Ramya Madhavareddy

11718845

**Assignment 2**

**Q2. Difference between stateless and stateful protocol.**

|  |  |
| --- | --- |
| **Stateful** | **Stateless** |
| 1. Stateful protocol stores the data and process information of previous sessions.  2. During the communication between client and server for a stateful protocol the client waits for the response of server and sends back the request if the response is not received.  3. As session information needs to be stored the stateful protocol needs lot of memory.  4. There is a lot of dependency between client and server in stateful protocols.  5.As the client waits for the server response and the client is dependent on server the transaction speed is low.  6.The implementation of stateful protocol is much complex | 1. Stateless protocols do not store the data and process of the previous sessions.  2. Stateless protocol communication barely waits for the response from the server.  3. The memory consumption for stateless protocol is much lower when compared to stateful protocols.  4. Stateless protocols have comparatively less dependence between server and client.  5. As there is no interdependence between client and server the communication speed is higher in stateless protocol.  6. Implementation of stateless protocol is easy when compared to stateful protocols. |

**Q4. Comaprison between Go Back N and selective repeat.**

1. Go Back N and selective repeat both of the protocols are based on sliding window protocol. Go Back N retransmits all the packets again after if it does not receive the packets in order .Where as in Selective repeat protocol the packets which are missing are retransmitted.

2. As Go Back N transmits the next packets of the lost packets in the window when a loss occurs a lot of duplicates are transmitted unnecessarily which increases the network congestion and wastes the network bandwidth where as in selective repeat as only the required packets are transmitted the number of duplicate packet transmission is low.

3. Go Back N is simpler to implement and understand because it involves lesser retransmissions when compared to selective repeat.

4. Go back N can be used in the scenario where the packet losses are fine in transmission.

Selective repeat can be used in the cases where the packet losses are less and bandwidth is more.

5. Go Back N requires less memory as it does not want to keep track of the acknowledgements.

**Q6. Difference between flow control and congestion control.**

Flow control is the mechanism used to control the flow of data or packets of data between the sender and receiver. It involves just the sender and receiver and the connection between the sender and receiver. In flow control whenever the receiver is overwhelmed with the messages from the sender the receiver just intimates the sender that its processing capacity of the packets is bit lower than the input capacity. This mechanism is implemented to reduce the loss if packets due to the high transmission of data. The flow control can be implemented in the network using the flow control protocols such as stop and wait and sliding window. Acknowledgments are introduced for sending back the response for the data being sent for preventing the loss of packets. Flow control mechanism is implemented in both transport layer and data link layer.

Whereas congestion control is the scenario in which multiple senders and receivers are involved and multiple parallel transmissions are happening in the network. The TCP header has 3 flags specifically allocated for congestion in network.

In Analogy with the real world, let us consider a class which has numerous students. If two students are talking to each other and one of the student speaks much faster and the other one is not able to catch up it actually indicates that the listener is missing many words while trying to catch up. So the remedy here is the listener now should acknowledge the speaker that “Hey!Can you just slow down a bit and speak it again please I’m just not able to get you”. Similarly for flow control we use acknowledgements based protocols.

So to create an analogy of congestion let us consider the entire students in the class are speaking to others who are a bit far from each other. Now if everyone in the class is speaking a lot at the same time it seems to be noisy and messy. This is what a congestion means, numerous hosts are communicating with much data transfer between them at the same time.

**Q8**.**Can we differentiate between a packet loss and acknowledgment loss as a sender**.

In a reliable transfer protocols we make sure the data is sent appropriately to the receiver. So we have introduced the concept of acknowledgments to ascertain the receiver has received the message. But sometimes the acknowledgements get lost during the transmission so in this case too we will wait for a certain time and if the acknowledgement is not received we consider it as acknowledgment loss and transmit the packet.

But the acknowledgement may not be sent from the receiver because the packet itself is lost while being sent from sender. But the sender does not know if it really got lost.

Hence I think in reliable data transfer protocol the sender will not be able to differentiate between a packet loss and a acknowledgment loss, as in both cases it just waits for the acknowledgment timeout and retransmits the data.

But we can be able to differentiate between a packet loss and acknowledgment loss at the receivers in some protocols like selective repeat and Go Back N as the receiver has the buffer of the packets received till now.

**Q10.Size of TCP Header excluding options**

The TCP header is generally 20-60 bytes in which 40 bytes is for options and 20 bytes is for other fields. So if we exclude the options from it the TCP header is of 20 bytes.

These 20 bytes of header has different fields such as Source port, destination port, Sequence number, acknowledgment number, Window, Checksum, Urgent Pointer, Do, Flags, RSV.

20 bytese

Source Port Destination Port

Sequence Number

Acknowledgement number

Do RSV Flags Window

Checksum Urgent Pointer

Options(40 bytes)

**Q12.why should not we set TCP timeout too high**

TCP timeout refers to the time limit the sender waits for the acknowledgment. If the sender does not receive any response i.e. acknowledgments till the timer value then the sender retransmits the data again.

So if we consider of setting a large time out value to minimize the issue of early timeouts then we will face the issue of slow transmission rate overall the network. As we set large time out values it means that in case of lost packet your are waiting for a lost packet for a longer time an holding on the next packet transmission till the timer. If you consider a timeout that is bit lower than the previous high then you will know faster that the packet has been lost and you will also be able to retransmit in the same time. So the things get to be done faster.

So if we set a larger time out value it means that you are responding late to packet loss and slowing down the transmission.

**Q14. 3 way handshake of TCP**

TCP is a connection oriented protocol so it establishes a connection every time before it starts sending the data. Generally TCP follows 3 way handshake protocols. Earlier 2 way handshake is used for establishment of connection but this has its own disadvantages.

