Design a C program for congestion control using leaky

bucket algorithm

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#define NOF\_PACKETS 10

int rand(int a)

{

int rn = (random() % 10) % a;

return rn == 0 ? 1 : rn;

}

int main()

{

int packet\_sz[NOF\_PACKETS], i, clk, b\_size, o\_rate, p\_sz\_rm=0, p\_sz, p\_time, op;

for(i = 0; i<NOF\_PACKETS; ++i)

packet\_sz[i] = rand(6) \* 10;

for(i = 0; i<NOF\_PACKETS; ++i)

printf("\npacket[%d]:%d bytes\t", i, packet\_sz[i]);

printf("\nEnter 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)/\*compare the packet siz with bucket size\*/

printf("\n\nIncoming packet size (%dbytes) is Greater than bucket capacity (%dbytes)-PACKET REJECTED", packet\_sz[i], b\_size);

else

printf("\n\nBucket capacity exceeded-PACKETS REJECTED!!");

else

{

p\_sz\_rm += packet\_sz[i];

printf("\n\nIncoming Packet size: %d", packet\_sz[i]);

printf("\nBytes remaining to Transmit: %d", p\_sz\_rm);

p\_time = rand(4) \* 10;

printf("\nTime left for transmission: %d units", p\_time);

for(clk = 10; clk <= p\_time; clk += 10)

{

sleep(1);

if(p\_sz\_rm)

{

if(p\_sz\_rm <= o\_rate)/\*packet size remaining comparing with output rate\*/

op = p\_sz\_rm, p\_sz\_rm = 0;

else

op = o\_rate, p\_sz\_rm -= o\_rate;

printf("\nPacket of size %d Transmitted", op);

printf("----Bytes Remaining to Transmit: %d", p\_sz\_rm);

}

else

{

printf("\nTime left for transmission: %d units", p\_time-clk);

printf("\nNo packets to transmit!!");

}

}

}

}

}

OUTPUT:

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated