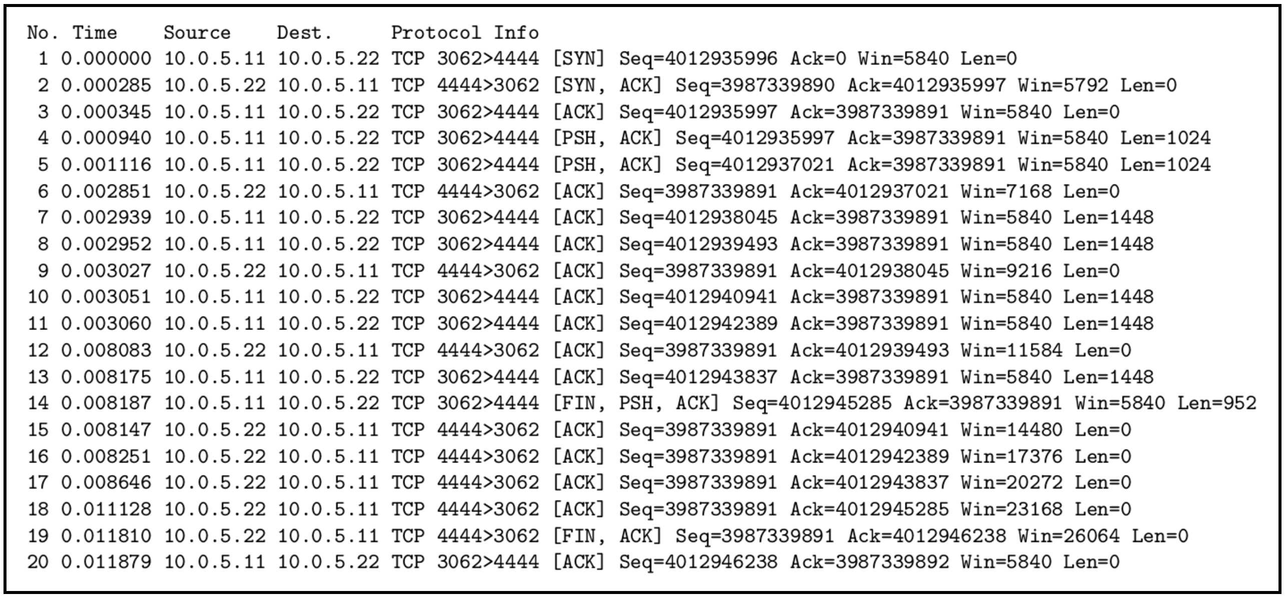
A Wireshark capture of synthetic TCP traffic between PC1 (10.0.5.11/24) the client, and PC2 (10.0.5.22/24) the server, follows. Use it to answer the following questions.



```
No. Time     Source    Dest.     Protocol Info
 1 0.000000 10.0.5.11 10.0.5.22 TCP 3062>4444 [SYN] Seq=4012935996 Ack=0 Win=5840 Len=0
 2 0.000285 10.0.5.22 10.0.5.11 TCP 4444>3062 [SYN, ACK] Seq=3987339890 Ack=4012935997 Win=5792 Len=0
 3 0.000345 10.0.5.11 10.0.5.22 TCP 3062>4444 [ACK] Seq=4012935997 Ack=3987339891 Win=5840 Len=0
 4 0.000940 10.0.5.11 10.0.5.22 TCP 3062>4444 [PSH, ACK] Seq=4012935997 Ack=3987339891 Win=5840 Len=1024
 5 0.001116 10.0.5.11 10.0.5.22 TCP 3062>4444 [PSH, ACK] Seq=4012937021 Ack=3987339891 Win=5840 Len=1024
 6 0.002851 10.0.5.22 10.0.5.11 TCP 4444>3062 [ACK] Seq=3987339891 Ack=4012937021 Win=7168 Len=0
 7 0.002939 10.0.5.11 10.0.5.22 TCP 3062>4444 [ACK] Seq=4012938045 Ack=3987339891 Win=5840 Len=1448
 8 0.002952 10.0.5.11 10.0.5.22 TCP 3062>4444 [ACK] Seq=4012939493 Ack=3987339891 Win=5840 Len=1448
 9 0.003027 10.0.5.22 10.0.5.11 TCP 4444>3062 [ACK] Seq=3987339891 Ack=4012938045 Win=9216 Len=0
10 0.003051 10.0.5.11 10.0.5.22 TCP 3062>4444 [ACK] Seq=4012940941 Ack=3987339891 Win=5840 Len=1448
11 0.003060 10.0.5.11 10.0.5.22 TCP 3062>4444 [ACK] Seq=4012942389 Ack=3987339891 Win=5840 Len=1448
12 0.008083 10.0.5.22 10.0.5.11 TCP 4444>3062 [ACK] Seq=3987339891 Ack=4012939493 Win=11584 Len=0
13 0.008175 10.0.5.11 10.0.5.22 TCP 3062>4444 [ACK] Seq=4012943837 Ack=3987339891 Win=5840 Len=1448
14 0.008187 10.0.5.11 10.0.5.22 TCP 3062>4444 [FIN, PSH, ACK] Seq=4012945285 Ack=3987339891 Win=5840 Len=952
15 0.008147 10.0.5.22 10.0.5.11 TCP 4444>3062 [ACK] Seq=3987339891 Ack=4012940941 Win=14480 Len=0
16 0.008251 10.0.5.22 10.0.5.11 TCP 4444>3062 [ACK] Seq=3987339891 Ack=4012942389 Win=17376 Len=0
17 0.008646 10.0.5.22 10.0.5.11 TCP 4444>3062 [ACK] Seq=3987339891 Ack=4012943837 Win=20272 Len=0
18 0.011128 10.0.5.22 10.0.5.11 TCP 4444>3062 [ACK] Seq=3987339891 Ack=4012945285 Win=23168 Len=0
19 0.011810 10.0.5.22 10.0.5.11 TCP 4444>3062 [FIN, ACK] Seq=3987339891 Ack=4012946238 Win=26064 Len=0
20 0.011879 10.0.5.11 10.0.5.22 TCP 3062>4444 [ACK] Seq=4012946238 Ack=3987339892 Win=5840 Len=0
```

