Assignment 5
Wireshark assignment for Transport & Link layer

Harsh Agarwal
CS15BTECH11019

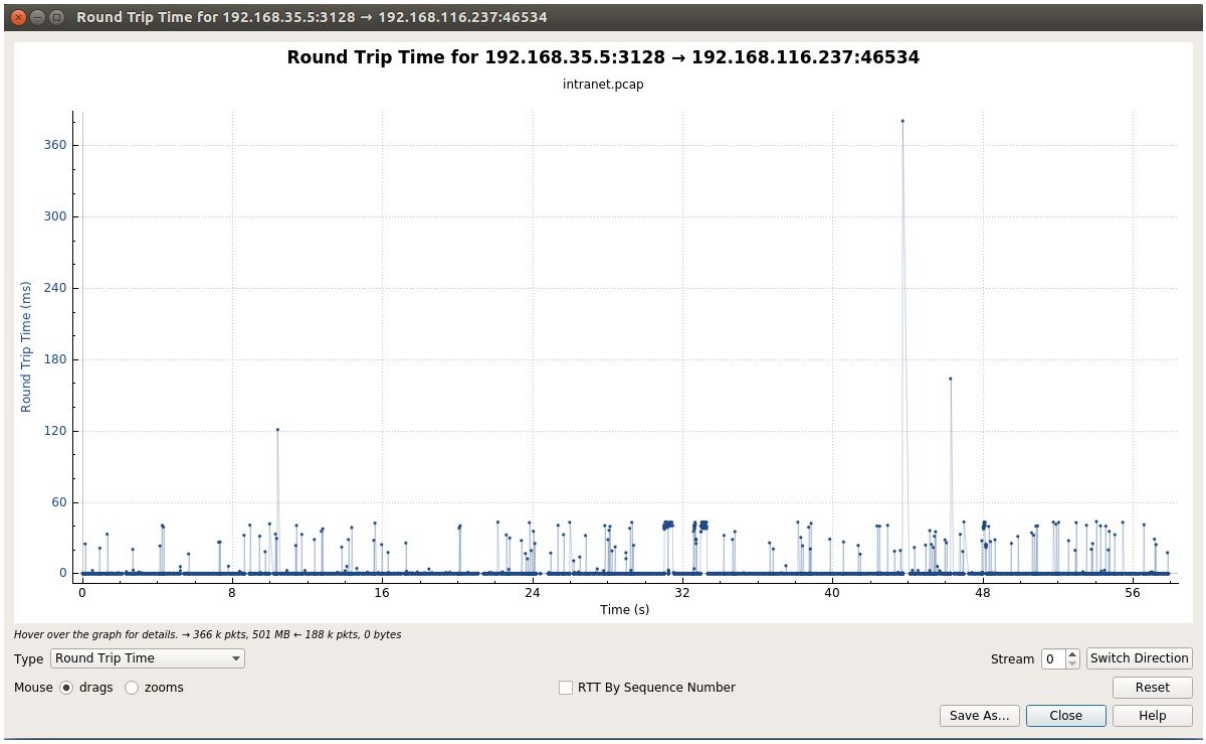
Transport Layer for Intranet

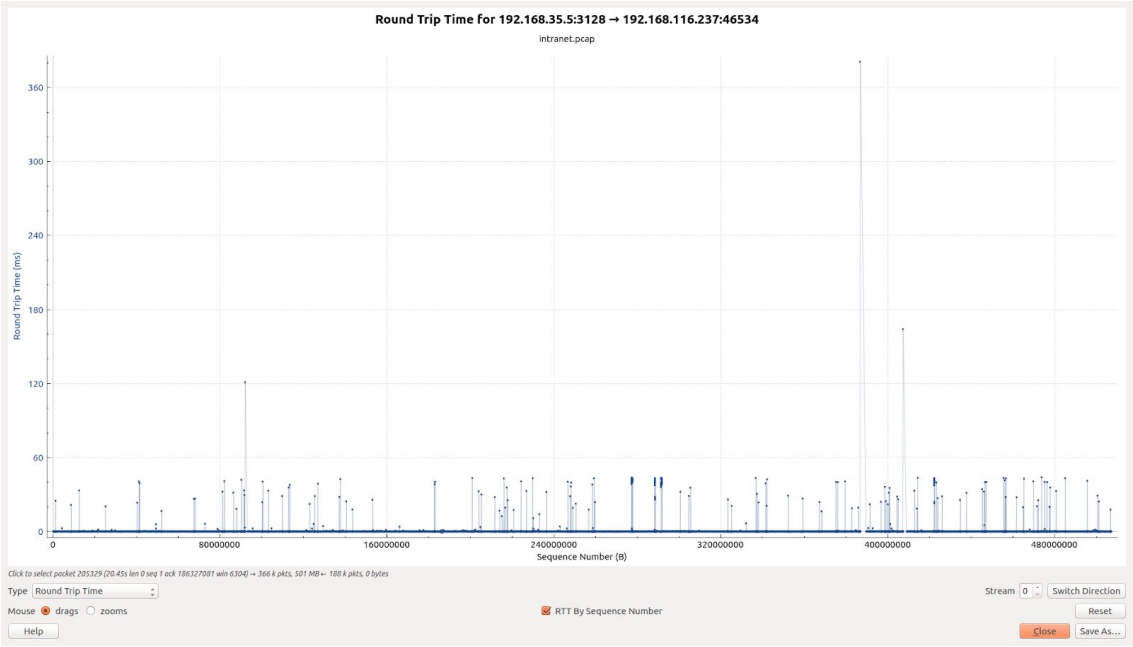
Analysis of downloading ubuntu-16.04.3-desktop-amd64.iso from intranet
http://intranet.iith.ac.in/files/os/ubuntu-15.04-desktop-amd64.iso
Server IP - 192.168.35.5:3128
My IP - 192.168.116.237:46534
Capture for 58 seconds
Command to filter this transaction :: **tcp.stream eq 0**

1)

a) I have plotted RTT(in ms) vs Time(s) & RTT(in ms) vs Sequence Number.

Most RTT values are close to 0, showing really less RTT.
There are a few fluctuations in the values with the maximum RTT being 380 ms.
Most of these fluctuation value are close to 50ms. There can be many reasons for these fluctuations.
Internet RTT was close to 10 ms, whereas intranet is close to 0ms. Since intranet is on a proxy server, this is much faster than internet(more proximity to the user).





b)

The IO Graph below is for tcp.analysis.bytes_in_flight vs time

bytes_in_flight tells how many bytes are now in flight for this connection. (i.e. un-ACKed bytes for this connection)

Since there is no direct way to determine the congestion window at the TCP sender(), the outstanding unacknowledged data is used to estimate the congestion window.

The red line shows window size advertised by the receiver.

The blue line show bytes in flight for the user.

