Batch: B1	**Roll No.: 1711072**
Experiment / assignment / tutorial No. 4	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

Experiment No.:4

TITLE: Implementation of Checksum for Computer Networks

AIM: To implement Layer 2 Error Control schemes: Checksum.

Expected Outcome of Experiment:
CO: Describe Data Link Layer, MAC layer technologies & protocols and implement the functionalities like error control, flow control.

Books/ Journals/ Websites referred:

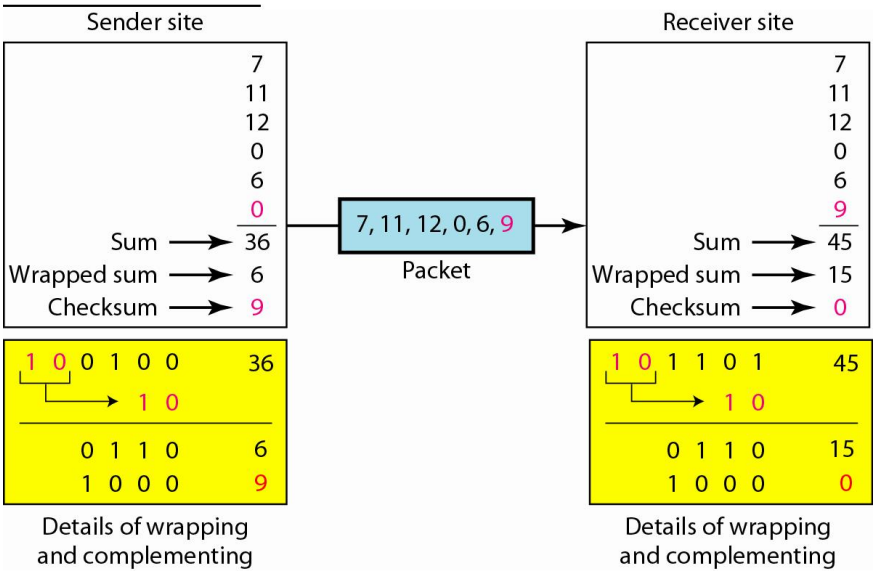
1. A. S. Tanenbaum, "Computer Networks", Pearson Education, Fourth Edition
2. B. A. Forouzan, "Data Communications and Networking", TMH, Fourth Edition

Pre Lab/ Prior Concepts:
Data Link Layer, Error Correction/Detection, Types of Errors

New Concepts to be learned: Checksum.

Checksum :
A checksum is a simple type of redundancy check that is used to detect errors in data.
Errors frequently occur in data when it is written to a disk, transmitted across a network or otherwise manipulated. The errors are typically very small, for example, a single incorrect bit, but even such small errors can greatly affect the quality of data, and even make it useless.
In its simplest form, a checksum is created by calculating the binary values in a packet or other block of data using some algorithm and storing the results with the data. When the data is retrieved from memory or received at the other end of a network, a new checksum is calculated and compared with the existing checksum. A non-match indicates an error; a match does not necessarily mean the absence of errors, but only that the simple algorithm was not able to detect any.

Simple Checksum:



Internet Checksum

The following process generates Internet Checksum

Assume the packet header is: 01 00 F2 03 F4 F5 F6 F7 00 00
(00 00 is the checksum to be calculated)

The first step is to form 16-bit words.
0100 F203 F4F5 F6F7

The second step is to calculate the sum using 32-bits.
0100 + F203 + F4F5 + F6F7 = 0002 DEEF

The third step is to add the carries (0002) to the 16-bit sum.
DEEF + 002 = DEF1

The fourth step is to take the complement. (1s becomes 0s and 0s become 1s)
~DEF1 = 210E

So the checksum is 21 0E.

The packet header is sent as: 01 00 F2 03 F4 F5 F6 F7 21 0E

* At the receiver, the steps are repeated.