Note that I put an image here for formatting. It is also shown as a table after the questions on the last page. The table might be more helpful.

(a) How can you identify the packets involved in opening the TCP connection? What is the initial sequence number (ISN) of the TCP client and the TCP server (Hint: there is one ISN on each direction)?

The packets involved in opening the TCP connection have flags SYN, SYN + ACK, and ACK, in that order. The ISN of the TCP client is 4012935996 and the ISN of the TCP server is 3987339890, where the first packet has a SYN flag and the second packet has a SYN and ACK flag.

(b) What is the sequence number used in the first byte of application data sent from the TCP client to the TCP server?

The sequence number used in the first byte of the application data sent from the TCP client is 4012935997 in packet 4.

(c) Determine the values of the receiving window sizes for the TCP client and the TCP server. How do they change? Note that TCP is full-duplex, there is a receiving window in each direction.

The receiving window size for the TCP client is 5840 and it does not change as the server never sends information to the client. The receiving window size for the TCP server starts at 5792 and continues to get bigger and bigger as no packets are getting lost, so the server thinks it is capable of receiving more packets. In this case, no packets were ever dropped so the window size continues getting bigger. If a packet was dropped, the window size would drastically decrease to reduce the drop rate in the future.

(d) How many packets are transmitted by PC1 and how many packets are transmitted by PC2? Is there any retransmission of a TCP segment (with actual data)? Are there any duplicate ACKs?

There are 11 packets transmitted by PC1 and 9 packets transmitted by PC2. There was no retransmission of any TCP segment and there were no duplicate ACKs, but some ACK packets were just sent late.

(e) Inspect the TCP headers. How many types of flags do you observe (such as ACK)? What do they mean?

SYN – flag bit used to establish a connection
ACK – flag bit used to indicate that the acknowledgement number is valid
PSH – flag bit used to send the data and not buffer it until a full buffer has been received.
FIN – flag bit used to release a connection, indicating that the sender has no more data to send.

(f) How can you identify the packets that are involved in closing the TCP connection? Which end can initiate the close?