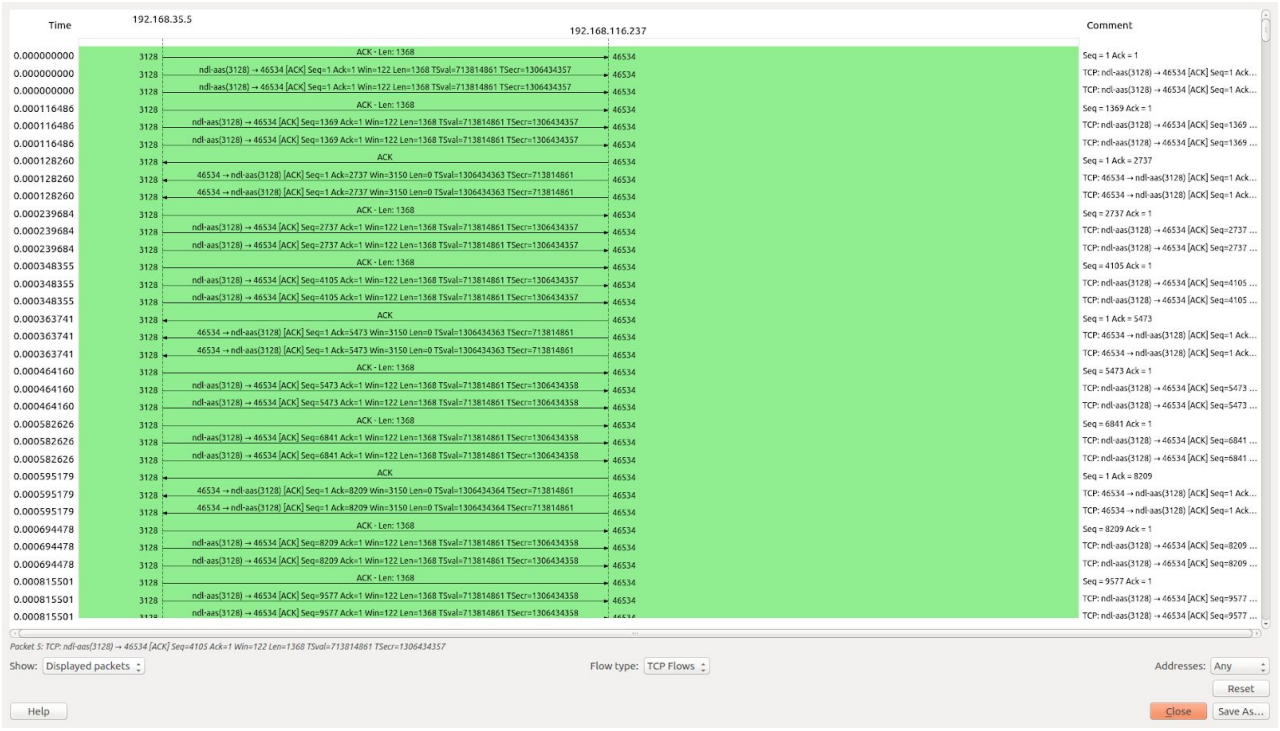
Since bytes_in_flight <= min(congestion_window , receiver_window)

And we observe blue line is always below red line , we can say with some assumptions that blue line plots congestion window

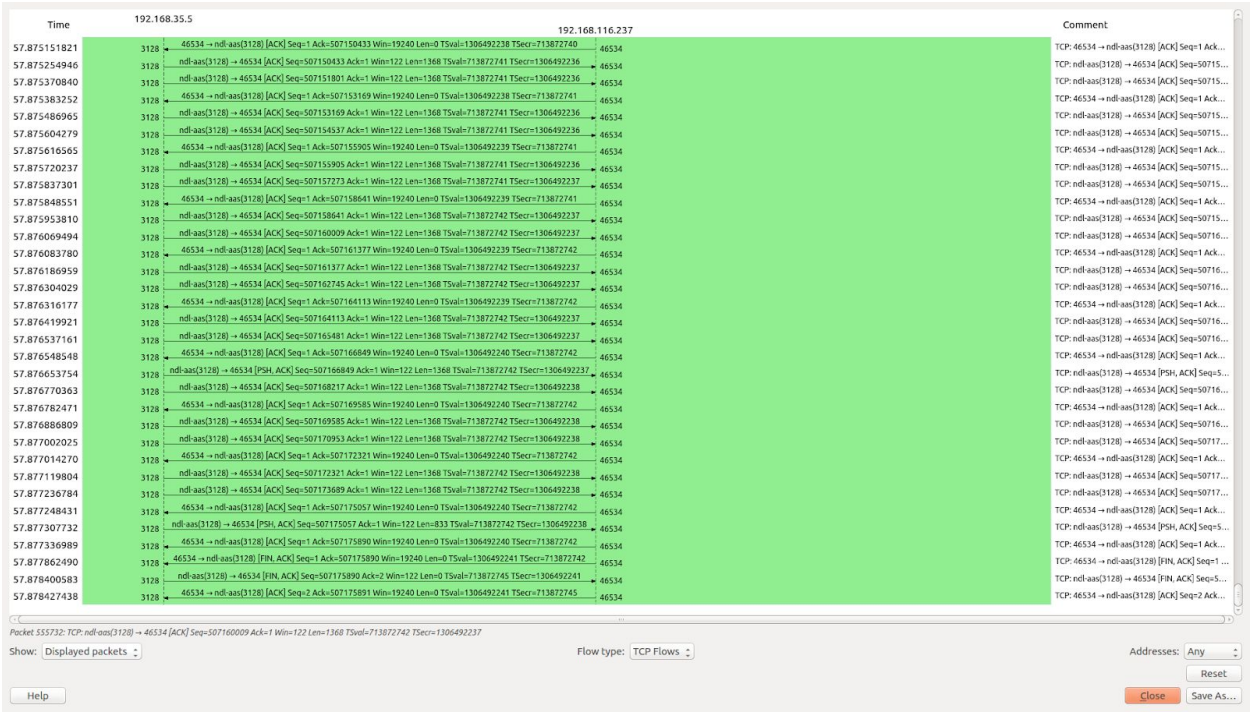
(Though there is a possibility that bytes_in_flight is less due to receiver sending ACK before the receive_window was full. Hence this graph gives a rough estimate of congestion_window (with the assumption that the above condition is not happening much)).

