

LAB 5
DHARMITHA S SHETTY
200948030

WAP to construct a RouteRequest and a RouteReply packets for a given network and display the same. Also, WAP that checks whether a receiving node must forward the RouteRequest packet or reply with a RouteReply packet.

```
#include<stdio.h>

#include<string.h>

int main()

{

    int seq, src, dest, ttl,type;

    int n, a[10];

    int node;

    char a1[100][12];


    printf("Enter the type of the packet(1/2/3)\n");

    scanf("%d",&type);

    if(type==1)

    {

        printf("Enter the route request packet\n");

        printf("Enter the source node, sequence number, destination node & Ttl:\n");

        scanf("%d %d %d %d", &src, &seq, &dest, &ttl);


        printf("\nRoute Request Packet :\n");

        printf("Type\tSource\tSeq No.\tDest\tTTL\n");

        printf("%d\t%d\t%d\t%d\t%d",type, src, seq, dest, ttl);

        printf("\n");
```

```
printf("\nEnter the number of intermediate nodes to reach the destination:");  
scanf("%d", &n);
```

```
printf("\nEnter the address of %d hops:\n",n);  
int i;  
for(i = 0; i < n; i++)  
    scanf("%d", &a[i]);
```

```
printf("\nRoute Reply packet\n");  
printf("\nSourceaddr:\t%d\n", src);  
for(i = 0; i < n; i++)  
    printf("%d hop:\t\t%d\n",i, a[i]);  
printf("Destaddr:\t%d\n", dest);
```

```
int k=1,flag_add_exist=0;  
int nnode;
```

```
while(k)  
{
```

```
    printf("\nEnter the address of a receiving node:\n");  
    scanf("%d", &node);  
    if(node==src)  
        nnode=0;  
    else if(node == dest)  
        nnode=n+1;  
    else
```

```
{  
for(i = 0; i < n; i++)  
    if(a[i]==node)  
{  
nnode=i+1;  
break;  
}  
}  
if(nnode<=ttl)  
{  
    for(i = 0; i < n; i++)  
        if(a[i]==node)  
        {  
flag_add_exist=1;  
break;  
}  
if(node==src || node==dest || flag_add_exist==1)  
{  
    if(node == dest)  
        printf("\nReply with Route Reply packet\n");  
    else  
        printf("\nForward the Route Request packet\n");  
}  
else  
    printf("404 not found\n");  
flag_add_exist =0;  
}  
else{
```

```

printf("TTL Time Out\nDiscard the Packet\n");
}

printf("\n1 to continue 0 to exit\n");
scanf("%d",&k);
}
}

if(type==2)
{
printf("Enter the route reply packet\n");
printf("Enter the source node, sequence number, destination node & Ttl:\n");
scanf("%d %d %d %d", &src, &seq, &dest, &ttl);
printf("Enter the number of intermediate nodes\n");
scanf("%d",&n);

int i,j;
for(i=0;i<n;i++)
{
printf("Enter the IP address of Node %d\n",i+1);
for(j=0;j<12;j++)
scanf("%c",&a1[i][j]);

}

printf("The Route Reply Packet:\n");
printf("Type\tSource\tSeq No.\tDest\tTTL\n");
printf("%d\t%d\t%d\t%d\t%d",type, src, seq, dest, ttl);
printf("\n");
for(i=0;i<n;i++)
{
printf("IP address of Node %d",i+1);

```

```
for(j=0;j<12;j++)  
    printf("%c",a1[i][j]);  
    printf("\n");  
}  
}  
if(type==3)  
    printf("Error Closing....\n");  
return 0;  
}
```

Output:

D:\200948030\1DSR\bin\Debug\1DSR.exe

Enter the type of the packet(1/2/3)

1

Enter the route request packet

Enter the source node, sequence number, destination node & Ttl:

0 3 2 5

Route Request Packet :

Type	Source	Seq No.	Dest	TTL
1	0	3	2	5

Enter the number of intermediate nodes to reach the destination:2

Enter the address of 2 hops:

1 2

Route Reply packet

Sourceaddr: 0

0 hop: 1

1 hop: 2

Destaddr: 2

Enter the address of a receiving node:

0

Forward the Route Request packet

1 to continue 0 to exit

1

Enter the address of a receiving node:

3

404 not found

1 to continue 0 to exit

1

Enter the address of a receiving node:

0

Forward the Route Request packet

1 to continue 0 to exit

D:\200948030\1DSR\bin\Debug\1DSR.exe

Enter the type of the packet(1/2/3)

