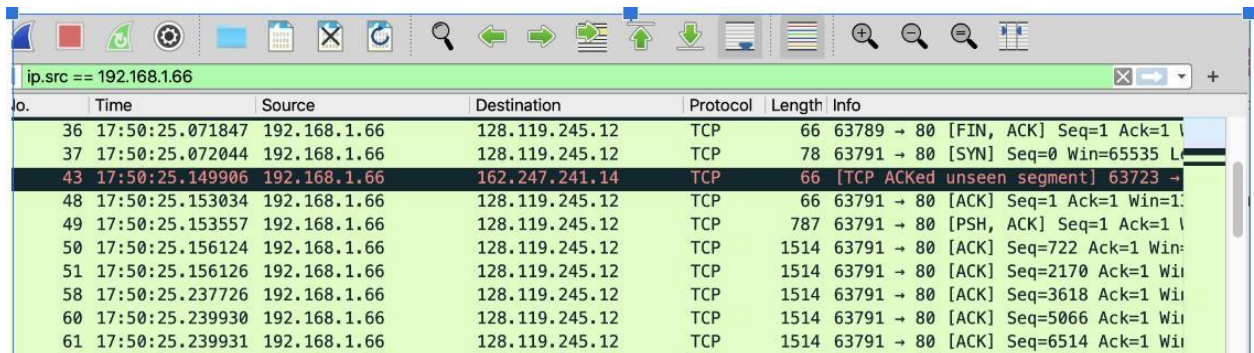


Part 2: TCP In this section we will explore several aspects of the TCP protocol.

First, we will upload a file to a webserver and observe the flow of packets over the TCP connection. Next, we will download a large file and introduce losses on the link to observe TCP's reliable data delivery service in action.

A.Upload a File to Web Server

In this section you will observe TCP uploading a text file. We recommend doing this section on your own computer as the TCP transfer on the VM is tricky. If you do work on the VM, disable HTTP in the Enabled Protocols. You should be able to find packets being uploaded, similar to the screenshot below.



The screenshot shows a Wireshark packet capture interface. The top toolbar includes buttons for file operations, network interfaces, and packet analysis. The packet list pane shows a filter 'ip.src == 192.168.1.66'. The packet details pane shows the selected packet (No. 43) with its structure: Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
36	17:50:25.071847	192.168.1.66	128.119.245.12	TCP	66	63789 → 80 [FIN, ACK] Seq=1 Ack=1
37	17:50:25.072044	192.168.1.66	128.119.245.12	TCP	78	63791 → 80 [SYN] Seq=0 Win=65535
43	17:50:25.149906	192.168.1.66	162.247.241.14	TCP	66	[TCP ACKed unseen segment] 63723 →
48	17:50:25.153034	192.168.1.66	128.119.245.12	TCP	66	63791 → 80 [ACK] Seq=1 Ack=1 Win=1
49	17:50:25.153557	192.168.1.66	128.119.245.12	TCP	787	63791 → 80 [PSH, ACK] Seq=1 Ack=1
50	17:50:25.156124	192.168.1.66	128.119.245.12	TCP	1514	63791 → 80 [ACK] Seq=722 Ack=1 Win=
51	17:50:25.156126	192.168.1.66	128.119.245.12	TCP	1514	63791 → 80 [ACK] Seq=2170 Ack=1 Wi
58	17:50:25.237726	192.168.1.66	128.119.245.12	TCP	1514	63791 → 80 [ACK] Seq=3618 Ack=1 Wi
60	17:50:25.239930	192.168.1.66	128.119.245.12	TCP	1514	63791 → 80 [ACK] Seq=5066 Ack=1 Wi
61	17:50:25.239931	192.168.1.66	128.119.245.12	TCP	1514	63791 → 80 [ACK] Seq=6514 Ack=1 Wi

Follow these instructions in the order they are given:

- Download the ASCII copy of Alice in Wonderland <http://gaia.cs.umass.edu/wiresharklabs/alice.txt> and store the file on your computer. (Read if you have time- it's a great story!)
- Load the page <http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html> •h Start Wireshark and begin packet capture.
- As you use the buttons displayed in the web page to upload the Alice in Wonderland file, observe the URL bar in the browser and answer Question 1 below.

- Once the file has been uploaded, a short congratulations message will be displayed in your browser window

Questions:

1. Did the URL displayed by your browser change to indicate the data that was going to be uploaded? Do some research on data or form uploads with HTTP and explain which HTTP method you think was used to upload the file. Make sure to state the reasons you think a particular method was used (hint: there are at least 2 ways to determine).
 - The URL displayed did change to `gaia.cs.umass.edu/wireshark-labs/lab3-1-reply.htm`
 - I think the method used was the HTTP POST method. The POST method is the traditional form-based file upload method. In this case we submit the file in an input field. This method is also commonly used for smaller file uploads such as images and documents.

TCP Basics

Answer the following questions for the TCP segments you observe in Wireshark. Attach a screenshot with markup for the questions. You might be able to combine several questions into one screenshot.

2. Connection Setup:

Find the TCP 3-way handshake:

- a. What display filter in Wireshark did you use to find the 3-way handshake?

- The filter I used was `tcp.flags.syn == 1 or tcp.flags.ack == 1`

- b. Does the browser or the server initiate the handshake?

- The browser initiates the handshake.

- c. What is the sequence number of the TCP SYN segment that initiates the TCP connection?

- the sequence number is 0 and is highlighted in the screen shot.

What information in the segment header that identifies the segment as a SYN segment? Show a screenshot of this information.

- d. What is the sequence number of the SYN-ACK segment sent by `gaia.cs.umass.edu` to the client computer in response to the SYN? Is it the same as 2c? Why or why not?

- The sequence number of the SYN-ACK segment sent by `gaia.cs.umass.edu` is the same as the SYN, 0, because the SYN-ACK is

