IP Datagram Header Checksum method

This is a review of the method used at the Sender and at the Receiver sides of the process. We show how the sender creates the value that will be placed into the checksum field of the IP datagram and how the receiver operates on the header fields to verify that there were no errors in transit.



**Figure 1: IP Datagram**

1. Look at the IP datagram above and notice that the header checksum uses up 2 bytes of the header. Notice which bytes they are, by counting. Recall that there are 4 bytes in each row and that the bits are numbered in Figure 1. So, the header checksum bytes are bytes 11 and 12.

2. For the purposes of this example, let the bytes be represented as follows, where each letter pair represents a hexadecimal representation of the bits in the header.

ab cd

ef gh

ij kl

mn op

qr st

uv wx

yz ab

cd ef

gh ij

kl mn

3. The bits that represent the header checksum will be those in 6th row of the sum above. They are uv wx.

4. The Sender makes those bits into 0’s and then performs a sum operation, using “ones-complement addition.” I have not asked you to learn how to perform this addition. The sum that the Sender sets up looks like this, where I have let S equal the result of the ones-complement sum:

ab cd

ef gh

ij kl

mn op

qr st

00 00

yz ab

cd ef

gh ij

kl mn

\_\_\_\_\_\_

S

5. Now the sender performs a second operation. It takes the ones-complement of the value S. I have not asked you to learn how to perform this operation either. I will represent the ones-complement of S as –S (minus S) because taking the ones-complement of a number produces the negative of that number in ones-complement arithmetic.

6. Now the sender inserts the value –S into the 11th and 12th bytes of the header, removing the 00 00 that it used to create the values S and –S.

7. The Sender sends the datagram.

8. On the Receiver side: The Receiver in effect creates the sum,

ab cd

ef gh

ij kl

mn op

qr st

<-S>

yz ab

cd ef

gh ij

kl mn

\_\_\_\_\_\_

S + -S

and checks that the result is 0. If it is 0 then the receiver concludes that there were no errors in transmission.

This summation is done in ones-complement arithmetic and so after the sum, the ones-complmeent of the result is 0. But, again, I have not asked you tolearn ones-complement arithmetic.

It is sufficient to know how to calculate the checksum and how to verify it at the receiver side.