



## **Computer Networks 1/ HW2**

**Student Name : Imadeddin Ahmad Hasan Salama**

**Teacher Name : Abdallah Rashed**

**HW#2 Solutions**

## Q1 Solution :

The Check sum is a way to be sure that your message isn't corrupted or there are any errors in the message received, and this check happens in the Transport layer over not reliable network layer.

The check sum is computed by splitting the packets into 16 bits chunks and then add them all and the carry should be added after that then getting the 1's complement for the result, the result is the checksum .

In our example the check sum is got as we can see

1101011111011110

0111100010010110 +

Adding the Packets

10101000001110100



1+

Adding the Carry

0101000001110101

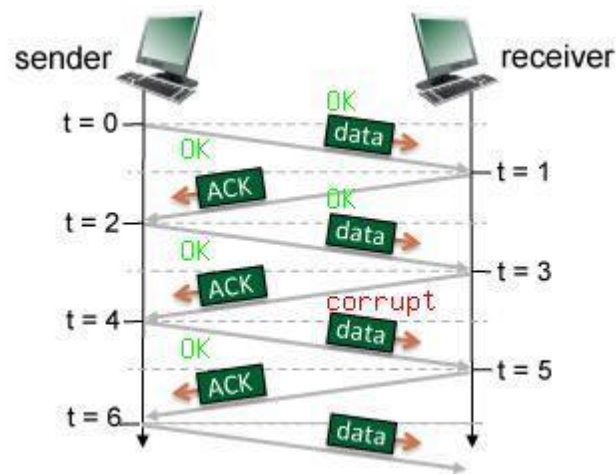
1's Complement to give :

1010111110001010 is the final result of the checksum

**The Final result of the Check sum is : 1010111110001010**

## Q2 Solution :

RDT 2.2 protocol is NACK free protocol so it sends the ack as positive answer to the sender to give him a way that it's received without corruption or error.



t	Sender state	Receiver state	Packet type sent	Seq# or Ack#
0	Send data0 and Wait Ack0	Wait0 data from below	data	0
1	Wait Ack0	Send Ack0 and Wait1 data from below	Ack	0
2	Send data1 and Wait Ack1	Wait1 data from below	data	1
3	Wait Ack1	Send ack1 and Wait2 data from below	Ack	1
4	Send data2 and Wait Ack2	Wait2 data from below	data	2
5	Wait Ack3	Send ack1 and wait the data again	ack	1
6	Receive Ack1 and resend packet2 again and wait Ack	Wait2 data	data	2

The idea here is about sending the last ack if the message is corrupted , so when sending the last succeeded ack what happens is the sender knows that it didn't receive it will so it will send it again . and this what happened in the question.

### **Q3 Solution:**

The Question here asks us to find the Estimated RTT and the time out , and the DevRTT which is a space enough after the RTT to give enough time for the time out for the packets to be sent , no slow server reactions ., no less throughput through resending .

**EstimatedRTT = (1-a)\*EstimatedRTT + a\*SampleRTT**

**DevRTT = (1-B) \*DevRTT + B\* | SampleRTT-EstimatedRTT |**  
**(typically, B= 0.25)**

**(typically, a= 0.125)**

**TimeoutInterval = EstimatedRTT + 4\*DevRTT**

\*The first iteration of EstimatedRTT=242 ms , The first iteration of VerRTT=51 ms.

EstimatedRTT1=0.875\*242+0.125\*320=211.75+40=251.75ms

DevRTT1=0.75\*51+0.25\*|320-251.75|=17.06+38.25=55.31ms

Timeout=251.75+4\*55.31=472.99ms

---

EstimatedRTT1=0.875\*251.75+0.125\*360=220.28+45=265.28ms

DevRTT1=0.75\*55.31+0.25\*|360-265.28|=41.48+23.68=65.16ms

Timeout=265.28+4\*65.16=525.92ms

---

EstimatedRTT1=0.875\*265.28+0.125\*260=232.12+32.5=264.62ms

DevRTT1=0.75\*65.16+0.25\*|260-264.62|=48.87+1.155=50.025ms

Timeout=264.62+4\*50.025=464.72ms

---