than UDP. This unreliability is also the reason behind why UDP generally does not have any layer in the application layer, but TCP is used to transfer packets to the application layer.

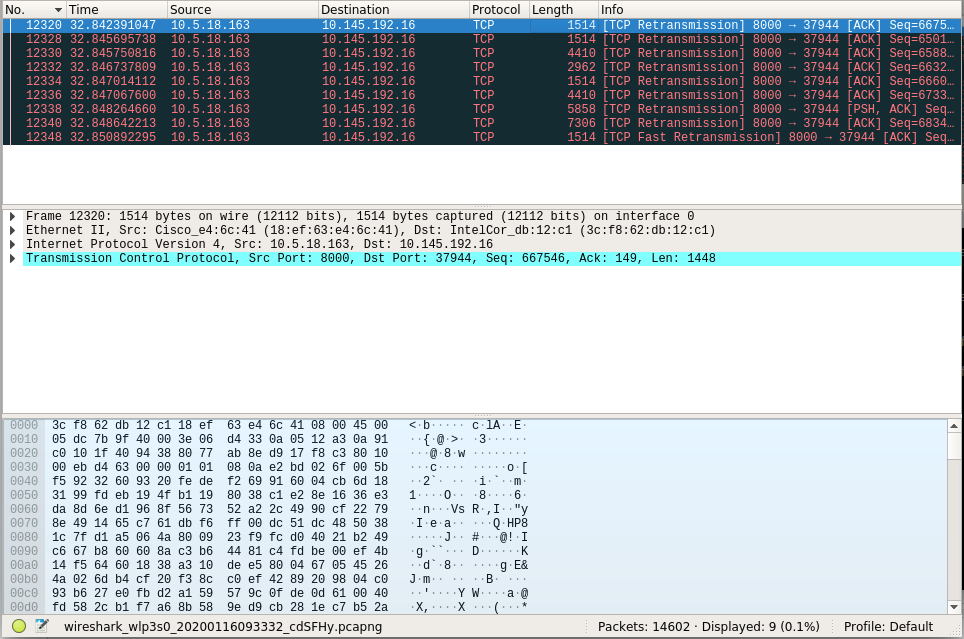
1. Observe the TCP and UDP throughput using Wireshark (Menu->Statistics->IO Graphs).  
   UDP:  
     
     
     
   TCP (from 1.jpg to 5.jpg, left to right):  
     
   
2. Compute the UDP throughput (amount of UDP data sent per second) for following cases of UDP traffic generation rates (bandwidth): (i) 64 Kbps (ii) 128 Kbps (iii) 256 Kbps (iv) 512 Kbps (v) 1024 Kbps (vi) 2048 Kbps

|  |  |  |
| --- | --- | --- |
| Bandwidth | Throughput (Measured on Wireshark) | Throughput (Measured on iPerf) |
| 64 Kbps | 68.14 Kbps | 62.44 Kbps |
| 128 Kbps | 133 Kbps | 125.14 Kbps |
| 256 Kbps | 265 Kbps | 248.71 Kbps |
| 512 Kbps | 528 Kbps | 501.6 Kbps |
| 1024 Kbps | 1054 Kbps | 976 Kbps |
| 2048 Kbps | 2107 Kbps | 1952 Kbps |

The throughputs measured by Wireshark and iPerf are close to the Bandwidth applied while sending the UDP Packets. We observe that the throughput measured by Wireshark is more than that measured by iPerf. This occurs as iPerf measures the payload data rate, that is the actual useful user-data sent inside packets. Wireshark captures all data and overheads, including user-data, plus packet headers around the user data, and frame headers around the packets.

##### **Q3. Analyze the number of TCP packets retransmitted (Use: tcp.analysis.retransmission filter) from Wireshark.**

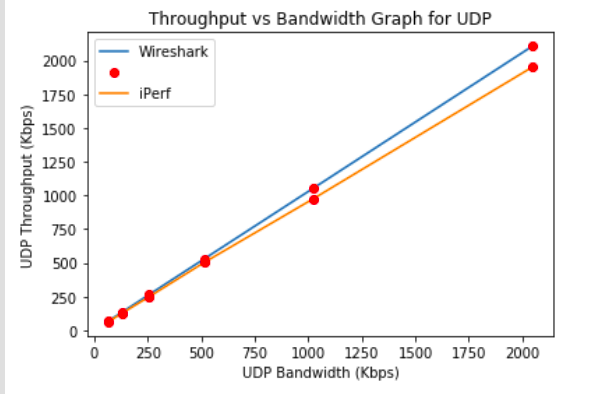
For most of the images and most of the transmission times there was no packet retransmission. However we occasionally observed packet retransmission for the third image (3.jpg). We observed 9 retransmissions (8 TCP Retransmission and 1 TCP Fast retransmission) in total.



Justification: The time when we conducted the transmission, the traffic congestion was very less hence the data transfer for most of the times occurred smoothly without any packet retransmission. However due to 3.jpg’s large size occasionally there occurred a small amount of packet retransmission, which may have occurred due to minor traffic congestion.

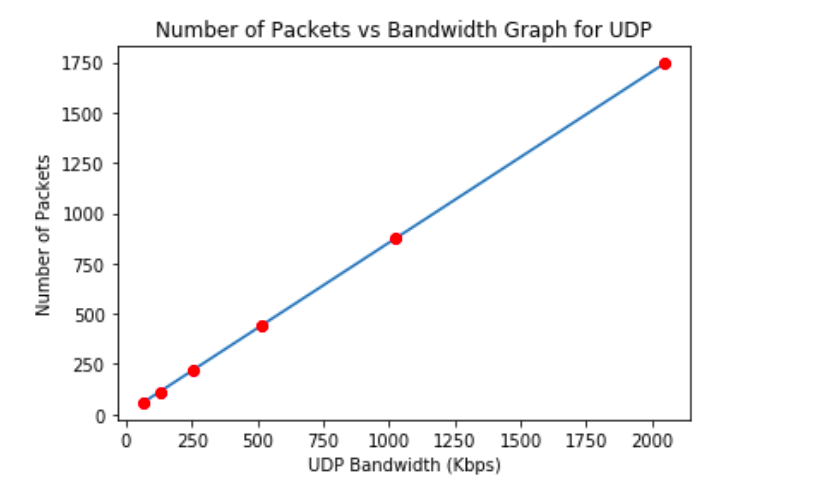
##### **Q4. Plot the following**

##### **a. UDP throughput with respect to the UDP bandwidth**



##### **b. Number of UDP packets transmitted with respect to UDP bandwidth**

|  |  |
| --- | --- |
| UDP Bandwidth | Number of Packets |
| 64 Kbps | 58 |
| 128 Kbps | 112 |
| 256 Kbps | 221 |
| 512 Kbps | 439 |
| 1024 Kbps | 874 |
| 2048 Kbps | 1745 |



We observe that all the graphs are linear. This is because the Data transfer rate (UDP Throughput) and the number of packets are both directly proportional to the UDP bandwidth.

Resources

* <https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html>
* <https://www.wireshark.org/docs/wsug_html_chunked/ChWorkDisplayFilterSection.html>
* <https://en.wikibooks.org/wiki/Internet_Technologies/Protocols>
* <https://superuser.com/questions/1275039/transport-level-protocol-vs-application-level-protocol>
* <http://www.omnisecu.com/tcpip/tcpip-model.php>

----------------------------------------------------- END--------------------------------------------------------