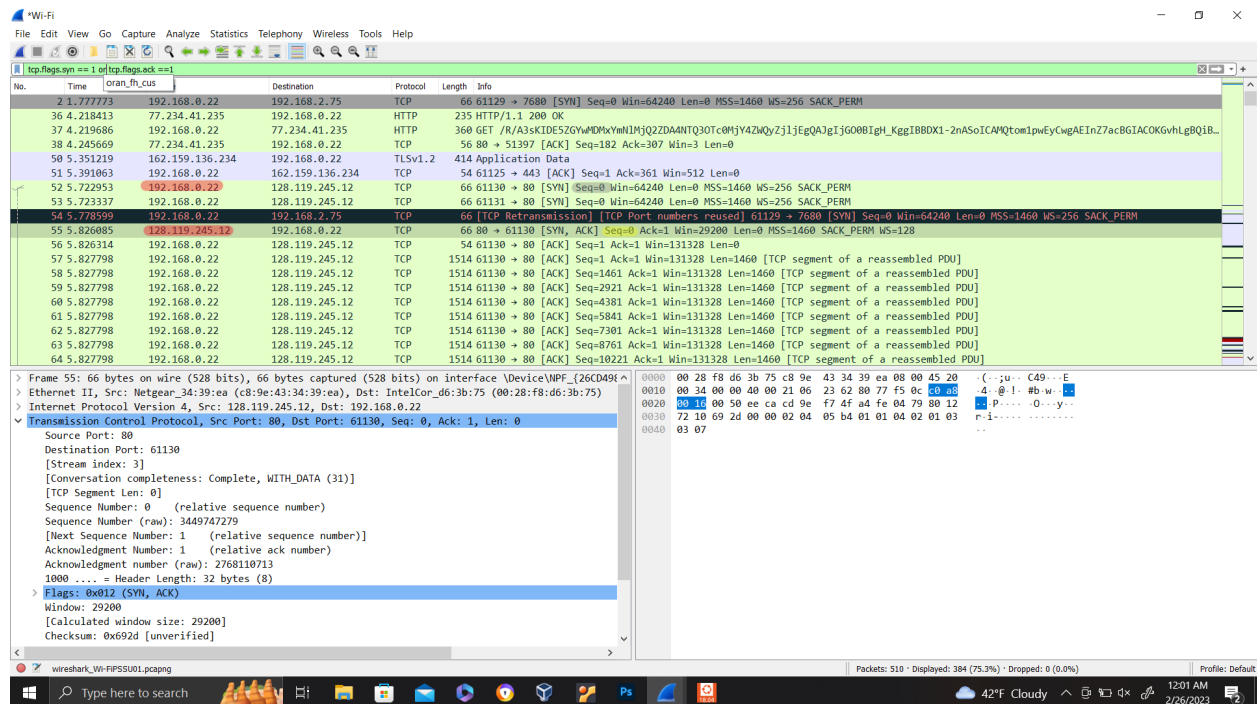
acknowledging the receipt of the SYN segment and establishing the initial sequence number for data sent from the server.

What is the value of the Acknowledgement field in the SYN-ACK segment?
How did gaia.cs.umass.edu determine that value?

- The value in the acknowledgment field is 1. When the server receives the SYN packet and responds with the SYN-ACK packet, the server sets the Acknowledgement field to the next expected sequence number, and since the previous was zero it would make sense that the following would be 1.

What is in the segment header that identifies the segment as the SYNACK?

- The source port number of the TCP segment contains the port number of the server. the Destination Port number will contain the port number of us the client. The sequence number field will contain the initial sequence number of the server. The Acknowledgement number will contain the acknowledgement number from the client. The control Flags will be set to indicate that this segment is a SYN-ACK segment and that the connection is being established. Also the window size field will contain the size of the server's receive window.
- e. How much time elapses from when the SYN is transmitted until the SYN ACK is received? What does this time represent? Hint: In class we have talked about the "small" SYN packets as having negligible transmission time. Explain your answer.
- 0.103 seconds = 103 milliseconds. This was calculated by taking the difference of the times between the time the SYN was transmitted and the SYN-ACK was received.(Highlighted in the screenshot.) This represents the time to establish a TCP connection between a client and server. This time includes propagation delay, transmission delay, and processing delay of the network devices and hosts involved in the communication.
- f. What is the IP address and port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? Is this port number well-known or an ephemeral port?
- the IP address of the client is 192.168.0.22. Its port number is 61130. This is not a well know ephemeral port.
- g. What is the IP address and port number used by gaia.cs.umass.edu? Is this port number well-known or an ephemeral port?
- The ip address of gaia.cs.umass.edu is 128.119.245.12. The port number it is using is 80, which is a well-known ephemeral port, used for HTTP.



TCP Upload

3. Beginning the File Upload:

a. According to Wireshark, which HTTP method was used to upload the file?

- According to Wireshark the POST method was used.
- i. What display filter in Wireshark did you use to find it?
- I used the http filter.
- ii. What is the sequence number of the TCP segment that carries this HTTP message?
- The sequence number is 1.(highlighted below)

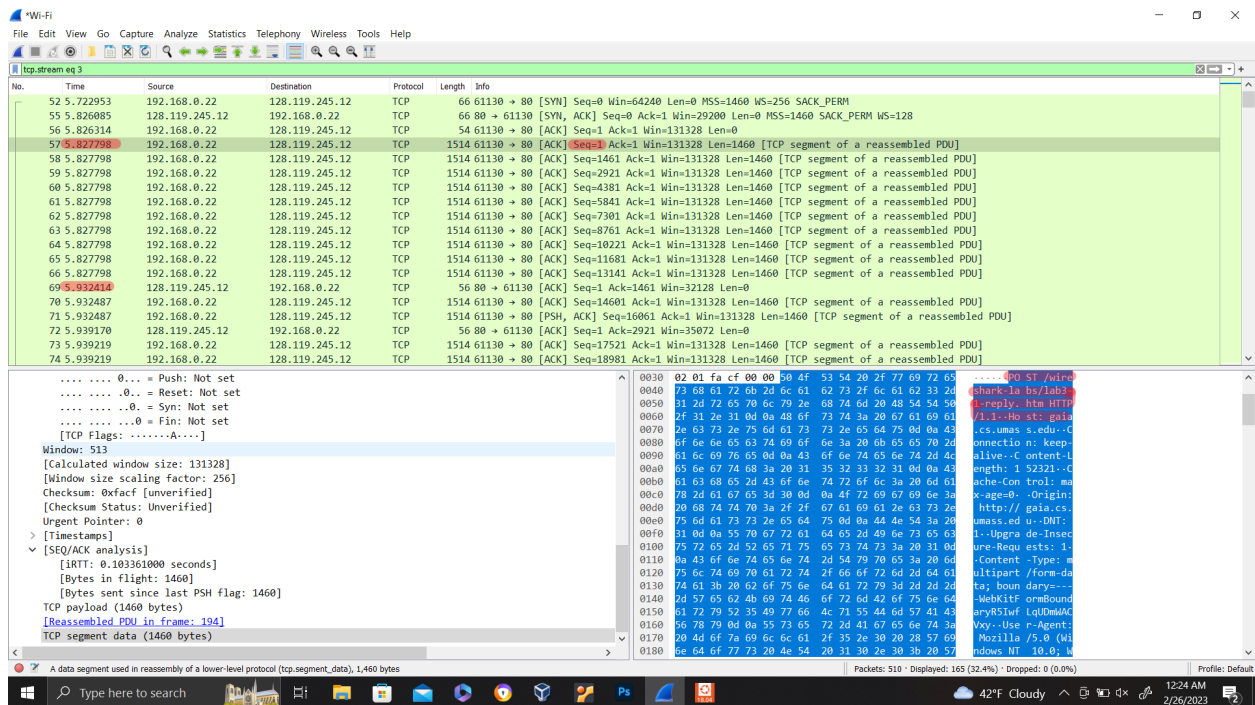
b. Find the first TCP segment that is transferring data, (i.e. part of the Alice in Wonderland file). Note: this segment might be carried with the HTTP message in 3 c. Consider this as the first segment of the file transfer. Calculate how long it takes to send this segment and receive an ACK packet – call this RTT(1st Data Packet). Given your previous measurement for RTT(SYN->SYNACK) and now for RTT(1st DataPacket), what is an approximation for the actual transmission time of 1st DataPacket? . Explain your reasoning and results.