In a 2 way handshake the client sends a message to the server with a sequence number x and it waits for the acknowledgment of server with the server chosen sequence number Y. Suppose the sender has sent a connection request X and the acknowledgment has delayed or timed out and the client sent a new connection request Z and now after the timeout the acknowledgment for the connection request X has been sent .The client now feels that it was the response to the latest connection request Z but that should not be the case.

Hence the 2 way handshake does not work properly. To overcome this we have used the 3 way shake protocol in which the acknowledgment from the server comes with both sequence and the acknowledgment numbers in which acknowledge number is the sequence number of previous packet. Hence the confusion on the response from server can be resolved.

**Q16. Symptoms of Congestion:**

The two symptoms of congestion are

1. Loss of packets or delay in the responses

2. Queuing of packets in the buffer

So whenever we see the situation in which most of the responses are reaching to the server after the timeouts which is resulting in continuous retransmission of packets we can infer from it that there is some congestion going on in the network.

And if the buffers are also full it also indicates that there is congestion in network due to which the packets are not getting processed properly.

**Q22.**

**a.**

The minimum distribution time for a client server distribution is

Dc-s > max{NF/us,,F/dmin}

So x=max(100\*25\*10^9/30\*10^6 ,25\*10^9/3\*10^6)

X=max(25\*10^4/3 sec,25\*10^3/3 sec)

X=max(83,333 sec, 8,333 sec)

Finally the minimum distribution time for client and server distribution is >= 83,333 seconds

**b.**

The minimum distribution time for a peer to peer distribution is

DP2P >= max{F/us,,F/dmin,,NF/(us + Sui)}

X=max(25\*10^9/30\*10^6 ,25\*19^9/3\*10^6 , 100\*25\*10^9/(30\*10^6 +40\*10^6)}

X= max(833.33 sec,8333.33 sec,35,714.28 sec}

>=35,714.28 sec

Hence the minimum distribution time for a peer to peer distribution is >=35,714.28 seconds

**Q25.**

The UDP segment has the following parameters in it.

Source IP,Destination IP,Source Port,Destination Port,Length and checksum.

Source port destination port

Length checksum

Data

Length of a UDP segment is based on both the header and data fields.

So the length of header of UDP segment is 8 bytes

And the each number in data is of 16 bits so in total the data is of 32 bits i.e 4 bytes.

So header length+data length = 8 bytes\_+ 4 bytes

=12 bytes

= 96 bits

Checksum:

Source -110011100110

Destination -1010000

9324- 0010010001101100

8545- 0010000101100001

= 10100010010010011 i.e it ha s 1 carry out so after adding this carry out to the number

We get 10100010010010100

After performing the one’s compliment we get 01011101101101011i.e the final check sum.

**Q27.**

1. In the second segment sent from host A to B the sequence number is 127+88=215

Source port= 1002

Destination port= 80

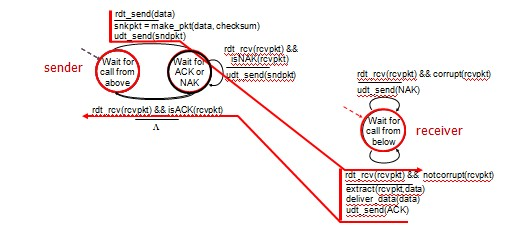
1. If the first segment arrives before the second segment the acknowledgment number of the first arriving segment is 215 the source port is 80 and destination port is 1002.
2. If the second segment arrives before the first segment then the acknowledgment number received for the segment is 128 since the next 44 bytes of 127 bytes is not received yet.

**Q29.**

Finite State machine for Reliable data transfer:

Finite State machines (FSM’s) are the base for any implementation of the mechanism suggested. It can be considered as the blue print of the flow that will be followed in the mechanism. The FSM mainly constitutes of States and the actions .Each State moves to another state on the application of any action. A start and stop states are mandatory for an FSM.

RDT with no data error:



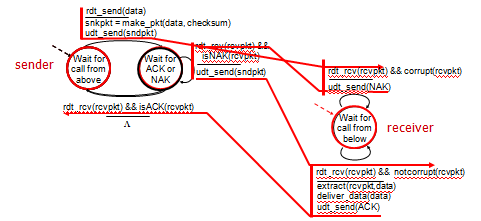
In a reliable data transfer without any data corrupt the flow is as follows.

Sender send the data and then moves to wait state for the acknowledgment from the

Receiver. Once it receives the acknowledgment from the receiver it moves to the start

state again and transfers data.

FSM for corrupt data



In the case of a corrupt data after it moves to the wait state it checks if the acknowledgment

received is ACK or NAK. If it receives NAK then it retransmits the same data again and

waits for the acknowledgment.

**Q31.**

1. Slow start phases time stamp

In the given graph slow start triggered on 3 timestamps

Between t0-t1

Between t5 to t6

Between t8 to t9

Congestion avoidance phase’s timestamp

In the given graph congestion avoidance phases triggered on

T1-t2,t2-t3,t3-t4,t4-t5,t6-t7,t7-t8

b.

The ssthresh during the time periods below are

T0-t1 –w3

T2-t3 –w3/2

T3-t4 –w2/2

T7-t8 –w3

T8-t9 -- w1

c.

The TCP Version is Reno as the Congestion window size is getting reduced half of the previous window size after loss of acknowledgments is happening. Hence we can consider it as Reno. In the case of Tahoe the window size gets reduced to 1 by default. But as per the graph given the size of the next window is reduced to only half of the previous window.