Nattapat Yuvasuta	59070501028
Nuttawut Kuadplod	59070501029
Niti Buesamae	59070501047
Panumas Wongsaeng	59070501056
Supadet Jomthepmala	59070501069

Part II : Transmitting Data with TCP

Include relevant parts of the Wireshark data to support each of your answers.

2.1 How many TCP segments (both control and data) are exchanged between PC1 and PC2 ?

Ans. 30

2.2 What are the sizes of the data segments in Question 2.1? Are they all the same ? If not, explain why they have different sizes.

Ans. 1024,1448(6 blocks),528(3 blocks). They aren't same because it can't send data 10*1024 = 10240 in 10 time then the amount of data 1024+6*1448+3*528 = 10240 (byte stream service)

2.3 How many segments are TCP control segments? What are they (look at the Flag field) ?

Ans. 10 segments there are Reserved, nonce, congestion, ECN-echo, Urgent , ACK, Push, Reset, Syn, Fin

2.4 Compare the total number of bytes transmitted, in both directions, including Ethernet, IP, and TCP headers, to the amount of application data transmitted.

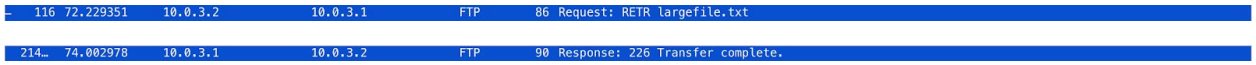
Ans. 66

Nattapat Yuvasuta	59070501028
Nuttawut Kuadplod	59070501029
Niti Buesamae	59070501047
Panumas Wongsaeng	59070501056
Supadet Jomthepmala	59070501069

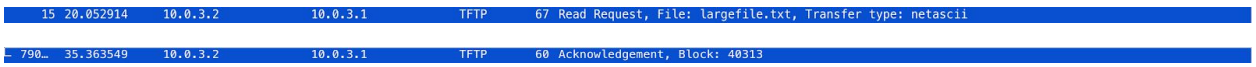
Part III: File Transfers Using TCP and UDP

3.1 From the timestamps recorded by Wireshark, obtain the times it took to transfer the large file with FTP and with TFTP. Which one, FTP or TFTP, transfers the file faster? Look at the Wireshark data and use your knowledge of FTP, TFTP, TCP, and UDP to explain the outcome.

