CS 335, Assignment 3
Vaibhav Sharma
200365101
s
Total = 46
1.
(i) [2] Mention any two services that TCP can provide but UDP cannot provide.
Solution - TCP provides Flow control and congestion control.

(ii) [2] Suppose a process in Host C has a UDP socket with port number 6789. Suppose both Host A and Host B each send a UDP segment to Host C with destination port number 6789. Will both of these segments be directed to the same socket at Host C? If so, how will the process at Host C know that these two segments originated from two different hosts?
Solution - Yes, Both packets will come to the same socket at Host C because both of them have same destination port and Host c will differentiate them on the basis of the source ip, header and source port.

(iii) [6] Suppose Client A initiates a session with Server S. At about the same time, Client B also initiates a session with Server S. Provide possible source and destination port numbers for:
(a) The segments sent from A to S.
Solution -

Source port number = A -> 2134
Destination port number = S -> 80
2134 -> 80

(b) The segments sent from B to S.
Solution -

Source port number = b -> 2131
Destination port number = S -> 80
2131 -> 80

(c) The segments sent from S to A.
Solution -

Source port number = S -> 80
Destination port number = A -> 2134
80 -> 2134

(d) The segments sent from S to B.
Solution -

Source port number = S -> 80
Destination port number = B -> 2131
80 -> 2131

(e) If A and B are different hosts, is it possible that the source port number in the segments from A to S is the same as that from B to S?

Solution -> Yes, because the port number on different host don't have any relationship with each other.
(f) How about if they are the same host?
Solution -> It's not possible that the source port number in the segments
from A to S is the same as that from B to S if they are on same host.
(iv) [2+2] Suppose you have the following 2 bytes: 01011100 and 01100101. What is the 1s complement of the sum of these two bytes? Give an example where 1 bit is flipped in each of the two bytes and yet the 1s complement does not change.
Solution -

Adding

```
          01011100
          01100101
          -------------
  Sum -   11000001

Checksum - 00111110
```

Now changing the first number to 11011100 and second to 11100101

Adding

```
          11011100
          11100101
          -------------
    Sum  11000001
Checksum - 00111110
```

Same checksum for two different bits added together

2.
(i) [3] What are the three properties of Reliable Data Transfer?
Solution -
No corrupt -> Reliable data transfer make sure is not corrupted and receiver receives exactly what user wants to send

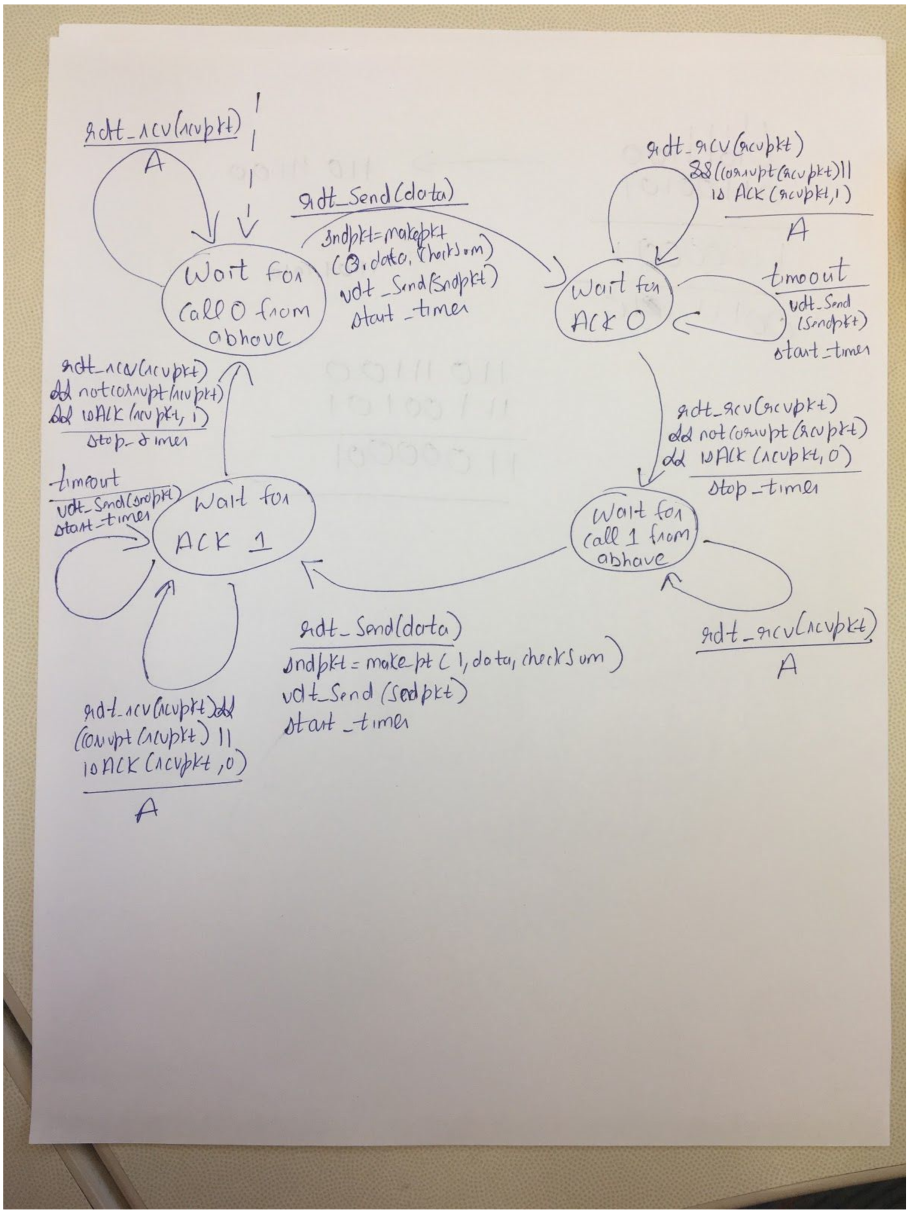
No loss -> Reliable data transfer make sure the data is not lost and in case it's lost then it be resent again.
In Order -> Reliable data transfer make sure the data sent by the sender is received by the receiver in same order.