1st RTT: 5.932414s - 5.827798s = 104.6 milliseconds

Given my previous measurement for RTT(SYN->SYNACK) the transmission delay

was about 10 milliseconds for the transmission time of the 1st packet. The propagation delay probably took most of the time, for the initial TCP connection. Although once connected, most likely the processing delay accounted for most of the RTT, given the transmission delay is usually miniscule.

Highlighted below are the packet times used for the calculation and the sequence number for the packet that carries the HTTP message, for question 3a) ii.



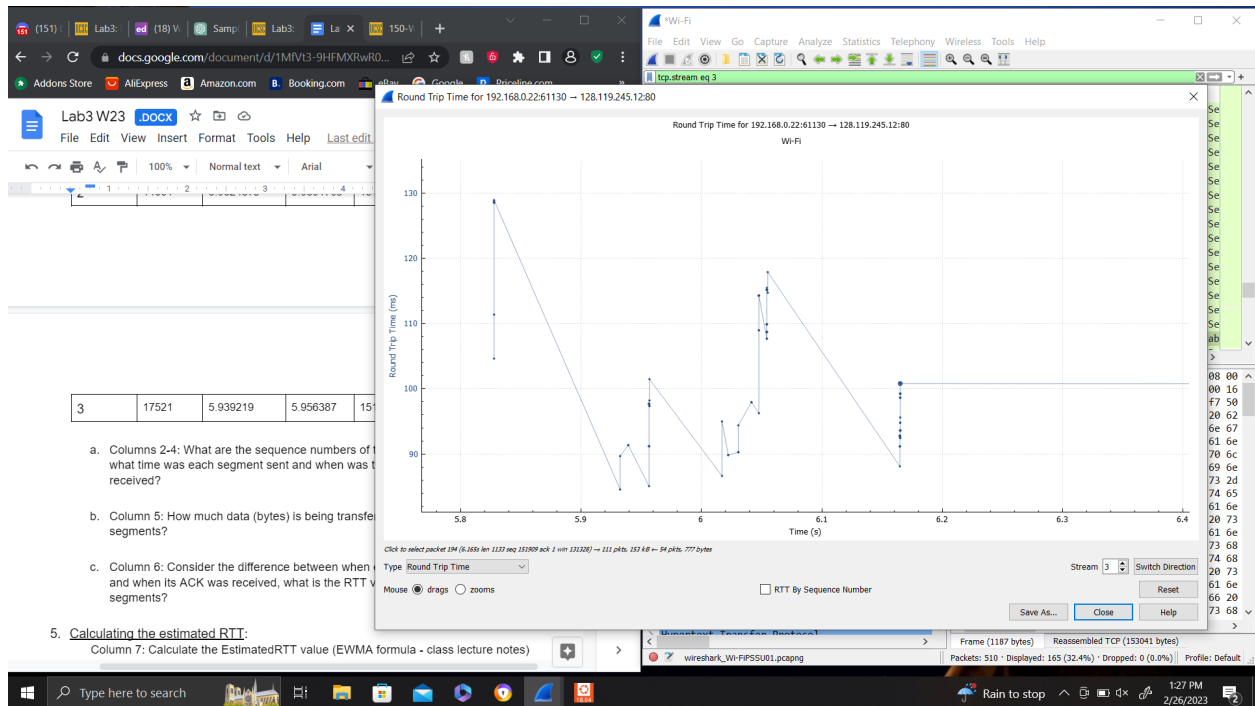
4. What's going on?

Let's make a table. Select 3 transmitted segments: the HTTP method (3a) and 2 segments which are transferring data ("the book").

#	SN	Time Transmitted	Time ACK Received	#bytes transfered	Sample RTT	Estimated RTT
1	1	5.827798s	5.932414s	1514	104.6ms	104.6ms
2	14601	5.932487s	5.939170s	1514	111.37ms	105.44ms

3	17521	5.939219	5.956387	1514	128.58ms	107.59ms
---	-------	----------	----------	------	----------	----------

- a. Columns 2-4: What are the sequence numbers of the segments you chose? At what time was each segment sent and when was the corresponding ACK received?
 - b. Column 5: How much data (bytes) is being transferred in each of these segments?
 - c. Column 6: Consider the difference between when each TCP segment was sent, and when its ACK was received, what is the RTT value for each of the three segments?
5. Calculating the estimated RTT:
- Column 7: Calculate the EstimatedRTT value (EWMA formula - class lecture notes) after the receipt of each ACK. To do this, assume that the initial value of the EstimatedRTT is equal to the measured RTT for the first segment. Then use the EstimatedRTT equation for the subsequent 2 segments. Use $\alpha = 0.125$
6. Graphing the RTT:
- Wireshark has a nice feature that allows you to plot the RTT for each of the transmitted TCP segments. Click on a TCP segment that is being sent from the client to the gaia.cs.umass.edu server. Then select: Statistics->TCP Stream Graph->Round Trip Time Graph. How does it compare with your calculations in your table? Include a screenshot of the graph for the duration of the file transfer. Note: the direction of the transfer is important!
- The RTT graph ranges from around 85ms to 130ms. My calculations seem to be right about the median of that range.



7. TCP ACKs: How much data does the receiver typically acknowledge in an ACK? How did you determine this? Referring to the ACK Generation slide we discussed in class, can you determine which ACK scheme the receiver is following?