FTP

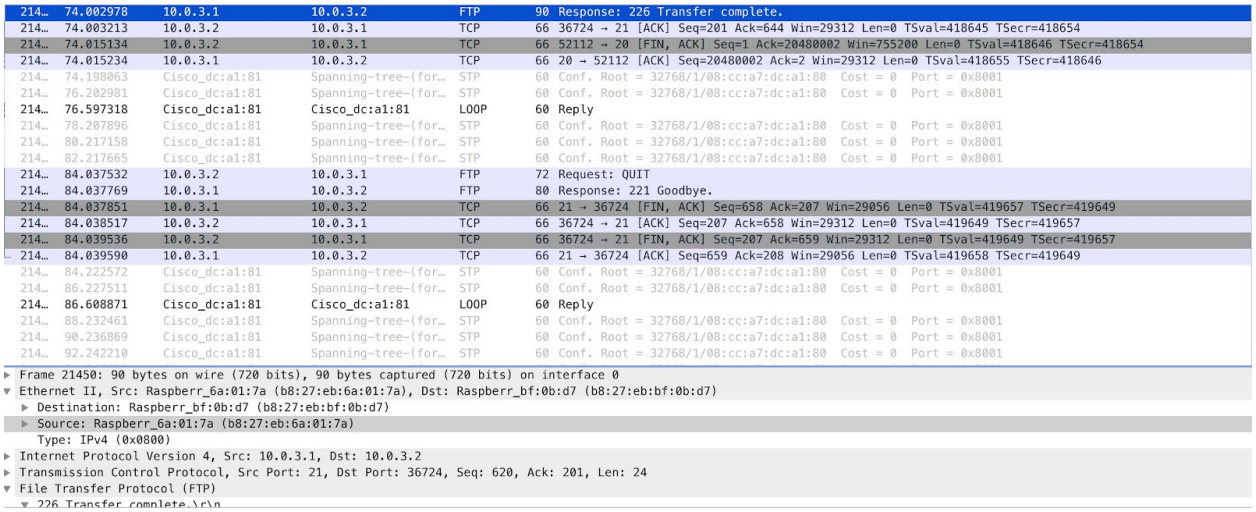


TFTP



Ans. FTP เร็วกว่า เนื่องจากว่า FTP ใช้โปรโตคอล TCP ที่มีการสร้างช่องทางการรับส่งข้อมูล ทำให้การันตีว่าข้อมูลจะถูกส่งไปยังผู้รับอย่างแน่นอน และลำดับข้อมูลอีกด้วย แต่ TFTP ใช้ โปรโตคอล UDP ซึ่งไม่มีการสร้างช่องทางการรับส่งข้อมูล ทำให้ไม่มีการการันตีว่าข้อมูลที่ส่งไปนั้นจะไปถึงยังผู้รับ อีกทั้งยังไม่มีการลำดับข้อมูลด้วย เมื่อข้อมูลสูญหายระหว่างทางทำให้ TFTP ต้องส่งข้อมูลใหม่ทั้งหมด เนื่องจากว่าไม่รู้ว่าข้อมูลส่วนไหนสูญหาย จึงทำให้ TFTP โอนถ่ายข้อมูลช้ากว่า FTP

3.2 How many parallel connections were created in the FTP session? What are those connections for? What are the source/destination port numbers of those connections?



Ans. จำนวน 2 session คือ Control connection(Passive mode) สำหรับสร้างช่องทางการรับส่งข้อมูล และ Data connection(Active mode) สำหรับการรับส่งข้อมูล โดย source ใช้พอร์ต 21 และ destination ใช้พอร์ต 36724

Nattapat Yuvasuta	59070501028
Nuttawut Kuadplod	59070501029
Niti Buesamae	59070501047
Panumas Wongsaeng	59070501056
Supadet Jomthepmala	59070501069

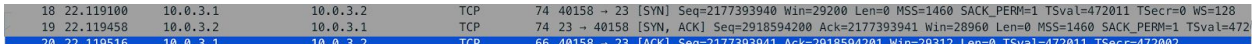
Part IV: TCP Connection Management

Use the saved Wireshark output to answer the following questions. Include relevant parts of the Wireshark data to support each of your answers.

4.1 Identify the segments of the three-way handshake. Which flags are set in the TCP headers? Explain how these flags are interpreted by the receiving TCP server or TCP client.

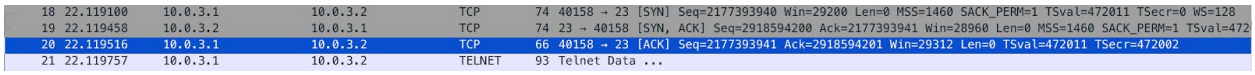
Ans. SYN and Ack are set in the TCP headers
PC1 **sends** a TCP **SYN**chronize packet to PC2
PC2 receives PC1's **SYN**
PC1 receives PC2's **SYN-ACK**
PC1 sends **ACK**nowledge
PC2 receives **ACK.**

4.2 During the connection setup, the TCP client and TCP server tell each other the first sequence number they will use for data transmission. What are the initial sequence numbers of the TCP client and the TCP server? In Wireshark, you need to go to Preferences ➔ Protocols ➔ TCP and uncheck the box that says “Relative Sequence Number” before answering this question.



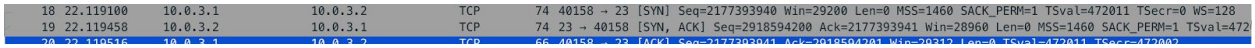
Ans. Sequence number of client is 2177393940 and sequence number of server is 2918594200

4.3 Identify the first segment that contains application data. What is the sequence number used in the first byte of application data sent from the TCP client to the TCP server?



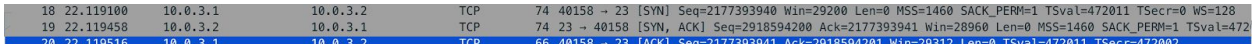
Ans. Sequence number is 2177393941

4.4 The TCP client and TCP server exchange the initial window sizes to get the maximum amount of data that the other side can send at any time. Determine the values of the initial window sizes for the TCP client and the TCP server.