We observe that there is a log of fluctuation in the graph (thereby showing an unstable network(instability shown by above RTT vs time graph also))

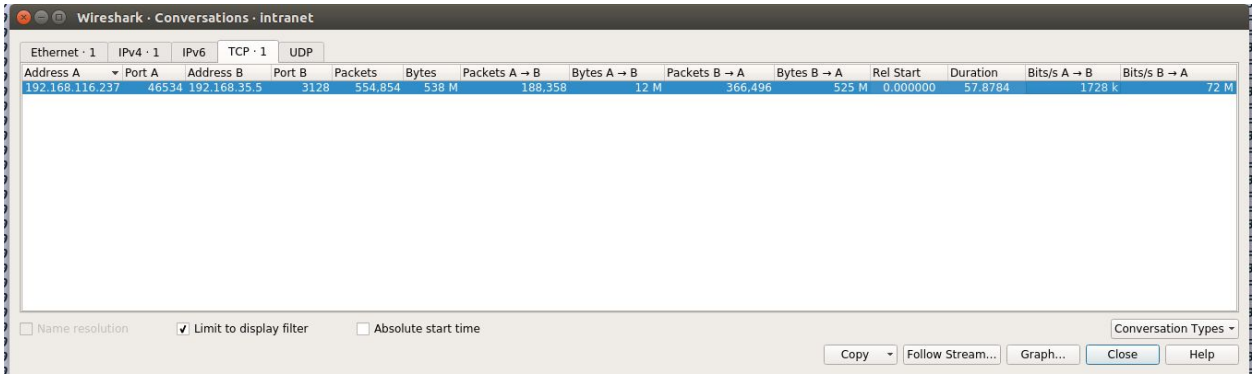
c) The flow graph feature shows a sender and a receiver view of the packet flow. (listing all seq-nos. & ack-nos. , etc). This is a really big graph as it tells of entire conversation.
I plotted TCP flow for the conversation between intranet and my IP.
The first screenshot is flow-graph first page. The initial seq,ack no. is 1 showing tcp handshake is starting (numbers are relative). And also sender is sending many pay-load (therefore this is not stop & wait). For first few iterations sender is sending same packet twice(meaning timeout must have occured). Also receiver sends 2 acks with seq=2737(1369+1368), meaning dup-ack. So on the conversation progresses.



The second screenshot is flow graph last page. The conversation ends with sender sending FIN ACKS to receiver (asking to close connection). Sender sends ACK to acknowledge that connection close message has been received.