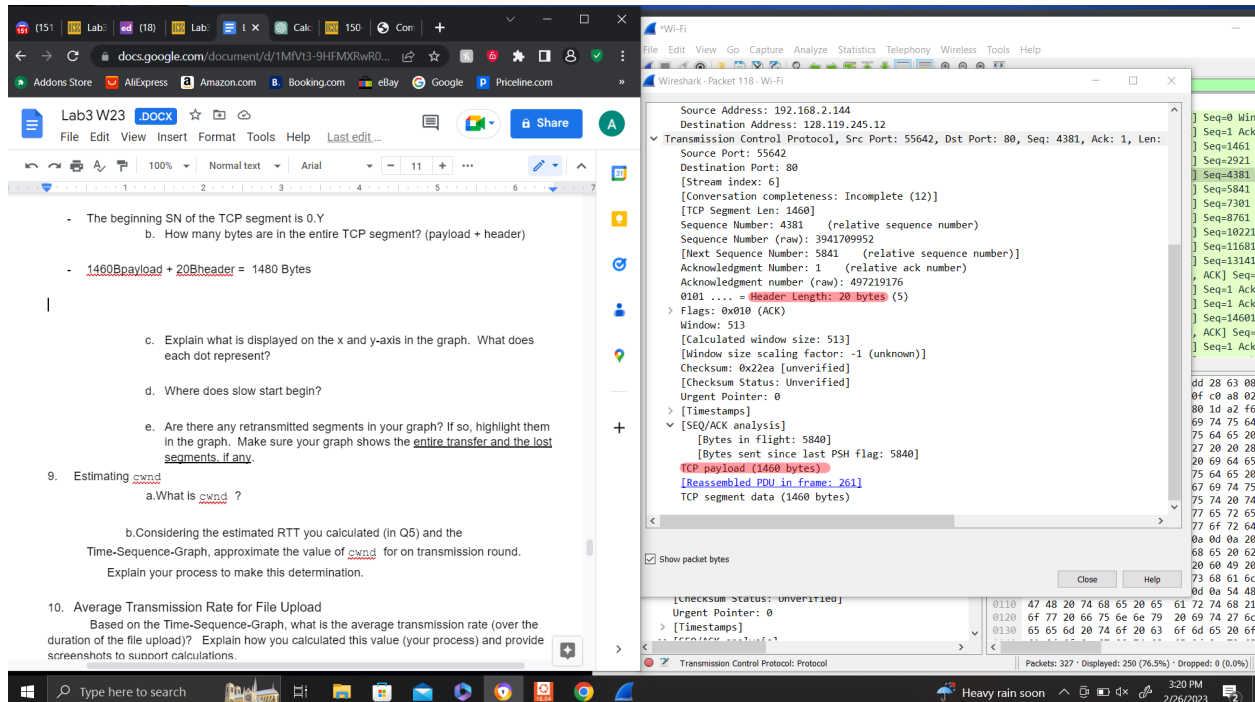
- The receiver typically receives 1460 bytes. This is determined by subtracting the sequence number in the TCP header of the ACK packet from the Acknowledgment Number field value.

TCP Congestion Control

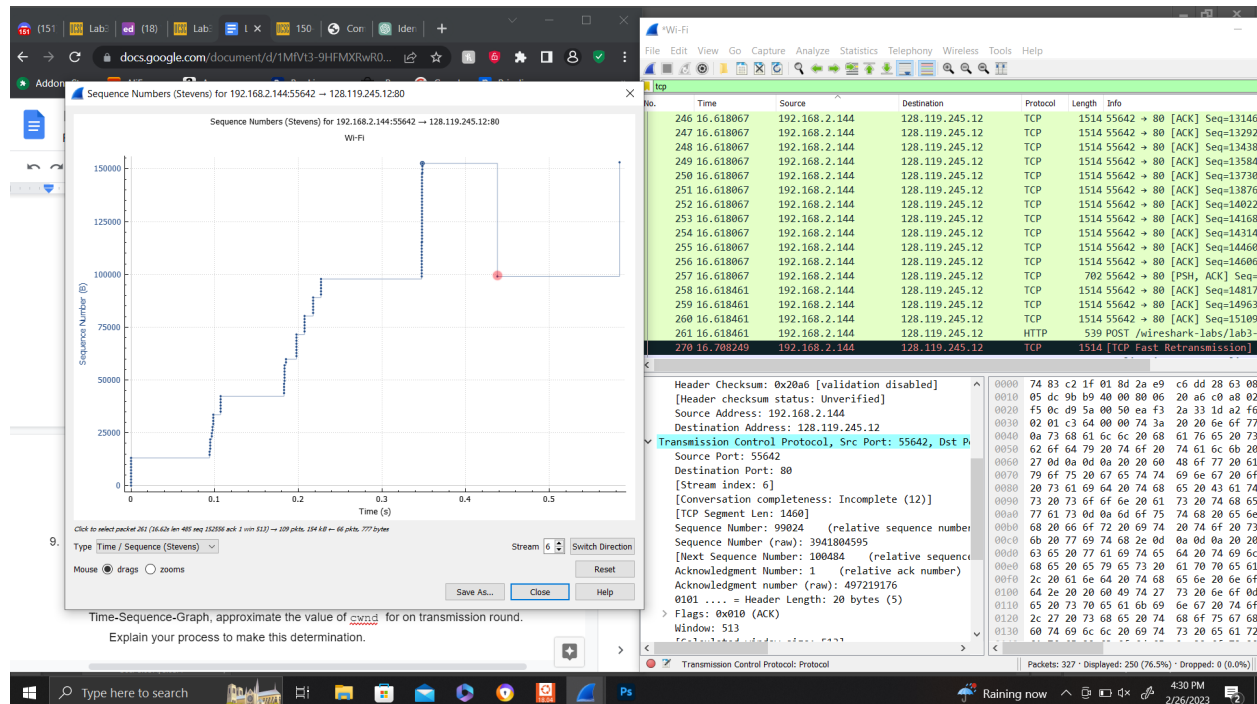
8. Time-Sequence-Graph(Stevens): Select a TCP packet with len != 0 that has been transmitted by the client and use the Time-Sequence-Graph(Stevens) plotting tool to create a graph.

- What is the beginning SN of the TCP Segment you have chosen?

- The beginning SN of the TCP segment is 0.Y
 - b. How many bytes are in the entire TCP segment? (payload + header)
- $1460\text{Bpayload} + 20\text{Bheader} = 1480 \text{ Bytes}$



- c. Explain what is displayed on the x and y-axis in the graph. What does each dot represent?
 - The x axis represents the time the packet was sent or received. The y axis represents the sequence number of the packets. Each dot on the graph represents a single packet.
- d. Where does slow start begin?
 - The slow start algorithm starts at the beginning of the TCP connection. This can be seen in the change in window size after the SYN packet.
- e. Are there any retransmitted segments in your graph? If so, highlight them in the graph. Make sure your graph shows the entire transfer and the lost segments, if any.
 - There is one retransmitted packet with the sequence number 99024.



9. Estimating $cwnd$

a. What is $cwnd$?

- $cwnd$ stands for congested window size. It is a mechanism used by TCP to control the amount of data that can be transmitted across a network at any given time. It is a dynamic value that determines the number of bytes that can be sent before an acknowledgment is received.

b. Considering the estimated RTT you calculated (in Q5) and the Time-Sequence-Graph, approximate the value of $cwnd$ for one transmission round.

Explain your process to make this determination.

