DHRUV GARG 16 BCE 1190

NETWORK AND COMMUNICATION

LAB 6

Implement the Go Back N flow control mechanism with the assumptions given below.

1. Dynamic counters in sender(3) and receiver(1)
2. Facilitates individual and cumulative acknowledgement
3. Supports 2 bits in frame so the sequence numbers should be counted as 0,1,2,3
4. An error should be induced to trigger resending of multiple packets

NOTE: *In this code, I have used 2 probability number constants P1 and P2 (in the receiver side code) to determine the probability with which the packets reach the receivers side. We can increase the probability of the packet being corrupted by increasing the value of P2.*

RECEIVER SIDE: Code

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include<time.h>
#include<stdlib.h>
#include<ctype.h>
#include<arpa/inet.h>

#define W 4
#define P1 50
#define P2 10

char a[10];
char b[10];
void alpha9(int);
int main()
{
   struct sockaddr_in ser,cli;
   int s,n,sock,i,j,c=1,f;
   unsigned int s1;
   s=socket(AF_INET,SOCK_STREAM,0);
   ser.sin_family=AF_INET;
   ser.sin_port=6500;
   ser.sin_addr.s_addr=inet_addr("127.0.0.1");
   bind(s,(struct sockaddr *) &ser, sizeof(ser));
   listen(s,1);
   n=sizeof(cli);
   sock=accept(s,(struct sockaddr *)&cli, &n);
   printf("\nTCP Connection Established.\n");
   s1=(unsigned int) time(NULL);
   srand(s1);
   strcpy(b,"Time Out ");
   recv(sock,a,sizeof(a),0);
   f=atoi(a);
   while(1)
   {
      for(i=0; i<W; i++)
      {
```

```
    recv(sock,a,sizeof(a),0);
    if(strcmp(a,b)==0)
        break;
}
i=0;
while(i<W)
{
    j=rand()%P1;
    if(j<P2)
    {
        send(sock,b,sizeof(b),0);
        break;
    }
    else
    {
        alpha9(c);
        if(c<=f)
        {
            printf("\nFrame %s Received ",a);
            send(sock,a,sizeof(a),0);
        }
        else
        {
            break;
        }
        c++;
    }
    if(c>f)
    {
                break;
            }
            i++;
        }
    }
    close(sock);
    close(s);
    return 0;
}
void alpha9(int z)
{
    int k,i=0,j,g;
    k=z;
    while(k>0)
    {
        i++;
        k=k/10;
    }
    g=i;
    i--;
    while(z>0)
    {
        k=z%10;
        a[i]=k+48;
        i--;
        z=z/10;
    }
    a[g]='\0';
}
```

RECEIVER SIDE: Screenshots

```
vmdhruv@ubuntu:~$ cd Documents
vmdhruv@ubuntu:~/Documents$ gcc -o goBackN_receiver goBackN_receiver.c
goBackN_receiver.c: In function ‘main’:
goBackN_receiver.c:78:5: warning: implicit declaration of function ‘close’ [-Wim
plicit-function-declaration]
     close(sock);
     ^
vmdhruv@ubuntu:~/Documents$ ./goBackN_receiver

TCP Connection Established.

Frame 1 Received
Frame 2 Received
Frame 3 Received
Frame 4 Received
Frame 5 Received
Frame 6 Received
Frame 7 Received
Frame 8 Received
Frame 9 Received
vmdhruv@ubuntu:~/Documents$
```

SENDER SIDE: Code

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include<time.h>
#include<stdlib.h>
#include<ctype.h>

#define W 4
char a[10];
char b[10];
void alpha9(int);
int main()
{
   int s,f,wl,c=1,x,i=0,j,n,p=0,e=0;
   struct sockaddr_in ser;
   s=socket(AF_INET,SOCK_STREAM,0);
   ser.sin_family=AF_INET;
   ser.sin_port=6500;
   ser.sin_addr.s_addr=inet_addr("127.0.0.1");
   connect(s,(struct sockaddr *) &ser, sizeof(ser));
   printf("\nTCP Connection Established.\n");
   printf("\nEnter the number of Frames: ");
   scanf("%d",&f);
   alpha9(f);
   send(s,a,sizeof(a),0);
   strcpy(b,"Time Out ");
   while(1)
   {
      for(i=0; i<W; i++)
      {
         alpha9(c);
         send(s,a,sizeof(a),0);
```

```
    if(c<=f)
    {
      printf("\nFrame %d Sent",c);
      c++;
    }
  }
  i=0;
  wl=W;
  while(i<W)
  {
    recv(s,a,sizeof(a),0);
    p=atoi(a);
    if(strcmp(a,b)==0)
    {
      e=c-wl;
      if(e<f)
      {
        printf("\nTime Out, Resent Frame %d
onwards",e);
      }
      break;
    }
    else
    {
      if(p<=f)
      {
        printf("\nFrame %s Acknowledged",a);
        wl--;
      }
      else
      {
        break;
      }
    }
      if(p>f)
      {
        break;
      }
      i++;
    }
    if(wl==0 && c>f)
    {
      send(s,b,sizeof(b),0);
      break;
    }
    else
    {
      c=c-wl;
      wl=W;
    }
  }
  printf("\n");
  close(s);
  return 0;
}
void alpha9(int z)
{
  int k,i=0,j,g;
  k=z;
  while(k>0)
  {
    i++;
    k=k/10;
  }
  g=i;
  i--;
  while(z>0)
  {
```

```
    k=z%10;
    a[i]=k+48;
    i--;
    z=z/10;
  }
  a[g]='\0';
}
```

SENDER SIDE: Screenshots

```
vmdhruv@ubuntu:~$ cd Documents
vmdhruv@ubuntu:~/Documents$ gcc -o goBackN_sender goBackN_sender.c
goBackN_sender.c: In function ‘main’:
goBackN_sender.c:24:25: warning: implicit declaration of function ‘inet_addr’ [-
Wimplicit-function-declaration]
     ser.sin_addr.s_addr=inet_addr("127.0.0.1");
                         ^
goBackN_sender.c:89:5: warning: implicit declaration of function ‘close’ [-Wimpl
icit-function-declaration]
     close(s);
     ^
vmdhruv@ubuntu:~/Documents$ ./goBackN_sender

TCP Connection Established.
```

```
Enter the number of Frames: 10

Frame 1 Sent
Frame 2 Sent
Frame 3 Sent
Frame 4 Sent
Frame 1 Acknowledged
Time Out, Resent Frame 2 onwards
Frame 2 Sent
Frame 3 Sent
Frame 4 Sent
Frame 5 Sent
Frame 2 Acknowledged
Frame 3 Acknowledged
Frame 4 Acknowledged
Time Out, Resent Frame 5 onwards
Frame 5 Sent
Frame 6 Sent
Frame 7 Sent
Frame 8 Sent
Frame 5 Acknowledged
Time Out, Resent Frame 6 onwards
Frame 6 Sent
Frame 7 Sent
Frame 8 Sent
Frame 9 Sent
Frame 6 Acknowledged
Time Out, Resent Frame 7 onwards
Frame 7 Sent
Frame 8 Sent
Frame 9 Sent
Frame 10 Sent
Frame 7 Acknowledged
Frame 8 Acknowledged
Frame 9 Acknowledged
Frame 10 Acknowledged
vmdhruv@ubuntu:~/Documents$
```