The packets involved in closing the TCP connection have flags FIN, FIN + ACK, then ACK, in that order. We see that packet 14 has the FIN flag, packet 19 has the FIN + ACK flags, and packet 20 has the ACK flag. Either end can initiate the termination of a connection.

(g) What does it mean for the TCP connection to be full duplex?

Full duplex means that either end of the connection can send and receive data. In our example, the client is sending data to the server, and the server is sending data back to the client to inform the client that the data sent from the client has been successfully received. On the other hand, half duplex means that only one end can send data while the other end can only receive the data.

No.	Time	Source	Dest.	Protocol Info	Flags	
1	0.000000	10.0.5.11	10.0.5.22	TCP 3062>4444	[SYN]	Seq=4012935996 Ack=0 Win=5840 Len=0
2	0.000285	10.0.5.22	10.0.5.11	TCP 4444>3062	[SYN,ACK]	Seq=3987339890 Ack=4012935997 Win=5792 Len=0
3	0.000345	10.0.5.11	10.0.5.22	TCP 3062>4444	[ACK]	Seq=4012935997 Ack=3987339891 Win=5840 Len=0
4	0.000940	10.0.5.11	10.0.5.22	TCP 3062>4444	[PSH,ACK]	Seq=4012935997 Ack=3987339891 Win=5840 Len=1024
5	0.001116	10.0.5.11	10.0.5.22	TCP 3062>4444	[PSH,ACK]	Seq=4012937021 Ack=3987339891 Win=5840 Len=1024
6	0.002851	10.0.5.22	10.0.5.11	TCP 4444>3062	[ACK]	Seq=3987339891 Ack=4012937021 Win=7168 Len=0
7	0.002939	10.0.5.11	10.0.5.22	TCP 3062>4444	[ACK]	Seq=4012938045 Ack=3987339891 Win=5840 Len=1448
8	0.002952	10.0.5.11	10.0.5.22	TCP 3062>4444	[ACK]	Seq=4012939493 Ack=3987339891 Win=5840 Len=1448
9	0.003027	10.0.5.22	10.0.5.11	TCP 4444>3062	[ACK]	Seq=3987339891 Ack=4012938045 Win=9216 Len=0
10	0.003051	10.0.5.11	10.0.5.22	TCP 3062>4444	[ACK]	Seq=4012940941 Ack=3987339891 Win=5840 Len=1448

11	0.003060	10.0.5.11	10.0.5.22	TCP 3062>4444	[ACK]	Seq=4012942389 Ack=3987339891 Win=5840 Len=1448
12	0.008083	10.0.5.22	10.0.5.11	TCP 4444>3062	[ACK]	Seq=3987339891 Ack=4012939493 Win=11584 Len=0
13	0.008175	10.0.5.11	10.0.5.22	TCP 3062>4444	[ACK]	Seq=4012943837 Ack=3987339891 Win=5840 Len=1448
14	0.008187	10.0.5.11	10.0.5.22	TCP 3062>4444	[FIN,PSH, ACK]	Seq=4012945285 Ack=3987339891 Win=5840 Len=952
15	0.008147	10.0.5.22	10.0.5.11	TCP 4444>3062	[ACK]	Seq=3987339891 Ack=4012940941 Win=14480 Len=0
16	0.008251	10.0.5.22	10.0.5.11	TCP 4444>3062	[ACK]	Seq=3987339891 Ack=4012942389 Win=17376 Len=0
17	0.008646	10.0.5.22	10.0.5.11	TCP 4444>3062	[ACK]	Seq=3987339891 Ack=4012943837 Win=20272 Len=0
18	0.011128	10.0.5.22	10.0.5.11	TCP 4444>3062	[ACK]	Seq=3987339891 Ack=4012945285 Win=23168 Len=0
19	0.011810	10.0.5.22	10.0.5.11	TCP 4444>3062	[FIN,ACK]	Seq=3987339891 Ack=4012946238 Win=26064 Len=0
20	0.011879	10.0.5.11	10.0.5.22	TCP 3062>4444	[ACK]	Seq=4012946238 Ack=3987339892 Win=5840 Len=0