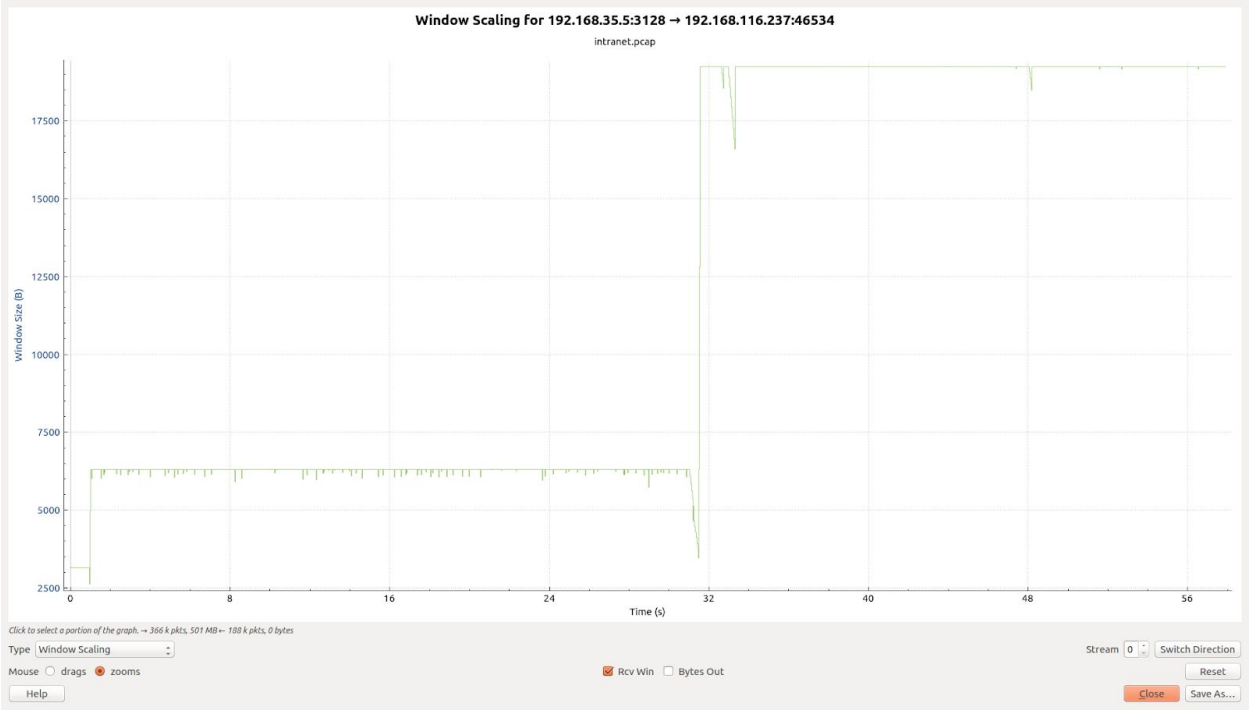


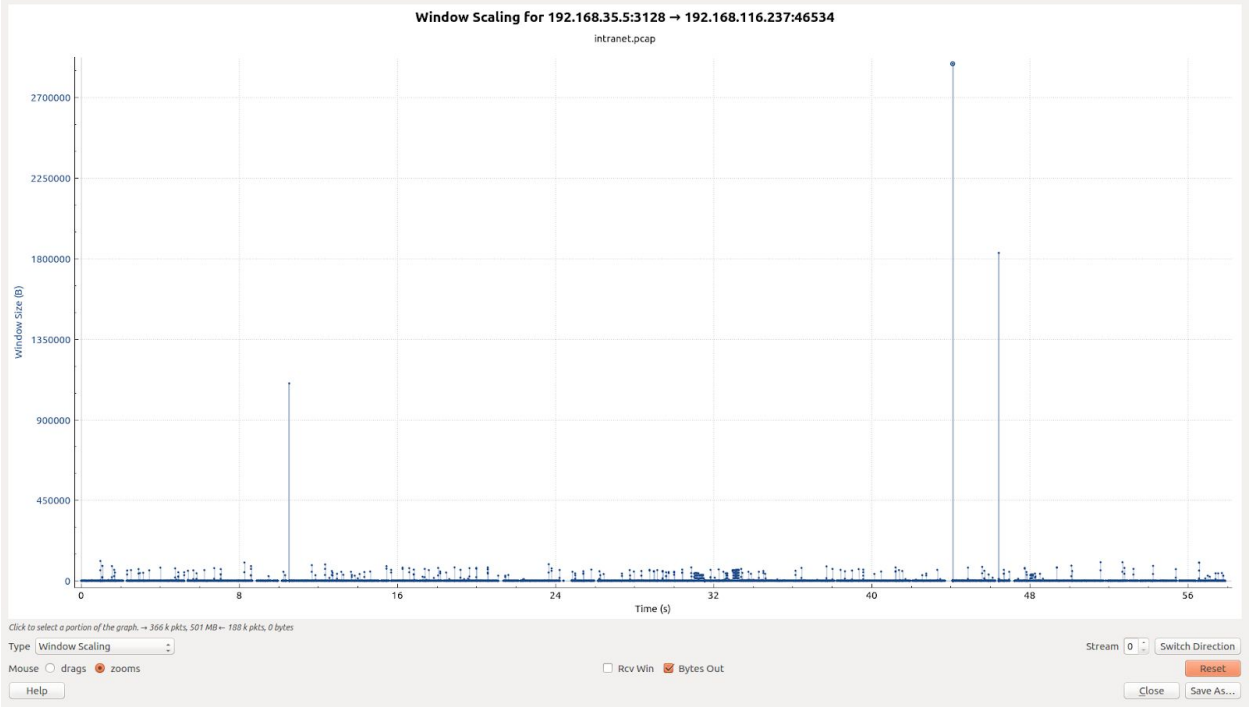
d)
Average Throughput from sender -> receiver :: 72 Mb/sec
Average Throughput from receiver -> sender :: 1728 kb/sec
Sender -> receiver is higher because here packets with payload is being sent, whereas in receiver -> sender just ACKs are being sent.



