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?

Answer:

To find the 1s complement of the sum of the three 8-bit bytes (01010011, 01100110, 01110100), we need to:

1. Add the three bytes together.
2. Take the 1s complement of the sum.

Let's start with step 1:

01010011 + 01100110 + 01110100

=101011101

Now, we take the 1s complement of the sum:

101011101

↓

010100010

So, the 1s complement of the sum of the three 8-bit bytes is 010100010.

Now, let's address the questions:

1. **Why does UDP use 1s complement for checksums?**
   * UDP uses 1s complement checksums because it's a simple and efficient method for error detection. The 1s complement allows for easy computation of the checksum and simplifies error detection by ensuring that a change in any bit results in a different checksum.
2. **How does the receiver detect errors with the 1s complement scheme?**
   * The receiver calculates the checksum over the received data, including the checksum itself. If the result is all 1s (binary 11111111), it indicates that no errors were detected. If any bit is flipped during transmission, the checksum will not match the expected value, indicating that an error occurred.
3. **Is it possible that a 1-bit error will go undetected? How about a 2-bit error?**
   * No, a 1-bit error will not go undetected because any change in the transmitted data will result in a different checksum. So, even a single-bit error will cause the receiver's calculated checksum to differ from the sender's transmitted checksum.
   * Similarly, a 2-bit error will also be detected because it will result in a different checksum. The 1s complement checksum detects errors effectively, regardless of the number of bits flipped during transmission.

P4. a. Suppose you have the following 2 bytes: 01011100 and 01100101. What is the 1s complement of the sum of these 2 bytes?

b. Suppose you have the following 2 bytes: 11011010 and 01100101. What is the 1s complement of the sum of these 2 bytes?

c. For the bytes in part (a), give an example where one bit is flipped in each of the 2 bytes and yet the 1s complement doesn’t change.

Answer:

1. Adding the two bytes gives 11000001. Taking the ones complement gives 00111110.
2. Adding the two bytes gives 01000000; the ones complement gives 10111111.
3. First byte = 01010100; second byte = 01101101.

Coding exercise:

Using python to implement the Internet checksum algorithm. Notice the callout: "use matplotlib". For those of you who are unfamiliar, matplotlib is a plotting library in Python. I want you to use it to visualize your results from the checksum calculation. This could be any creative way you choose to illustrate the effectiveness of the checksum algorithm.

