P3. UDP and TCP use 1s complement for their checksums. Suppose you have the following three 8-bit bytes: 01010011, 01100110, 01110100. What is the 1s complement of the sum of these 8-bit bytes? (Note that although UDP and TCP use 16-bit words in computing the checksum, for this problem you are being asked to consider 8-bit sums.) Show all work.

Why is it that UDP takes the 1s complement of the sum; that is, why not just use the sum? With the 1s complement scheme, how does the receiver detect errors? Is it possible that a 1-bit error will go undetected? How about a 2-bit error

**01010011+01100110=10111001**

**10111001+01110100=100101101 carry=1**

**00101101+1=00101110**

**1s complement: 11010001**

**UDP use the 1s complement of the sum so that the client can add the checksum to the original message. If any 0 occurs, it knows there is an error.**

**All the 1-bit error will be detected, but 2-bit errors can be undetected.**

P8. Draw the FSM for the receiver side of protocol rdt3.0.

