```
// LEAKY BUCKET
#include<stdio.h>
int main()
{
  int bksize=10,pksize,buffer,outrate=6,i,n;
  printf("Enter the no. of packets: ");
  scanf("%d",&n);
  printf("bucketsize: %d ,\t output-rate: %d \n",bksize,outrate);
  for(i=1;i<=n;i++)
  {
    printf("second: %d\n Enter the incoming packet size : \n",i);
    scanf("%d",&pksize);
    printf("\nTime\tpacketsize\tbuffer\taccept/reject\tsend \n");
    if(pksize<=bksize-buffer)
    {
      buffer=buffer+pksize;
      if(buffer>=outrate)
      {
        buffer=buffer-outrate;
        printf("%d\t%d\t\t%d\taccept%d\t\t%d\n",i,pksize,buffer,pksize,outrate);
      }
      else
      {
        printf("%d\t%d\t\t%d\taccept%d\t\t%d\n",i,pksize,buffer,pksize,0);
      }
    }
    else
    {
      int dis=pksize-(bksize-buffer);
      buffer=bksize-outrate;
      printf("%d\t%d\t\t%d\treject%d\t\t%d\n",i,pksize,buffer,dis,0);
    }
  }
  printf("continue process...\n");
}
```

```
//gobacknserver
#include<sys/socket.h>
#include<stdio.h>
#include<unistd.h>
#include<string.h>
#include<stdlib.h>
#include <arpa/inet.h>
struct packet
{
  int data;
  int seq;
};
struct ackn
{
   int seq;
   int ack;
};
void main()
{
   int receiver,sender,len,c,t,wind,j,l,k,r,ran,f,i,PORT=1026;
   int opt=1;
   struct ackn a[10];
   char buffer[1024];
   struct packet p[15];
   struct sockaddr_in senderadd,recvadr;
   receiver=socket(AF_INET,SOCK_STREAM,0);
   recvadr.sin_family=AF_INET;
   recvadr.sin_port=htons(PORT);
   recvadr.sin_addr.s_addr=inet_addr("172.20.3.28");
   setsockopt(receiver, SOL_SOCKET,SO_REUSEADDR | SO_REUSEPORT, &opt,sizeof(opt));
   if (bind(receiver,(struct sockaddr*)&recvadr,sizeof(recvadr))<0)
      printf("Not Connected\n");
   listen(receiver,3);
   sender=accept(receiver,(struct sockaddr*)&senderadd,&len);
   i=read(sender,buffer,1024);
   buffer[i]='\0';
   printf("Received %s\n",buffer);
   send(sender,"hello",strlen("hello"),0);
   j=1;
   k=1;
   c=0;
   while(j<=4)
   {
       r=read(sender,(char *)&p[j],sizeof(p[j]));
          j++;
   }
   t=4;
   j=1;
```

```
  while(c<8)
  {
        ran=rand()%3;
        if(ran==0)
      {
        a[j].ack=-1;
           a[j].seq=p[j].seq;
        }
        else
        {
          a[j].ack=1;
          a[j].seq=p[j].seq;
        }
      if(a[j].ack==1)
      {
        printf("Received Packet with data %d and seq %d\n",p[j].data,p[j].seq);
           send(sender,(char*)&a[j],sizeof(a[j]),0);c++;j++;
        if(t<8)
        {
          t++;read(sender,(char*)&p[t],sizeof(p[t]));
        }
      }
        else
        {
          send(sender,(char*)&a[j],sizeof(a[j]),0);
        printf("\n\n");
        for(k=j;k<=t;k++)
        {
          read(sender,(char*)&p[k],sizeof(p[k]));
        }
       }
close(receiver);
}
```

```
//gobacknclient
#include<stdio.h>
#include<sys/socket.h>
#include<unistd.h>
#include<string.h>
#include <arpa/inet.h>
struct ackn
{
  int seq;
  int ack;
};
struct packet
{
  int data;
  int seq;
};
void main()
{
  int sender,len,wind,j,l,k,c,i,r,PORT=1026;
  char buffer[1024];
  struct ackn a;
  struct packet p[10];
  struct sockaddr_in receiver;
  sender=socket(AF_INET,SOCK_STREAM,0);
  receiver.sin_family=AF_INET;
  receiver.sin_port=htons(PORT);
  receiver.sin_addr.s_addr=inet_addr("172.20.3.28");
  connect(sender,(struct sockaddr*)&receiver,sizeof(receiver));
  send(sender,"hai",strlen("hai"),0);
  i=read(sender,buffer,1024);
  buffer[i]='\0';
  printf("Received %s\n",buffer);
  j=1;
  l=1;
  k=1;
  printf("Sending frame \n");
  while(j<=8)
  {
    printf("Enter Packet data \n");
    scanf("%d",&p[j].data);
    p[j].seq=j;
    j++;
  }
  while(l<=4)
  {
    send(sender,(char*)&p[l],sizeof(p[l]),0);
```

```
   printf("sending packet with data %d and seq no %d\n",p[l].data,p[l].seq);
   l++;
  }
  c=0;
  while(c<8)
  {
   r=read(sender,(char *)&a,sizeof(a));
   if (a.ack==1)
   {
         c=c+1;
         printf("received ack for packet %d\n",a.seq);
      if(l<=8)
         {
          printf("sending packet with data %d and seq no %d\n",p[l].data,p[l].seq);
          send(sender,(char*)&p[l],sizeof(p[l]),0);
          l++;
         }
   }
   if (a.ack==-1)
   {
      printf("\n\ntime expired for packet %d\n",a.seq);
         for(k=a.seq;k<l;k++)
         {
            printf("resending packet with data %d and seq no %d\n",p[k].data,p[k].seq);
         send(sender,(char*)&p[k],sizeof(p[k]),0);
         }
   }
  }
  close(sender);
}
```

```
sngce@sngce-OptiPlex-3000:~/vivek$ gcc gobknclient.c
sngce@sngce-OptiPlex-3000:~/vivek$ ./a.out
Received hello
Sending frame
Enter Packet data2
Enter Packet data3
Enter Packet datah
Enter Packet dataEnter Packet dataEnter Packet dataEnter
 datasending packet with data 2 and seq no 1
sending packet with data 3 and seq no 2
```

```
sngce@sngce-OptiPlex-3000:~/vivek$ gcc gobknserver.c
sngce@sngce-OptiPlex-3000:~/vivek$ ./a.out
Received hai
Received Packet with data 1 and seq 1
Received Packet with data 2 and seq 2


Received Packet with data 3 and seq 3
Received Packet with data 4 and seq 4
Received Packet with data 0 and seq 5
Received Packet with data 1 and seq 6
```

```
sngce@sngce-OptiPlex-3000:~/vivek$ cc leakybucket.c
sngce@sngce-OptiPlex-3000:~/vivek$ ./a.out
Enter the no. of packets: 3
bucketsize: 10 ,         output-rate: 6
second: 1
 Enter the incoming packet size :
5

Time    packetsize      buffer  accept/reject   send
1       5               5       accept5         0
second: 2
 Enter the incoming packet size :
8

Time    packetsize      buffer  accept/reject   send
2       8               4       reject3         0
second: 3
 Enter the incoming packet size :
2

Time    packetsize      buffer  accept/reject   send
3       2               0       accept2         6
continue process...
```