- $cwnd = (\text{window size}) + \text{bytes acknowledged during one RTT}$

$$= 513 + 1514 = 2027$$

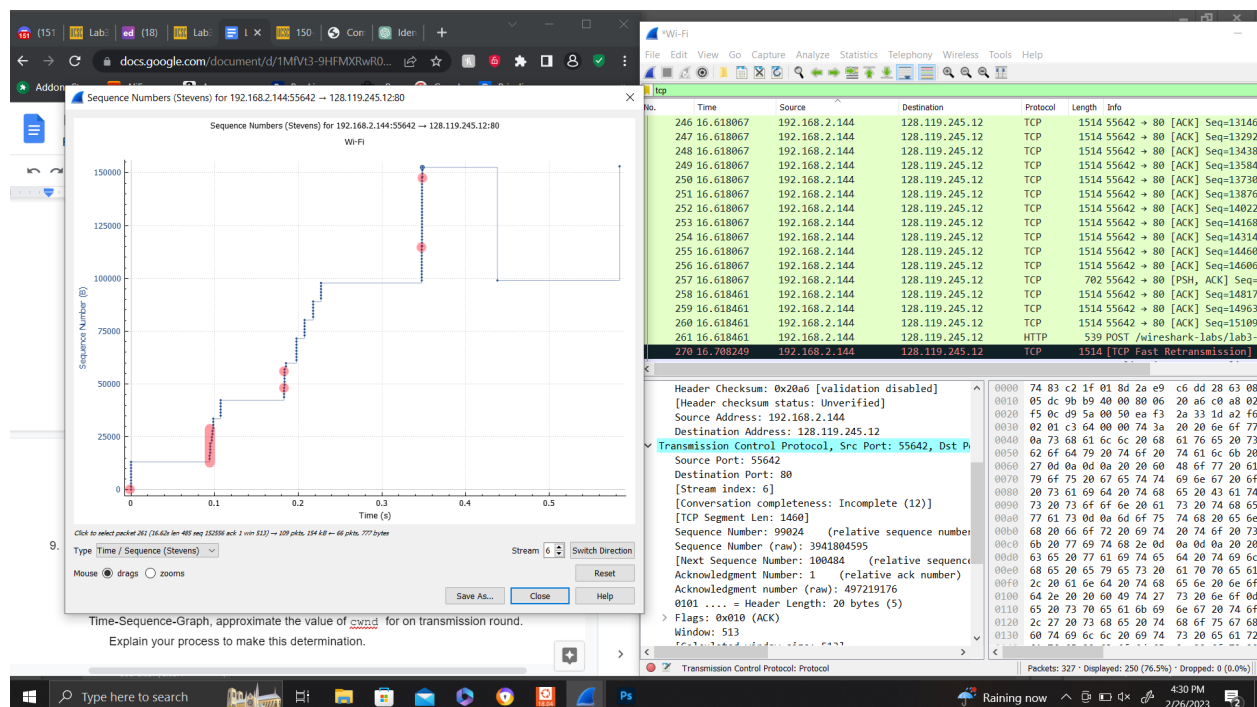
10. Average Transmission Rate for File Upload

Based on the Time-Sequence-Graph, what is the average transmission rate (over the duration of the file upload)? Explain how you calculated this value (your process) and provide screenshots to support calculations.

From the slides in lecture the transmission rate = $cwnd/RTT$.

11. Ideally we would look at a graph of $cwnd$ in order to determine whether a connection is in the slow start or congestion avoidance phase; however our graph is SN over time.

- a. Think about the difference between the two graphs, and see if you can observe any periods of slow start on your Time-Sequence graph. If so, where are they? Mark these periods on your screenshot.



- b. Referring to the lecture slide “TCP Overview”, which of the features of TCP on the slide do you notice in this exercise?

- From the features of TCP slide, in this exercise I noticed:
- Point-to-Point, Reliable, in-order byte

stream, pipelined, send
& receive buffers,
Connection-oriented:,
Flow and congestion
controlled.

- c. Is this upload an example of
a Client-Server or
Peer-to-peer transfer?
 - This upload is an
example of a
Client-Server transfer.

B. Downloading a File from a Server (with packet loss)

Environment: Permissions can be an issue and you should use the **VM** for these problems.

In this section we will examine how loss affects a TCP connection. Loss will be simulated with the `tc qdisc` command to apply a loss rate to an incoming interface.

No introduced losses:

- 12. In this problem we will not intentionally introduce any loss. Start Wireshark and use `wget` to download the file at: <http://ipv4.download.thinkbroadband.com/50MB.zip>

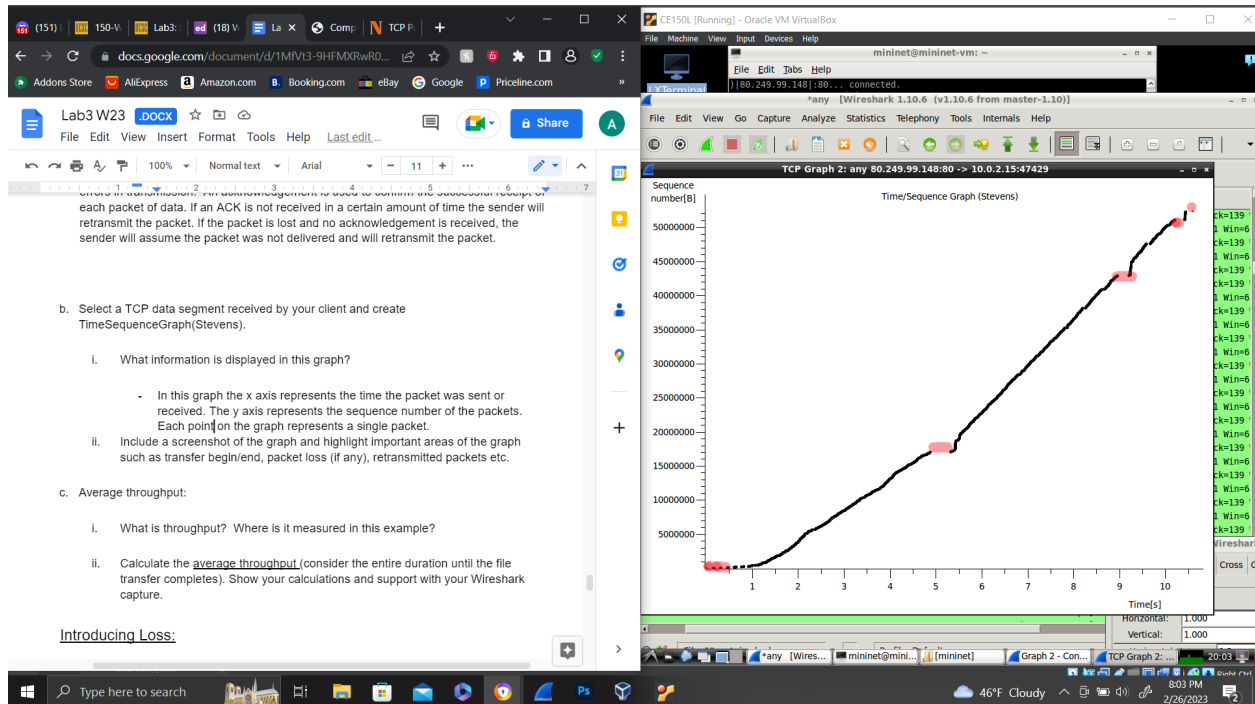
- a. Can packet loss still occur during this download? Why or why not?