(ii) [2+2] Suppose host A sends a packet to host B. How can A be ensured that the packet did not corrupt in the channel? If the packet was corrupted, what can be done?
Solution - When a host A send a packet to host B then checksum is used to make sure the packet did not get corrupt. If the packet is corrupted then the host B let host A know to resend the packet. There are many different ways to let the sender know, either by NACK or sequence ACK or timeout.

(iii) [2+2] In our rdt protocols, why do we need to introduce sequence numbers and timers?
Solution -

Introduction to sequence number is important as it takes out the need to send back NACK and receiver can just send the ACK with the sequence number of the the successfully received packet.
It also make sure the packets are sent in order.
Introductions to timers is important in case when sender send a packet to receiver and it gets lost or when the ACK sent by the receiver is lost then the sender needs a timeout after which it will send packet again if there is no ACK.

(iv) [5] Draw the Finite State Machine (FSM) for the receiver side of protocol rdt3.0.



3.
(i) [4] Consider the data transfer between two hosts A and B, with the stop-and-wait protocol, rdt3.0. A is located on the West Coast of the United States and B is located on the East Coast. The speed-of-light round trip propagation delay between A and B, RTT, is approximately 30 milliseconds. Suppose that A and B are connected by a channel with a transmission rate, R, of 1Gbps. Given the packet size, L, of 1500 bytes, determine the channel utilization.
Solution -

1 gbps , 1500 bytes= 12000bits,

Dtrans = L/R = 12000/10^9 = 12 microseconds

Utilisation - (L/R)/ (RTT +(L/R))

= 0/0012/ 30.0012 = 0.00004 = 0.04%

(ii) [6] Following the problem (i), consider the pipelined technique instead of stop-and-wait protocol. How big would the window size have to be for the channel utilization to be greater than 98%?

If the pipelined technique is used

Utiisation - (W*(L/R))/(RTT +(L/R))

= (W*0.0012)/30.0012 in order for Utilisation to be bigger than 0.98

W = (0.98 * 30.0012)/0.0012 = 24,501

Window size need to be at least 24,501 for 98%

(iii) [2] What does the receive window (rwnd) field in a TCP segment indicate?

Solution - Receive window in TCP is used for flow control. It indicates the sender that how many bytes receiver can accept at that time.

(iv) [2+2] Why are the flow control and congestion control needed?

Solution - The flow control is important because if the sender is sending packets and buffer is full then it will lead to packet loss and flow control prevent that from happening.

Congestion control is important as it helps prevent long delays and packet loss due to full buffer. There is End-to-End congestion control used by TCP and Network assisted congestion control where routers send feedback to prevent packet loss and delay.