Ans. Window size of client is 29200 and Window size of client is 28960

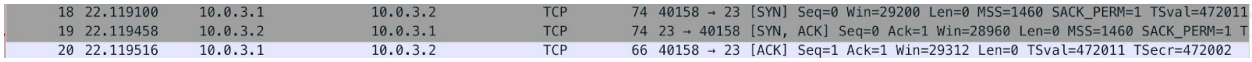
4.5 What is the MSS value that is negotiated between the TCP client and the TCP server?



Ans. MSS value is 1460

Nattapat Yuvasuta	59070501028
Nuttawut Kuadplod	59070501029
Niti Buesamae	59070501047
Panumas Wongsaeng	59070501056
Supadet Jomthepmala	59070501069

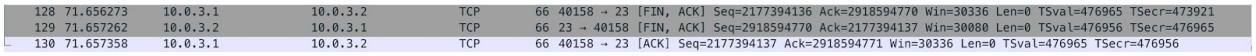
4.6 How long does it take to open the TCP connection?



18	22.119100	10.0.3.1	10.0.3.2	TCP	74	40158 → 23 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=472011
19	22.119458	10.0.3.2	10.0.3.1	TCP	74	23 → 40158 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 T
20	22.119516	10.0.3.1	10.0.3.2	TCP	66	40158 → 23 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=472011 TSecr=472002

Ans. 0.000416 sec.

4.7 In Step 3, identify the packets that are involved in closing the TCP connection. Which flags are set in these packets? Explain how these flags are interpreted by the receiving TCP server or TCP client.

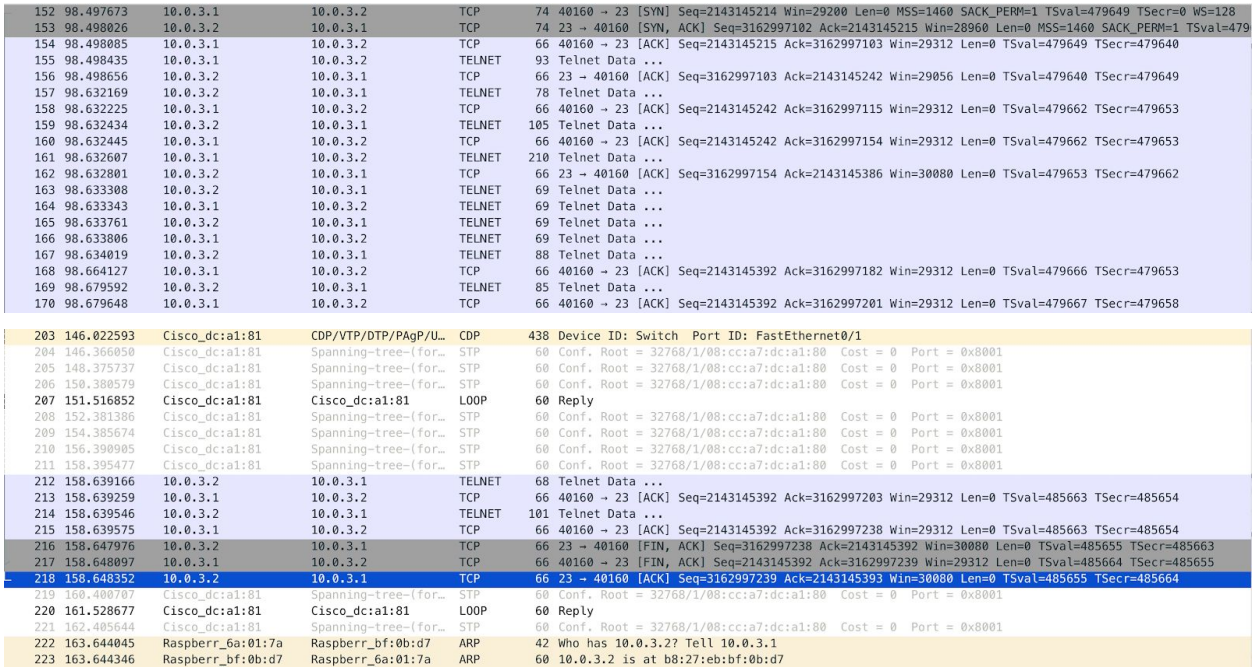


128	71.656273	10.0.3.1	10.0.3.2	TCP	66	40158 → 23 [FIN, ACK] Seq=2177394136 Ack=2918594770 Win=30336 Len=0 TSval=476965 TSecr=473921
129	71.657262	10.0.3.2	10.0.3.1	TCP	66	23 → 40158 [FIN, ACK] Seq=2918594770 Ack=2177394137 Win=30080 Len=0 TSval=476956 TSecr=476965
130	71.657358	10.0.3.1	10.0.3.2	TCP	66	40158 → 23 [ACK] Seq=2177394137 Ack=2918594771 Win=30336 Len=0 TSval=476965 TSecr=476956