- Yes, packet loss can still occur even though TCP is designed to be a reliable protocol for transmitting data. Packet loss can occur for reasons such as network congestion, or errors in transmission. An acknowledgement is used to confirm the successful receipt of each packet of data. If an ACK is not received in a certain amount of time the sender will retransmit the packet. If the packet is lost and no acknowledgement is received, the sender will assume the packet was not delivered and will retransmit the packet.

- b. Select a TCP data segment received by your client and create `TimeSequenceGraph(Stevens)`.

- i. What information is displayed in this graph?

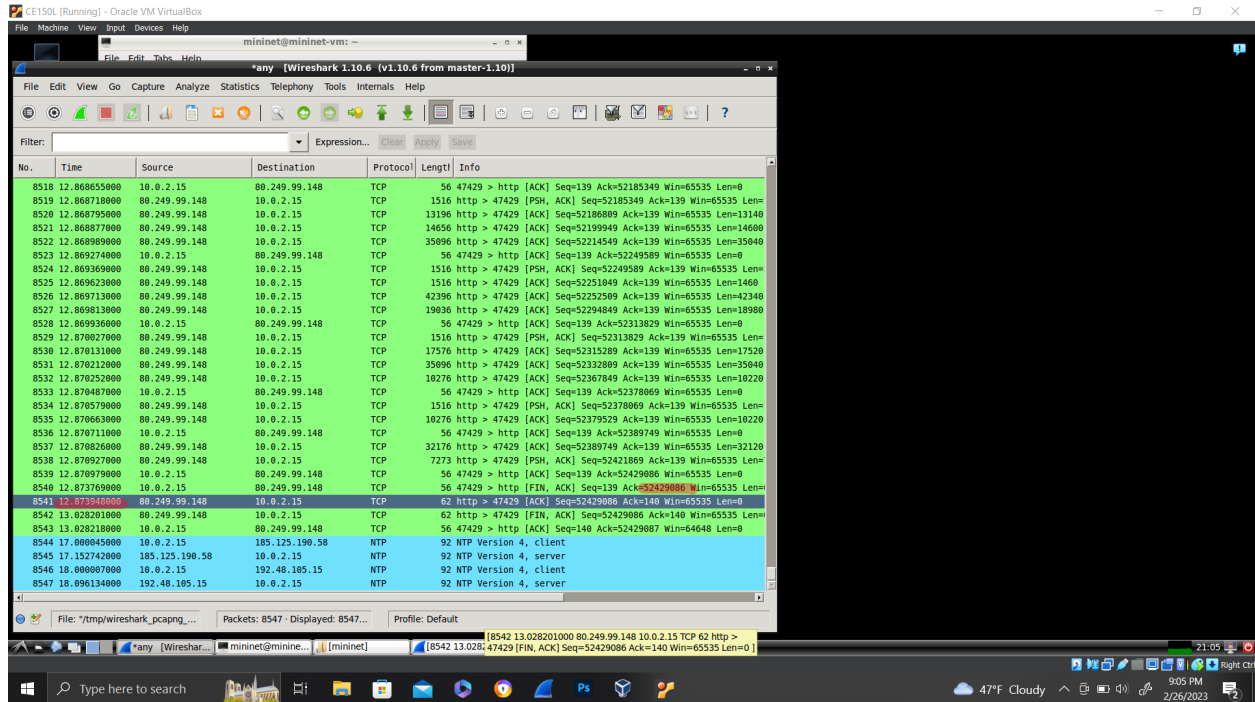
- In this graph the x axis represents the time the packet was sent or received. The y axis represents the sequence number of the packets. Each point on the graph represents a single packet.
- ii. Include a screenshot of the graph and highlight important areas of the graph such as transfer begin/end, packet loss (if any), retransmitted packets etc.
- highlighted are the beginning and end of the transfer and the spaces of packet loss which also indicated packets were retransmitted.



c. Average throughput:

- i. What is throughput? Where is it measured in this example?
- Throughput is the amount of data that can be transmitted through a network connection over a given period of time. In this example it is measured by dividing the total size of the packets by the time difference between the beginning and end of the download.
 - ii. Calculate the average throughput (consider the entire duration until the file transfer completes). Show your calculations and support with your Wireshark capture.
 - The download started at time = 2.266337 and ended at time = 12.873948. Therefore the time to download the 52,429,086 bytes, indicated by the

ACK value at the end of the download is. $52,429,086B / (12.873948 - 2.266337) = 4,942,591.31$ Bytes/second



Introducing Loss:

Now we will simulate loss on an interface using the command `tc qdisc`. When the command is first used, you must use ***add dev*** for the interface being changed. It only needs to be set on the sender's side. After adding the device, use ***change dev*** to set the loss rate.

Follow the commands below to simulate loss on eth0:

- `sudo tc qdisc add dev eth0 root netem loss 0%`
- **Change loss to 100%** `sudo tc qdisc change dev eth0 root netem loss 100%`
- **Change loss back to 0%** `sudo tc qdisc change dev eth0 root netem loss 0%`

100% loss event:

Read through this paragraph before starting the next step:

- First start Wireshark
- then open 2 terminals and have these commands typed and ready before you begin:
 - In one terminal, download the [10MB.zip](#) file using wget
 - While the download is in progress, change loss to 100%. After a second, change loss to 0%.

13. Answer the following questions based on the Wireshark. Take a screenshot and highlight the requested information below:

Receiver's Window Size

- a) What does the window size indicate in the packet header indicate about a Receiver?
- The window size indicates the receiver's current capacity to receive and buffer data from the sender.
- b) What is the initial window size advertised by your Client?
- The initial window size advertised by my client is 29200 bytes.
- c) What was the initial window size advertised by the Server?
- The initial window size advertised by the server is 65535 bytes.

Lab3 W23.docx

13. Answer the following questions based on the Wireshark. Take a screenshot and highlight the requested information below:

Receiver's Window Size

a) What does the window size indicate in the packet header indicate about a Receiver?

- The window size indicates the receiver's current capacity to receive and buffer data from the sender.

b) What is the initial window size advertised by your Client?

- The initial window size advertised by my client is 29200 bytes.

c) What was the initial window size advertised by the Server?

- The initial window size advertised by the server is 65535 bytes.