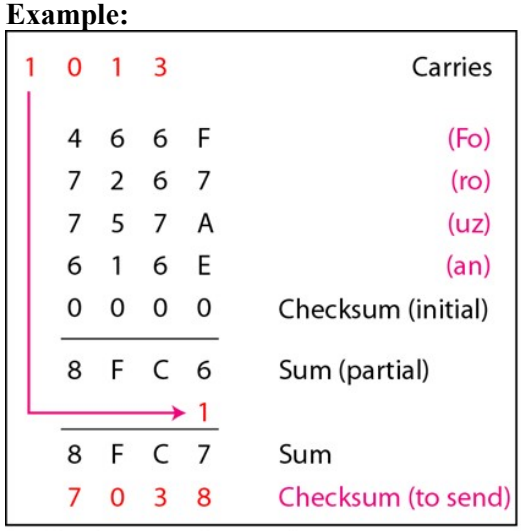
The first step is to form 16-bit words.
0100 F203 F4F5 F6F7 210E

The second step is to calculate the sum using 32-bits.
0100 + F203 + F4F5 + F6F7 + 210E = 0002 FFFD

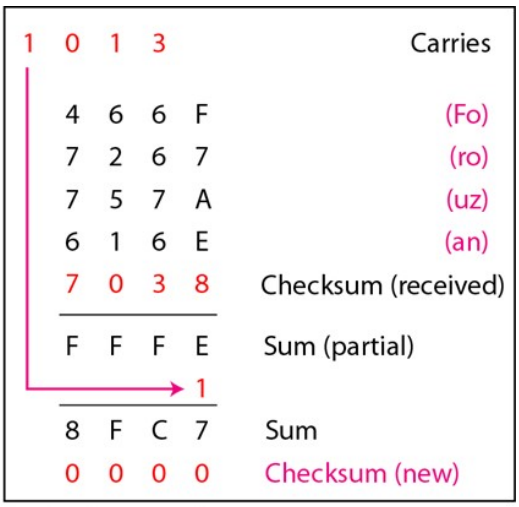
The third step is to add the carries (0002) to the 16-bit sum.
FFFD + 0002 = FFFF which means that no error was detected.

(In 1s complement, zero is 0000 or FFFF.)

Example:



a. Checksum at the sender site



a. Checksum at the receiver site

IMPLEMENTATION: (printout of codes)

```
d=input("Enter the data to send: ")
d=[(d[i:i+4]) for i in range(0, len(d),4)]
print('16 bit word format: ',','.join(str(x) for x in d))
def checksum(d):
  data=[int(i,16) for i in d]
  res=hex(sum(data))
  print("Sum is: ",res)
  res=res[2:]
  wrap_sum=hex(int(res[0],16)+int(res[1:],16))
  wrap_sum=wrap_sum[2:]
  print('Wrapped sum: ',wrap_sum)
  diff=hex(int('FFFF',16)-int(wrap_sum,16))
  diff=diff[2:]
  print('15\'s compliment: ',diff)
checksum(d)
r=input("Enter the data received: ")
r=[(r[i:i+4]) for i in range(0, len(r),4)]
checksum(r)
```

Output Screen:

```
Enter the data to send: 466F726F757A616E
16 bit word format:  466F,726F,757A,616E
Sum is:  0x18fc6
Wrapped sum:  8fc7
15's compliment:  7038
Enter the data received: 466F726F757A616E7038
Sum is:  0x1fffe
Wrapped sum:  ffff
15's compliment:  0
```

CONCLUSION:

The Python code for Checksum was implemented successfully as it ran correctly for all test cases.

Post Lab Questions

1. State the advantages and disadvantages of Internet Checksum.

Ans.

Advantages:

1. Checksum method is simpler than CRC method for error checking.
2. A checksum is mainly employed in data validation when implementing software.
3. Checksum method is simple to implement.

Disadvantages:

1. Checksum is an old method for error checking.
2. Due to its simple implementation, it can only detect single bit errors.
3. It cannot be used for error correction in analogue data transmission.