Ans. ใช้ FIN และ ACK เริ่มต้นจาก client ทำการส่ง FIN ไปยัง server เพื่อขอร้องให้มีการหยุดรับส่งข้อมูล จากนั้น server จะทำการส่ง FIN+ACK มายัง client เพื่อ confirm ว่าจะหยุดรับส่งข้อมูล และเมื่อ client ต้องการยืนยัน ก็ส่ง ACK ตอบกลับไป เป็นอันเสร็จสิ้นการรับส่งข้อมูล

4.8 Describe how the closing of the connection in Step 4 is different from Step 3. How long does the Telnet server wait until it closes the TCP connection?

Include relevant parts of the Wireshark output to support each of your answers.

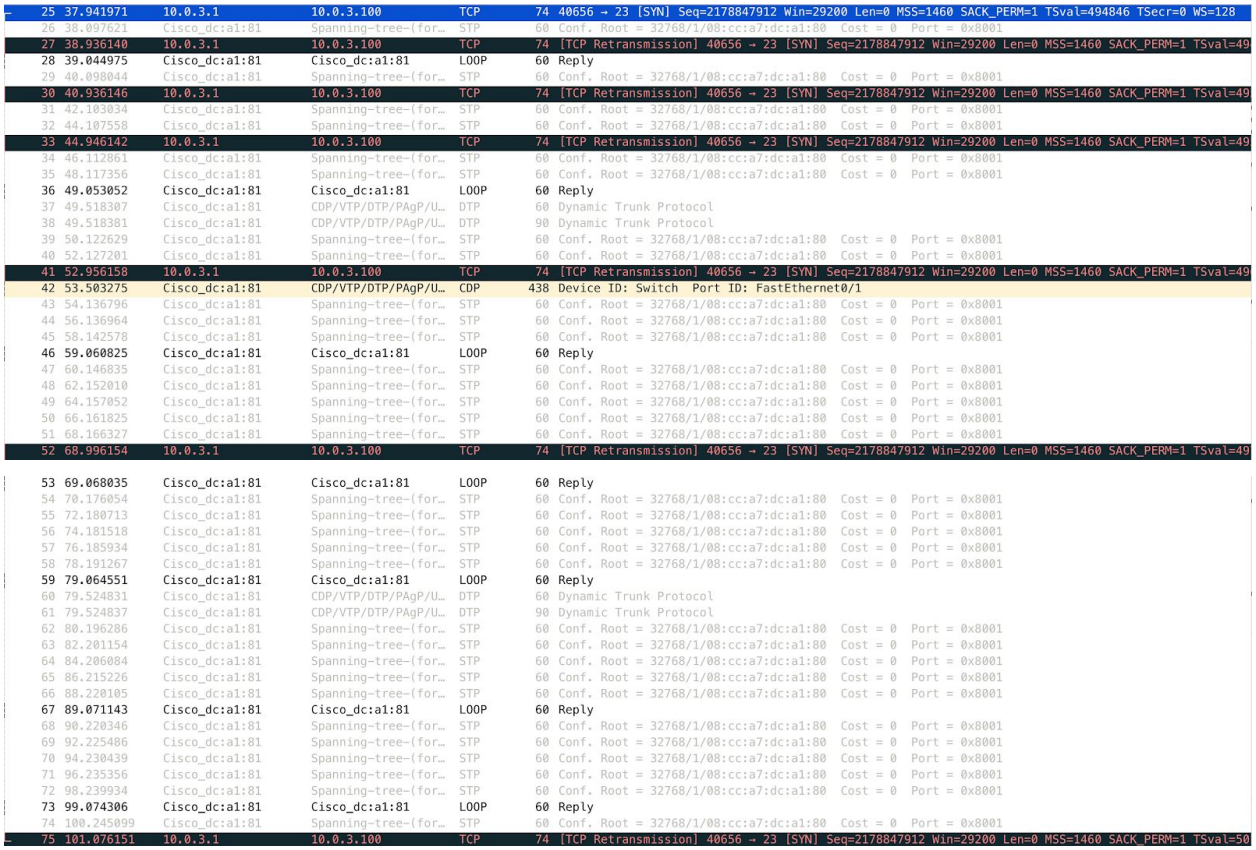


152	98.497673	10.0.3.1	10.0.3.2	TCP	74	40160 → 23 [SYN] Seq=2143145214 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=479649 TSecr=0 WS=128
153	98.498026	10.0.3.2	10.0.3.1	TCP	74	23 → 40160 [SYN, ACK] Seq=3162997102 Ack=2143145215 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=479
154	98.498085	10.0.3.1	10.0.3.2	TCP	66	40160 → 23 [ACK] Seq=2143145215 Ack=3162997103 Win=29312 Len=0 TSval=479649 TSecr=479640
155	98.498435	10.0.3.1	10.0.3.2	TELNET	93	Telnet Data ...
156	98.498656	10.0.3.2	10.0.3.1	TCP	66	23 → 40160 [ACK] Seq=3162997103 Ack=2143145242 Win=29056 Len=0 TSval=479640 TSecr=479649
157	98.632169	10.0.3.2	10.0.3.1	TELNET	78	Telnet Data ...
158	98.632225	10.0.3.1	10.0.3.2	TCP	66	40160 → 23 [ACK] Seq=2143145242 Ack=3162997115 Win=29312 Len=0 TSval=479662 TSecr=479653
159	98.632434	10.0.3.2	10.0.3.1	TELNET	105	Telnet Data ...
160	98.632445	10.0.3.1	10.0.3.2	TCP	66	40160 → 23 [ACK] Seq=2143145242 Ack=3162997154 Win=29312 Len=0 TSval=479662 TSecr=479653