2

Enter the route reply packet

Enter the source node, sequence number, destination node & Ttl:

0 2 3 1

Enter the number of intermediate nodes

2

Enter the IP address of Node 1

125.33.12.5

Enter the IP address of Node 2

221.2.54.7

The Route Reply Packet:

Type	Source	Seq No.	Dest	TTL
2	0	2	3	1

IP address of Node 1

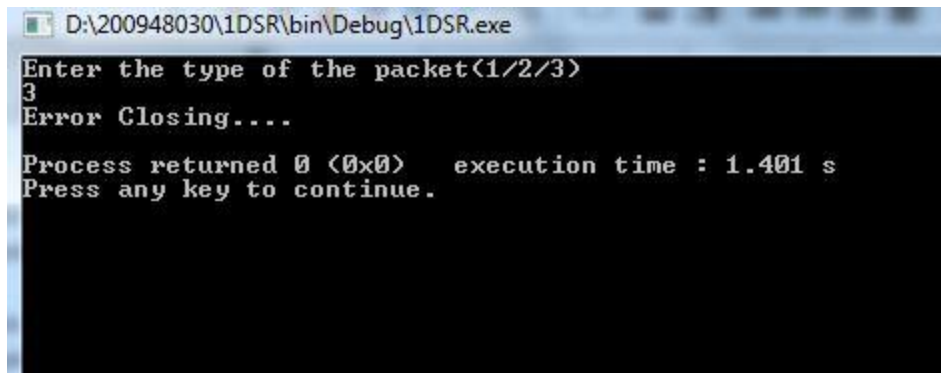
125.33.12.5

IP address of Node 2

221.2.54.7

Process returned 0 (0x0) execution time : 51.749 s

Press any key to continue.



```
D:\200948030\1DSR\bin\Debug\1DSR.exe
Enter the type of the packet<1/2/3>
3
Error Closing....
Process returned 0 (0x0) execution time : 1.401 s
Press any key to continue.
```

WAP to implement exponential backoff algorithm to avoid frequent RouteRequest flooding in the network when the destination is in another disjoint set.

```
#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()
{
    int ch,b,e,p,i,ch1;

    printf("sent request packet? 1 for yes and 0 for no: \n");

    scanf("%d",&ch);

    if(ch==0)

        exit(0);

    printf("start exponential backoff algorithm: \n");

    printf("enter base and exponent:\n");

    scanf("%d%d",&b,&e);

    for(i=0;i<=e;i++)
    {
        p=pow(b,i);

        printf("%d seconds passed\n",p);

        printf("recieved reply packet? 1 for yes and 0 for no:\n");

        scanf("%d",&ch1);
```

```
    if(ch1==1)
    {
        printf("reply packet recieved\n");
        exit(0);
    }
}

printf("destination node is isolated\n");
return 0;
}
```

Output:


```
D:\200948030\2DSR\bin\Debug\2DSR.exe
sent request packet? 1 for yes and 0 for no:
1
start exponential backoff algorithm:
enter base and exponent:
3 6
1 seconds passed
recieved reply packet? 1 for yes and 0 for no:
0
3 seconds passed
recieved reply packet? 1 for yes and 0 for no:
0
9 seconds passed
recieved reply packet? 1 for yes and 0 for no:
0
27 seconds passed
recieved reply packet? 1 for yes and 0 for no:
0
81 seconds passed
recieved reply packet? 1 for yes and 0 for no:
0
243 seconds passed
recieved reply packet? 1 for yes and 0 for no:
0
729 seconds passed
recieved reply packet? 1 for yes and 0 for no:
0
destination node is isolated
Process returned 0 (0x0)   execution time : 17.495 s
Press any key to continue.
```

```
D:\200948030\2DSR\bin\Debug\2DSR.exe
sent request packet? 1 for yes and 0 for no:
1
start exponential backoff algorithm:
enter base and exponent:
2 6
1 seconds passed
recieved reply packet? 1 for yes and 0 for no:
0
2 seconds passed
recieved reply packet? 1 for yes and 0 for no:
0
4 seconds passed
recieved reply packet? 1 for yes and 0 for no:
0
8 seconds passed
recieved reply packet? 1 for yes and 0 for no:
0
16 seconds passed
recieved reply packet? 1 for yes and 0 for no:
0
32 seconds passed
recieved reply packet? 1 for yes and 0 for no:
0
64 seconds passed
recieved reply packet? 1 for yes and 0 for no:
0
destination node is isolated
Process returned 0 (0x0)   execution time : 16.019 s
Press any key to continue.
```