

Computer Network Laboratory

Assignment 3

Name: Hemant Singh

Enrollment Number: 17114038

Class: 3rd year, B.Tech CSE

Course: CSN-361



Github Link: <https://github.com/hemant84/CSN-361>

Problem Statements:

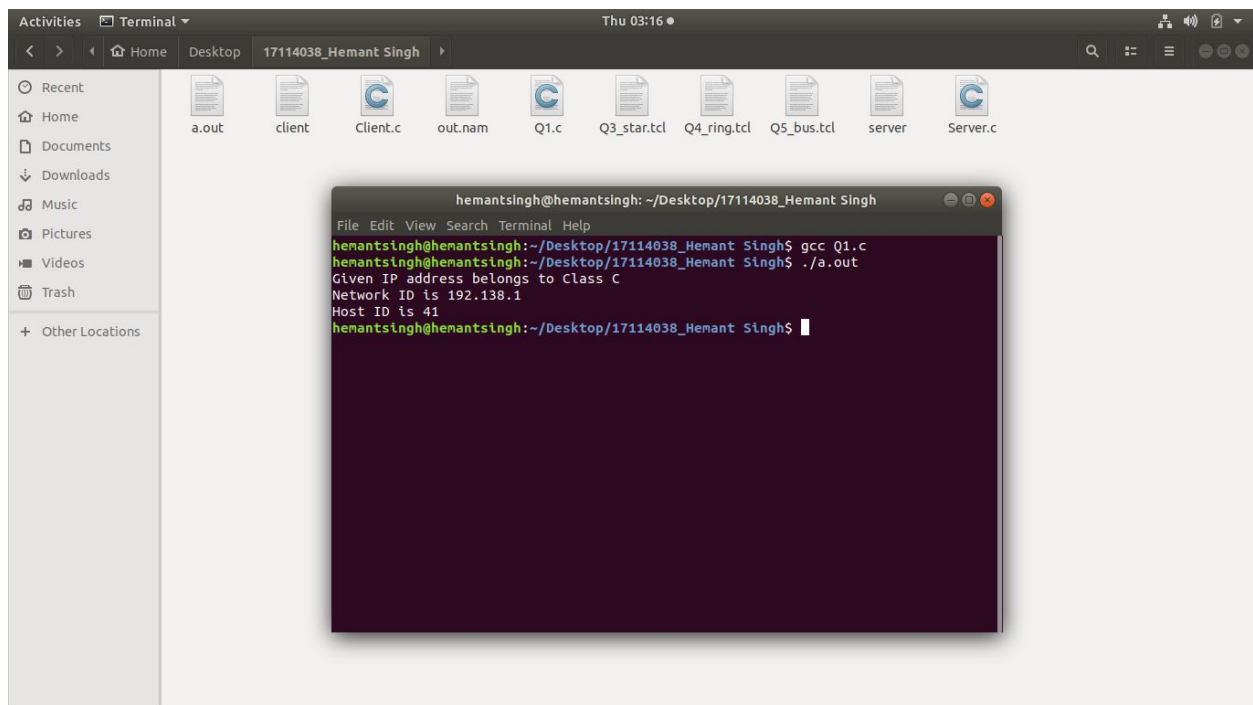
Problem 1 :

Write a socket program in C to determine class, Network and Host ID of an IPv4 address.

Algorithms used :

1. **For determining the class:** The idea is to check first octet of IP address. As we know, for class **A** first octet will range from **1 – 126**, for class **B** first octet will range from **128 – 191**, for class **C** first octet will range from **192- 223**, for class **D** first octet will range from **224 – 239**, for class **E** first octet will range from **240 – 255**.
2. **For determining the Network and Host ID:** We know that Subnet Mask for Class **A** is **8**, for Class **B** is **16** and for Class **C** is **24** whereas Class **D** and **E** is not divided into Network and Host ID.

Screenshot :



Problem 2 :

Write a C program to demonstrate File Transfer using UDP.