161	98.632607	10.0.3.1	10.0.3.2	TELNET	210	Telnet Data ...
162	98.632801	10.0.3.2	10.0.3.1	TCP	66	23 → 40160 [ACK] Seq=3162997154 Ack=2143145386 Win=30080 Len=0 TSval=479653 TSecr=479662
163	98.633308	10.0.3.2	10.0.3.1	TELNET	69	Telnet Data ...
164	98.633343	10.0.3.1	10.0.3.2	TELNET	69	Telnet Data ...
165	98.633761	10.0.3.2	10.0.3.1	TELNET	69	Telnet Data ...
166	98.633806	10.0.3.1	10.0.3.2	TELNET	69	Telnet Data ...
167	98.634019	10.0.3.2	10.0.3.1	TELNET	88	Telnet Data ...
168	98.664127	10.0.3.1	10.0.3.2	TCP	66	40160 → 23 [ACK] Seq=2143145392 Ack=3162997182 Win=29312 Len=0 TSval=479666 TSecr=479653
169	98.679592	10.0.3.2	10.0.3.1	TELNET	85	Telnet Data ...
170	98.679648	10.0.3.1	10.0.3.2	TCP	66	40160 → 23 [ACK] Seq=2143145392 Ack=3162997201 Win=29312 Len=0 TSval=479667 TSecr=479658
203	146.022593	Cisco_dc:a1:81	CDP/VTP/DTP/PAgP/U…	CDP	438	Device ID: Switch Port ID: FastEthernet0/1
204	146.366050	Cisco_dc:a1:81	Spanning-tree-(for…	STP	60	Conf. Root = 32768/1/08:cc:a7:dc:a1:80 Cost = 0 Port = 0x8001
205	148.375737	Cisco_dc:a1:81	Spanning-tree-(for…	STP	60	Conf. Root = 32768/1/08:cc:a7:dc:a1:80 Cost = 0 Port = 0x8001
206	150.380579	Cisco_dc:a1:81	Spanning-tree-(for…	STP	60	Conf. Root = 32768/1/08:cc:a7:dc:a1:80 Cost = 0 Port = 0x8001
207	151.516852	Cisco_dc:a1:81	Cisco_dc:a1:81	LOOP	60	Reply
208	152.381386	Cisco_dc:a1:81	Spanning-tree-(for…	STP	60	Conf. Root = 32768/1/08:cc:a7:dc:a1:80 Cost = 0 Port = 0x8001
209	154.385674	Cisco_dc:a1:81	Spanning-tree-(for…	STP	60	Conf. Root = 32768/1/08:cc:a7:dc:a1:80 Cost = 0 Port = 0x8001