Find a TCP data segment received by your client and create a TimeSequenceGraph(Stevens) graph with this packet selected.

d) Do you expect that any two students in class would have exactly the same graph? Why or why not? What would be different?

e) At what time does packet loss begin? At what time does it end? Attach a screenshot and highlight the region where 100% loss begins and ends in the graph.

f) Examine the areas in the graph at the beginning of the data transfer and aft

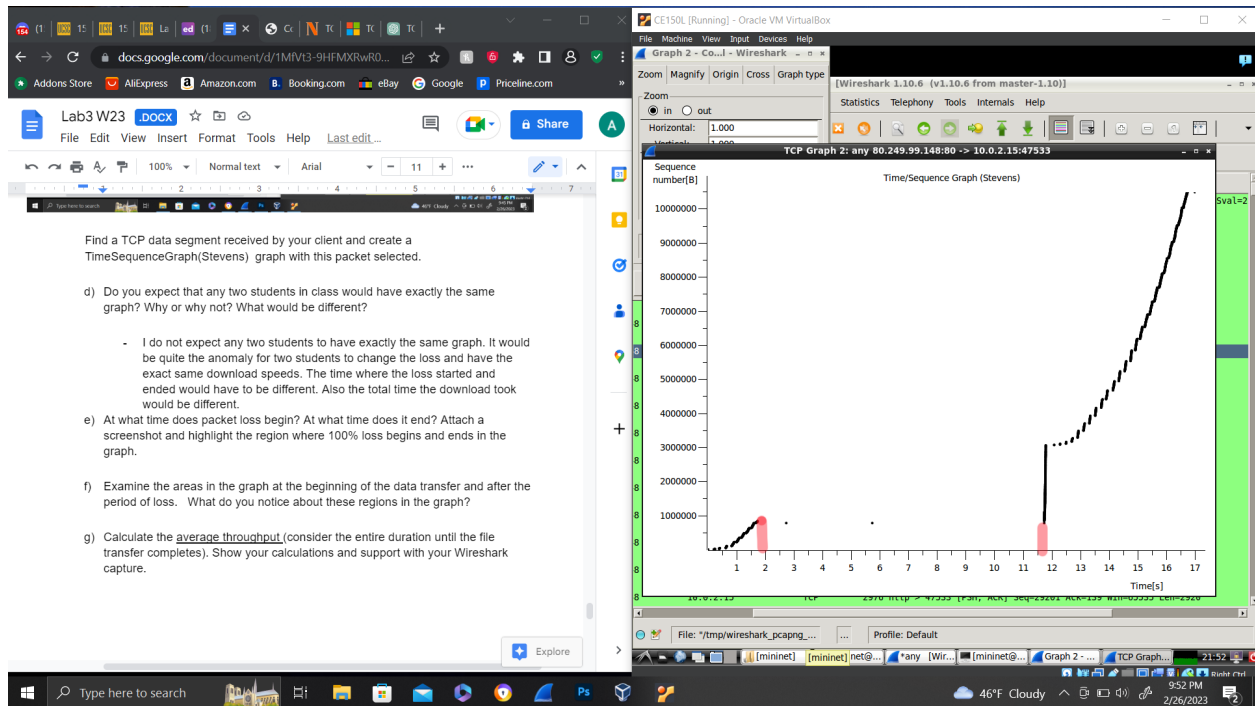
Wireshark capture details:

No.	Time	Source	Destination	Protocol	Length	Info
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	76	47533 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	62	47533 > 47533 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	56	47533 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	194	47533 > http [PSH, ACK] Seq=1 Ack=1 Win=29200 Len=138
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	62	47533 > 47533 [ACK] Seq=1 Ack=139 Win=65535 Len=0
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	2976	http > 47533 [ACK] Seq=1 Ack=139 Win=65535 Len=2920
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	56	47533 > http [ACK] Seq=139 Ack=2921 Win=35840 Len=0
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	1516	http > 47533 [PSH, ACK] Seq=2921 Ack=139 Win=65535 Len=1400
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	56	47533 > http [ACK] Seq=139 Ack=4381 Win=37960 Len=0
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	1516	http > 47533 [PSH, ACK] Seq=4381 Ack=139 Win=65535 Len=1400
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	56	47533 > http [ACK] Seq=139 Ack=5841 Win=48880 Len=0
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	1516	http > 47533 [PSH, ACK] Seq=5841 Ack=139 Win=65535 Len=1400
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	1516	http > 47533 [PSH, ACK] Seq=7301 Ack=139 Win=65535 Len=1400
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	56	47533 > http [ACK] Seq=139 Ack=8761 Win=46720 Len=0
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	1516	http > 47533 [PSH, ACK] Seq=8761 Ack=139 Win=65535 Len=1400
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	56	47533 > http [ACK] Seq=139 Ack=10221 Win=49640 Len=0
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	1516	http > 47533 [PSH, ACK] Seq=10221 Ack=139 Win=65535 Len=1400
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	56	47533 > http [ACK] Seq=139 Ack=11681 Win=52560 Len=0
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	1516	http > 47533 [PSH, ACK] Seq=11681 Ack=139 Win=65535 Len=1400
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	56	47533 > http [ACK] Seq=139 Ack=13141 Win=55480 Len=0
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	1516	http > 47533 [PSH, ACK] Seq=13141 Ack=139 Win=65535 Len=1400
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	56	47533 > http [ACK] Seq=139 Ack=14601 Win=58400 Len=0
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	10276	http > 47533 [PSH, ACK] Seq=14601 Ack=139 Win=65535 Len=10220
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	56	47533 > http [ACK] Seq=139 Ack=24821 Win=64240 Len=0
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	2976	http > 47533 [PSH, ACK] Seq=24821 Ack=139 Win=65535 Len=2920
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	56	47533 > http [ACK] Seq=139 Ack=27741 Win=64240 Len=0
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	1516	http > 47533 [PSH, ACK] Seq=27741 Ack=139 Win=65535 Len=1400
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	56	47533 > http [ACK] Seq=139 Ack=29201 Win=64240 Len=0
8	0.0.2.15	80.249.99.148	10.0.2.15	TCP	2976	http > 47533 [PSH, ACK] Seq=29201 Ack=139 Win=65535 Len=2920

Find a TCP data segment received by your client and create a TimeSequenceGraph(Stevens) graph with this packet selected.

- d) Do you expect that any two students in class would have exactly the same graph? Why or why not? What would be different?
- I do not expect any two students to have exactly the same graph. It would be quite the anomaly for two students to change the loss and have the exact same download speeds. The time where the loss started and ended would have to be different. Also the total time the download took would be different.
- e) At what time does packet loss begin? At what time does it end? Attach a screenshot and highlight the region where 100% loss begins and ends in the graph.

Packet loss begins just before two seconds into the download and ends at 11.5 second into the total time of the download.



- f) Examine the areas in the graph at the beginning of the data transfer and after the period of loss. What do you notice about these regions in the graph?
- g) Calculate the average throughput (consider the entire duration until the file transfer completes). Show your calculations and support with your Wireshark capture.
- The download started at time = 7.579347sec and ended at time = 24.52875. Therefore the time to download the 10,486,045 bytes, indicated by the ACK value at the end of the download is. $10,486,045\text{B} / (24.52875 - 7.579347) = 618,667.513 \text{ Bytes/second}$

Lab3 W23

1) Examine the areas in the graph at the beginning of the data transfer and after the period of loss. What do you notice about these regions in the graph?

g) Calculate the average throughput (consider the entire duration until the file transfer completes). Show your calculations and support with your Wireshark capture.

