Experiment - 13

Q)) Write a program for congestion control using Leaky bucket algorithm.

code:

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>

#define NOF_PACKETS 5
 /*
 int rand(int a)          rn → random.
 {
     int rn = (random() % 10) % a;
     return rn == 0 ? 1 : rn;
 }
 */
/*
 #include<stdlib.h>
 long int random(void);
```

The random() function uses a nonlinear additive feedback random number generator employing a default table of size 31 long integer to return successive pseudo-random number in the range from 0 to RAND_MAX.

The period of this random number generator is very large, approx

$16 * ((2^{31}) - 1)$.

```
int main()
{ int packet_sz[NOF PACKETS], i, clk, b_size,
  o_rate, p_sz_rm = 0, p_sz, p_time, op;
  for(int i = 0 ; i < NOF_PACKETS ; ++i)
    packet_sz[i] = random()%100 ;
  for (int i = 0; i < NOF_PACKETS; ++i)
    printf (" \n packets [%d]: %d bytes\t",
    i, packet_sz[i]);
  printf ("\n Enter the output rate:");
  scanf (" %d", &o_rate);
  printf (" Enter the Bucket-size:");
  scanf ("%d", &b_size);
  for ( i = 0; i < NOF_PACKETS ; ++i)
  {
    if (packet_sz[i] + p_sz_rm) > b_size)
      if (packet_sz[i] > b_size)
        print ("\n\n Incoming packet
    size (%d bytes) is Greater than
        bucket capacity (%d bytes)-
    PACKET REJECTED", packet_sz[i],
          b_size);
    else
      printf ("\n\n Bucket capacity
  exceeded - PACKETS REJECTED!!");
  else
  {
    p_sz_rm += packet_sz[i];
    printf ("\n\n Incoming packet size:
      %d", packet_sz[i]);
    printf ("\n Bytes remaining to Transmit-
          %d", p_sz_sm);
```

```
// p-time = random() * 10;
//printf("\n Time left for transmission:
    %d units", p-time);
// for (clk = 10; clk <= p-time; clk += 10)
while (p-sz-rm > 0)
{
    sleep(1);
    if (p-sz-rm)
    {
     if (p-sz-rm <= o-rate)
         op = p-sz-rm, p-sz-rm = 0;
     else
         op = o-rate, p-sz-rm -= o-rate;
    printf("\n Packet of size %d Transmitted",
                op);
    printf(" ---- Bytes Remaining to
          Transmit : %d", p-sz-rm);
     }
     else
     {
         printf("\n No packets to
                transmit!!");
     }
  }
 }
}
}
```

output:

Packet [0]: 83 bytes
Packet [1]: 86 bytes
Packet [2]: 77 bytes
Packet [3]: 15 bytes
Packet [4]: 93 bytes.
Enter the output rate → 30
Enter the Bucket size : 85

Incoming Packet size : 83
Bytes remaining to transmit : 83
Packets of size 30 transmitted ....
Bytes Remaining to transmit : 53
Packets of size 30 transmitted ...
Bytes Remaining to transmit : 23
Packets of size 23 transmitted ...
Bytes remaining to transmit : 0

26/12/24