210	156.390905	Cisco_dc:a1:81	Spanning-tree-(for…	STP	60	Conf. Root = 32768/1/08:cc:a7:dc:a1:80 Cost = 0 Port = 0x8001
211	158.395477	Cisco_dc:a1:81	Spanning-tree-(for…	STP	60	Conf. Root = 32768/1/08:cc:a7:dc:a1:80 Cost = 0 Port = 0x8001
212	158.639166	10.0.3.2	10.0.3.1	TELNET	68	Telnet Data ...
213	158.639259	10.0.3.1	10.0.3.2	TCP	66	40160 → 23 [ACK] Seq=2143145392 Ack=3162997203 Win=29312 Len=0 TSval=485663 TSecr=485654
214	158.639546	10.0.3.2	10.0.3.1	TELNET	101	Telnet Data ...
215	158.639575	10.0.3.1	10.0.3.2	TCP	66	40160 → 23 [ACK] Seq=2143145392 Ack=3162997238 Win=29312 Len=0 TSval=485663 TSecr=485654
216	158.647976	10.0.3.2	10.0.3.1	TCP	66	23 → 40160 [FIN, ACK] Seq=3162997238 Ack=2143145392 Win=30080 Len=0 TSval=485655 TSecr=485663
217	158.648097	10.0.3.1	10.0.3.2	TCP	66	40160 → 23 [FIN, ACK] Seq=2143145392 Ack=3162997239 Win=29312 Len=0 TSval=485664 TSecr=485655
218	158.648352	10.0.3.2	10.0.3.1	TCP	66	23 → 40160 [ACK] Seq=3162997239 Ack=2143145393 Win=30080 Len=0 TSval=485655 TSecr=485664
219	160.400707	Cisco_dc:a1:81	Spanning-tree-(for…	STP	60	Conf. Root = 32768/1/08:cc:a7:dc:a1:80 Cost = 0 Port = 0x8001
220	161.528677	Cisco_dc:a1:81	Cisco_dc:a1:81	LOOP	60	Reply
221	162.405644	Cisco_dc:a1:81	Spanning-tree-(for…	STP	60	Conf. Root = 32768/1/08:cc:a7:dc:a1:80 Cost = 0 Port = 0x8001
222	163.644045	Raspberr_6a:01:7a	Raspberr_bf:0b:d7	ARP	42	Who has 10.0.3.2? Tell 10.0.3.1
223	163.644346	Raspberr_bf:0b:d7	Raspberr_6a:01:7a	ARP	60	10.0.3.2 is at b8:27:eb:bf:0b:d7