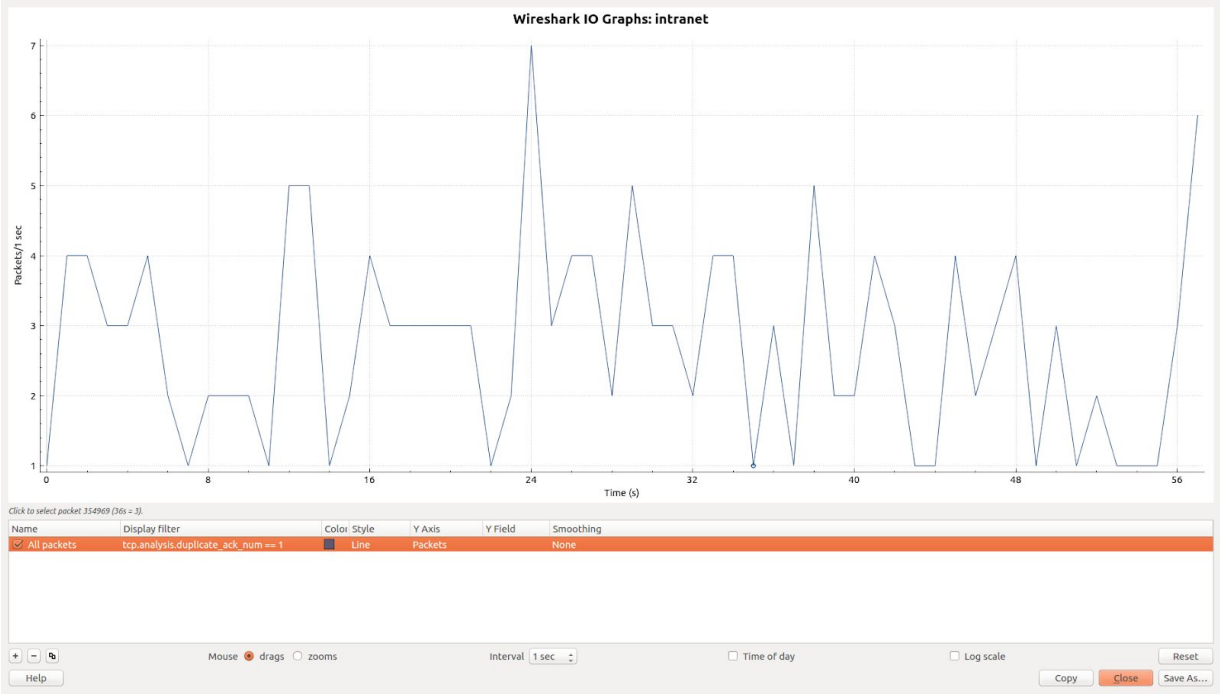
e)
The graph shows The size of the advertised receiver window incl. scaling factor.
The green plot shows receiver_window. In the plot
0-1s size is constant at 3510B
At 1s value increases to 6300B
1-31 s value ~6300B with fluctuations in middle
31-32s value decreases to 3450B
At 32s value shoots up to 19200B(with a few fluctuations)
The reason for the sudden shooting up can be that the initially sender sends smaller number of packets so as not to overwhelm the receiver. When sender sees it can send more packets , the value shoots up for increase throughput.