- The download started at time = 7.579347sec and ended at time = 24.52875. Therefore the time to download the 10,486,045 bytes, indicated by the ACK value at the end of the download is. $10,486,045B / (24.52875 - 7.579347) = 618,667.513$ Bytes/second

20% loss event:

14. Restart Wireshark and retry the experiment, this time with loss 20% for a second and then return to loss 0%. Find a TCP data segment received by your client and create a TimeSequenceGraph(Stevens) graph with this packet selected. TCAttach a screenshot of the trace and highlight the periods of loss.

20% loss event:

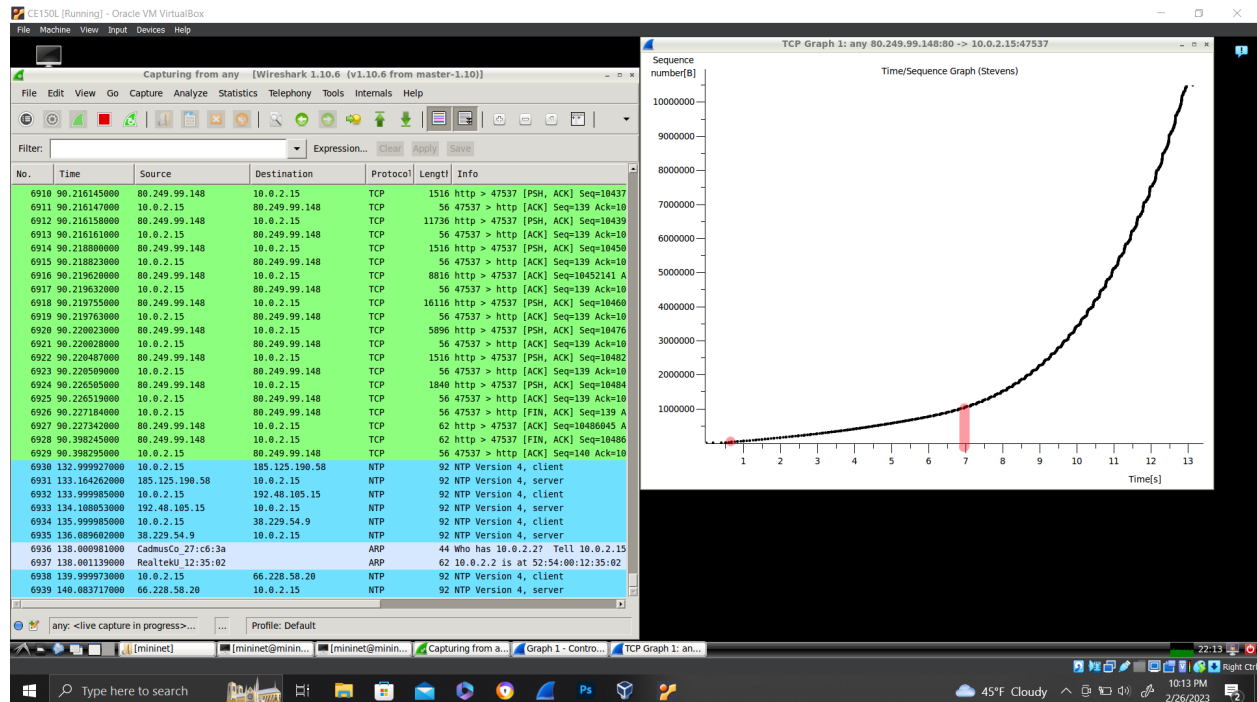
14. Restart Wireshark and retry the experiment, this time with loss 20% for a second and then return to loss 0%. Find a TCP data segment received by your client and create a TimeSequenceGraph(Stevens) graph with this packet selected. TCAttach a screenshot of the trace and highlight the periods of loss.

a. Mark a few times when packet loss begins and ends? At what time does it end? Attach a screenshot and highlight the region where 20% loss begins and ends in the graph.

- The packet loss begins at around a half second into the download and ends around seven seconds into the download.

20% loss event:

14. Restart Wireshark and retry the experiment, this time with loss 20% for a second and then return to loss 0%. Find a TCP data segment received by your client and create a TimeSequenceGraph(Stevens) graph with this packet selected. TCAttach a screenshot of the trace and highlight the periods of loss.
- a. Mark a few times when packet loss begins and ends? At what time does it end? Attach a screenshot and highlight the region where 20% loss begins and ends in the graph.
- The packet loss begins at around a half second into the download and ends around seven seconds into the download.



b. Examine the areas in the graph at the beginning of the data transfer and after the period of loss. What do you notice about these regions in the graph?

- These areas are exponentially increasing in the sequence number vs. time.
- c. Calculate the average throughput (consider the entire duration until the file transfer completes). Show your calculations and support with your Wireshark capture.
- The download started at time = 8.482596sec and ended at time = 90.227342sec. Therefore the time to download the 10,486,045 bytes, indicated by the ACK value at the end of the download is. $10,486,045\text{B} / (90.227342 - 8.48259) = 128,277.899$ Bytes/second

Comparison of TCP Performance:

15. Make a table comparing the loss rates in the exercises above, your throughput calculation and the total delay to download the 10MB file.

Loss Rate	Throughput	Total Delay
No Loss	4,942,591 Bytes/sec	10.61sec

100%	618,678 Bytes/sec	16.94sec
20%	128,278 Bytes/sec	81.74sec

Compare your throughput results for no introduced loss, 100% loss and 20% loss. Do your results make sense to you? What does the difference in throughput tell you about the operation of TCP during periods of congestion or lossy links?

- The results make sense. The short 100 percent loss did not affect the throughput and overall download as much as the longer 20 percent loss period. This tells me that during periods of loss TCP reduces the sending rate through congestion control to lower congestion.

UDP Performance:

16. Assume UDP was used for this packet transfer.

- a) What is a fundamental difference between UDP segment transmission compared to TCP?
 - The fundamental difference between UDP segment transmission compared to TCP is that UDP provides minimal reliability and error correction, while TCP provides reliable transmission with error correction, flow control and congestion control.
- b) In the presence of packet loss, how do you expect the average end-to-end delay to download the file to be affected? Explain your reasoning.
 - I expect the average end to end delay to download the file to increase. UDP does not provide error correction or flow control. Therefore, if a packet is lost UDP will not detect it and the sender will not retransmit it. So the receiver will have to request them again.
- c) How does the UDP sender behave if there is a packet loss during the file transfer?
 - If packet loss occurs during the file transfer the UDP sender will not be aware of the loss and will not automatically resend the packet.
- d) From the UDP sender's perspective, does time to deliver the file change when there is packet loss? Explain your reasoning.

- Yes the time to deliver the packet changes when there is packet loss. If there is packet loss the the sender will be unaware, so to complete the actual delivery of the file the packets would have to be requested again.

e) What does the UDP receiver do about packet loss?

- In UDP the receiver doesn't have any standard protocol to handle packet loss. The receiver will not automatically notify the sender that the packet has not been received. The receiver has to rely on some other layer of the protocol or application to handle packet loss.