Ans. ใน Step 3 นั้น เราทำการสั่งปิด telnet จาก PC1 นั่นคือ PC1 เป็นคนส่ง FIN+ACK มายัง PC2 แต่ใน Step 4 นี้เราทำการสั่งปิด telnel จาก PC1 เช่นกัน ต่างกันที่เราไม่พิมพ์อะไรเลยเมื่อมีหน้าต่างเด้งขึ้นมาให้กรอกข้อมูล ทำให้ PC2 สั่งปิด session จาก PC1 นั่นคือ PC2 เป็นคนส่ง FIN+ACK มายัง PC1 ซึ่งเวลาที่ใช้ในการรอระหว่าง PC2 สั่งปิด session จาก PC1 ประมาณ 1~2 นาที

Nattapat Yuvasuta	59070501028
Nuttawut Kuadplod	59070501029
Niti Buesamae	59070501047
Panumas Wongsaeng	59070501056
Supadet Jomthepmala	59070501069

4.9 How often does the TCP client try to establish a connection? How much time elapses between the repeated attempts to open a connection?

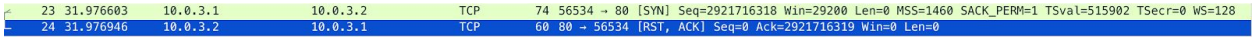


Ans. The TCP client try to establish 6 times. At 1st took 1 sec, 2rd spent 2 sec, 3nd spent 4 sec, 4th spent 8 sec, 5th spent 16 sec and 6th spent 32 sec

4.10 Does the TCP client send out any control segments when it gives up on establishing a connection? Why or Why not?

Ans. No, it doesn't. It just tell that can't connect

4.12 What kind of segment is returned by TCP at PC2 to close this connection? How long does the process of ending the connection take?



Ans. RST+ACK to close this connection. It takes 0.000343 sec to close this connection.

4.13 Suppose you want to know if a certain network process is running at a given host, how would you exploit the knowledge of TCP connection establish to do it?

Ans. Send SYN ,If server return SYN,ACK it's mean the port is opened but if it return RST,ACK it's mean the port is closed.

Nattapat Yuvasuta	59070501028
Nuttawut Kuadplod	59070501029
Niti Buesamae	59070501047
Panumas Wongsaeng	59070501056
Supadet Jomthepmala	59070501069

Part V: TCP Bulk Data Transfer

Include relevant parts of the Wireshark data to support each of your answers.

5.1 How frequent does the receiver send ACKs? Determine and explain the rule used by TCP to send ACKs.

Ans. They are some rule and delay of ACK segment send. Data from PC1 send very fast, so the ACK is later after the data has already been received.

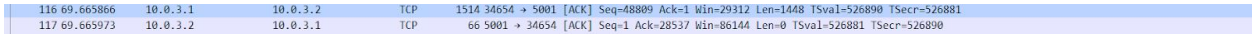
ACK segment will ask for the next Sequence number that is not yet received. For example. If PC1 send sequence 1 with size 500, PC2 will send back ACK 501.

5.2 How many bytes of data does the receiver acknowledge in a typical ACK? What is the largest amount of data acknowledged in a single ACK?

Ans. Typical ACK acknowledge 944,1024, 1448 bytes data (and much more).

Biggest TCP payload of 1514

ACK for sequence 28537 from seq. 48809

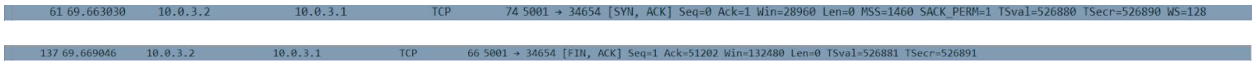


5.3 What is the maximum and minimum window size advertised by the receiver? How does the window size vary during the lifetime of the TCP connection?

Ans. Window size increase when successful sending data without error. And decrease when congestion occur.

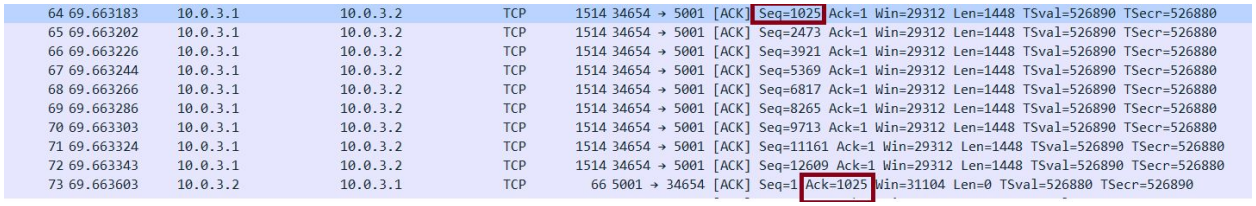
Minimum of 28960 (when SYN ACK).

Maximum of 132480 (when receive the last segment).



5.4 Select an arbitrary ACK segment sent by PC2 to PC1 and relate it to a segment sent by PC1. How long did it take from the transmission of the segment until the ACK arrives at PC1?

Ans. The delay is approximately 0.00042 second.



5.5 Does the TCP sender generally transmit the maximum amount of data allowed by the advertised window size? Explain.

Ans. No, because it is limited by the Maximum Segment Size.

Nattapat Yuvasuta	59070501028
Nuttawut Kuadplod	59070501029
Niti Buesamae	59070501047
Panumas Wongsaeng	59070501056
Supadet Jomthepmala	59070501069

5.6 After the TCP sender has sent all its data, what segment does it send?

Ans. After all ACKs received segments, receiver send the statistic of transmission rate with PSH ACK flag back to PC1. Then Close the connection by sending FIN ACK segment.