The 2nd graph is of Outstanding bytes vs time
I.e bytes that are unacked at receiver side. Most time it is between 1500B - 3000B (mostly below receive_window but for sometimes there are some shoot up beyond receive_window too)

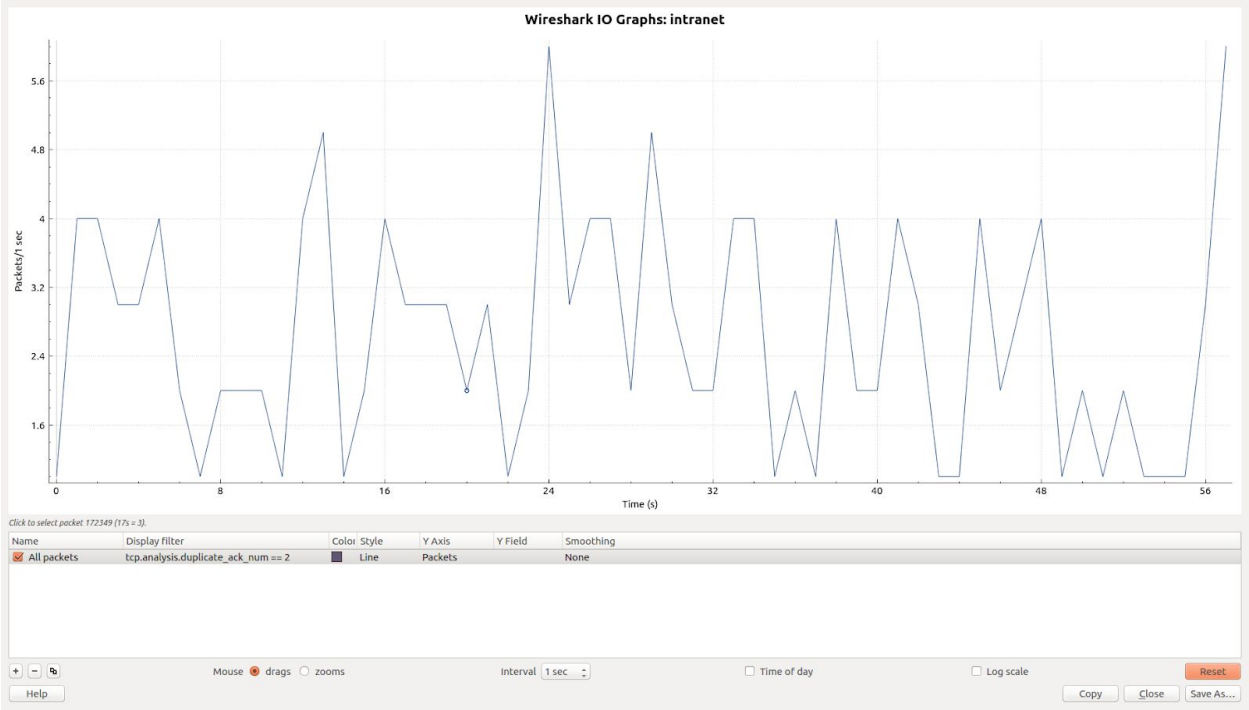

Window Scaling for 192.168.35.5:3128 → 192.168.116.237:46534
intranet.pcap
Window Size (B)
17500
15000
12500
10000
7500
5000
2500
0
8
16
24
32
40
48
56
Time (s)
Click to select a portion of the graph. → 366 k pkts, 501 MB← 188 k pkts, 0 bytes
Type Window Scaling
Stream 0
Switch Direction
Mouse drags zooms
Rcv Win
Bytes Out
Reset
Help
Close
Save As...


