Possible Quiz Questions for Quiz #3 on September 12th, 2019

The following is a list of possible questions for our Quiz #3 on September 12th. Some of the questions will not be asked in the quiz. All the questions that will appear in the quiz will appear exactly as shown below (however, parameters may be changed). The quiz is closed textbook, closed notes and closed neighbors. Note that the questions, which did not appear in this quiz, still may appear in the exams.

#1: What are “ports” used by socket connections? Why does socket use “ports”?

#2: Show the structure of the server-side socket process (what APIs are used in what order in a server process) – for a server that establishes socket connections, receives a payload message from clients, and sends a response to each client.

#3: Show the structure of the client-side socket process for the server-side process for #2 above.

#4: What are “blocking API calls ("blocking function calls")”?

#5: Which socket APIs are blocking APIs (mention all the blocking socket APIs to be used in Project #1 in this course)?

#6: What does “socket” API do?

#6: What does “bind” API do?

#7: What does “listen” API do?

#8: What does “accept” API do?

#9: Why does “accept” API duplicates a socket connection through another port when a server process establishes a socket connection with a remote client?

#10: What information does “sockaddr_in” structure (i.e., “struct sockaddr_in”) holds?

#11: On success, what information does the sockaddr_in structure specified as the second parameter in accept API hold?
#12: The following program structure is for a server-side process using sockets. What will happen, if the primary socket ("server_s") is closed, instead of a duplicated connection by closesocket(client_s) at the end of the server-side process structure?

```c
// Main loop to listen, accept, etc.
while (1) {
    listen(server_s, 100);
    ......
    client_s = accept(server_s, ......);
    ......
    retcode = recv(client_s, ......);
    ......
    send(client_s, ......);
    ......
    closesocket(client_s);
}
```

#13: How does “stop-and-wait flow control” work?

#14: What is the primary weakness in “stop-and-wait flow control”?

#15: What is “packet transmission delay”? How is “packet transmission delay calculated (show the formula)?

#16: What is “signal propagation delay”? How is “signal propagation delay calculated (show the formula)?

#17: What is “link utilization”? How is “link utilization” calculated for stop-and-wait flow control (show the formula)?

#18: Show the formula to calculate the link utilization for stop-and-wait flow control.

#19: How does “sliding-window flow control” work?

#20: What is the primary advantage of “sliding-window flow-control” over “stop-and-wait flow-control”?

#21: If the sliding-window flow-control is used in the Internet, what are the two primary weaknesses in the sliding-window flow-control?

#22: How “TCP slow-start and linear growth” work?

#23: How “TCP slow-start and linear growth” solve the two problems you identified for question #21 above?