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?

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | |
| Calculate the sum of the given 3 bytes. | | |
| Add first two 8-bit bytes: | | |
| | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | | | |
| | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | | | |
| | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | | | |
| First， add the result with the 3rd byte. | | |
| | | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | | | |
| | | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | | | |
| | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | | | |
| Second，wrap around the extra bit. | | |
| | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | | | |
| | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | | | |
| | | | | | | | | 1 | | | |
| | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | | | |
| The sum three 8-bit bytes is 00101110. Invert all the bits to get the check sum. | | |
| Check sum is 11010001. | | |

Just add it as part of the data and judge whether it is all 1 in the end. If you only use and sum, you need to save the sum separately. Finally, the sum of the data is calculated and compared. To store sum separately, you need to use one more register.

When one bit of the sum result is found to be 0, it indicates that an error has occurred.

An error in one bit can be detected, but an error in two bits may not.

P8

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