Window Scaling for 192.168.35.5:3128 → 192.168.116.237:46534
intranet.pcap
Window Size (B)
2700000
2250000
1800000
1350000
900000
450000
0
0
8
16
24
32
40
48
56
Time (s)
Click to select a portion of the graph. → 366 k pkts, 501 MB← 188 k pkts, 0 bytes
Type Window Scaling
Stream 0
Switch Direction
Mouse drags zooms
Rcv Win
Bytes Out
Reset
Help
Close
Save As...

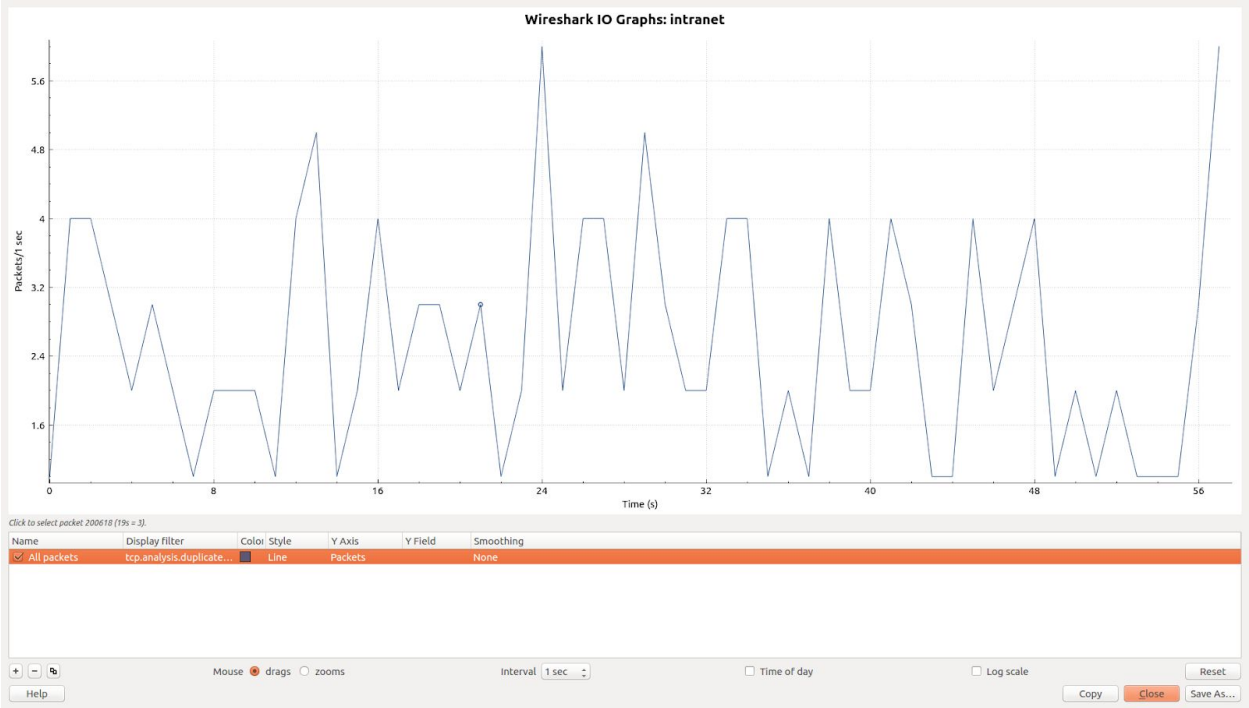
f) 1-duplicate Ack
Fraction :: 160 / 555772 (0.0%)
Command used :: **tcp.analysis.duplicate_ack_num == 1**



2-duplicate Ack
Fraction :: 153 / 555772 (0.0%)
Command used :: **tcp.analysis.duplicate_ack_num == 2**



3-duplicate Ack
Fraction :: 149 / 555772 (0.0%)
Command used :: **tcp.analysis.duplicate_ack_num == 3**



Number of dup-Acks is less compared to other packets. There is difference in the above 3 graphs because for some lost packets it is not needed to go upto 3-dup ACK. 3-dup ACK is mostly used in congestion control (via fast retransmit).

Drive Links for PCAP files

intranet.pcap (for intranet capture)
https://drive.google.com/open?id=1YJnjXx6cBlhVjDtB3ecMyJZfjLmDqR3C