Algorithms used :

1. `int sockfd = socket(domain, type, protocol)`
2. `int setsockopt(int sockfd, int level, int optname, const void *optval, socklen_t optlen);`
3. `int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen);`
4. `int listen(int sockfd, int backlog);`
5. `int new_socket= accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);`
6. `int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen);`

Data structures used :

- Int, char *, char []: To store the socket , strings, buffer
- struct sockaddr_in : for storing the port number and creating an instance of client and server

Screenshot :

```
hemantsingh@hemantsingh: ~/Desktop/17114038_Hemant Singh
File Edit View Search Terminal Help
hemantsingh@hemantsingh:~/Desktop/17114038_Hemant Singh$ gcc Server.c -o server
hemantsingh@hemantsingh:~/Desktop/17114038_Hemant Singh$ ./server
file descriptor 3 received
Successfully binded!
Waiting for file name...
File Name Received: Q1.c
File Successfully opened!
Waiting for file name...
[ ]

hemantsingh@hemantsingh: ~/Desktop/17114038_Hemant Singh
File Edit View Search Terminal Help
printf("Network ID is %s\n", network);
printf("Host ID is %s\n", host);
}
else
printf("In this Class, IP address is not"
" divided into Network and Host ID\n");
}
int main()
{
char str[] = "192.138.1.41";
char ipClass = findClass(str);
printf("Given IP address belongs to Class %c\n",
ipClass);
separate(str, ipClass);
return 0;
}
-----
Please enter file name to receive:
```

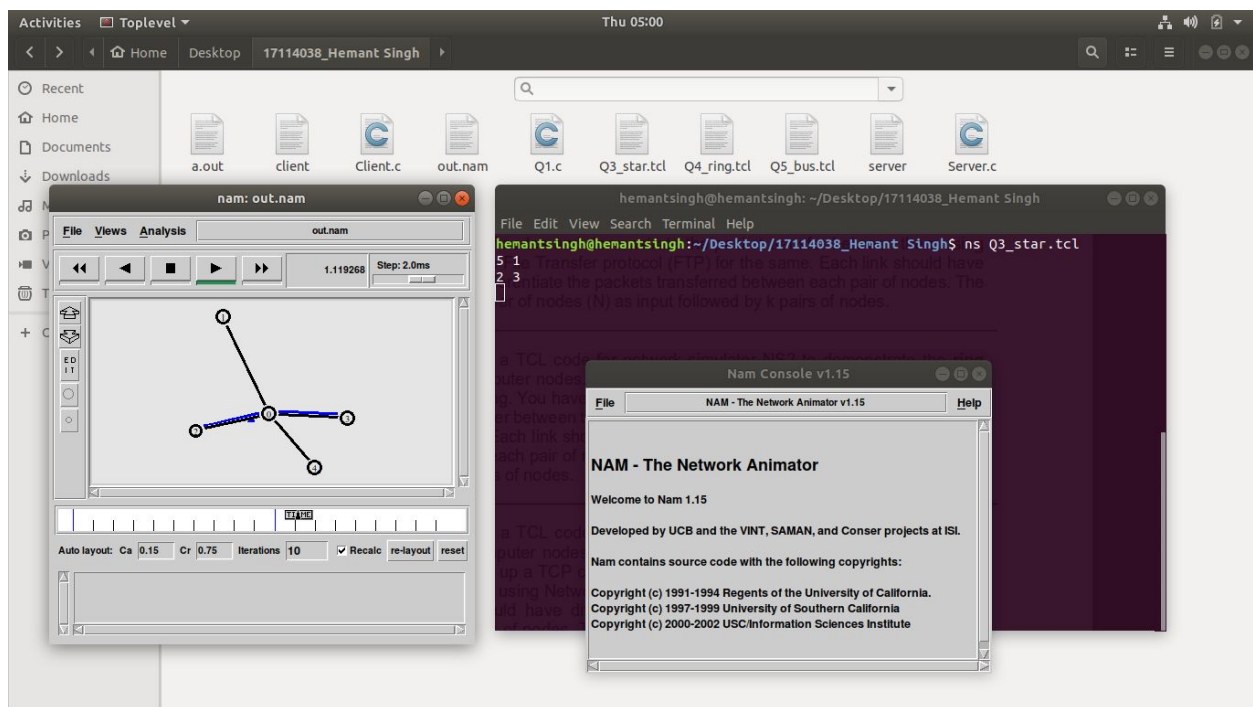
Problem 3 :

Write a TCL code for network simulator NS2 to demonstrate the star topology among a set of computer nodes. Given N nodes, one node will be assigned as the central node and the other nodes will be connected to it to form the star. You have to set up a TCP connection between k pairs of nodes and demonstrate the packet transfer between them using Network Animator (NAM). Use File Transfer protocol (FTP) for the same. Each link should have different color of packets to differentiate the packets transferred between each pair of nodes. The program should take the number of nodes (N) as input followed by k pairs of nodes.

Algorithms used :

- Easy to detect faults and to remove parts.
- Requires more cable length than a linear bus topology.

Screenshot :





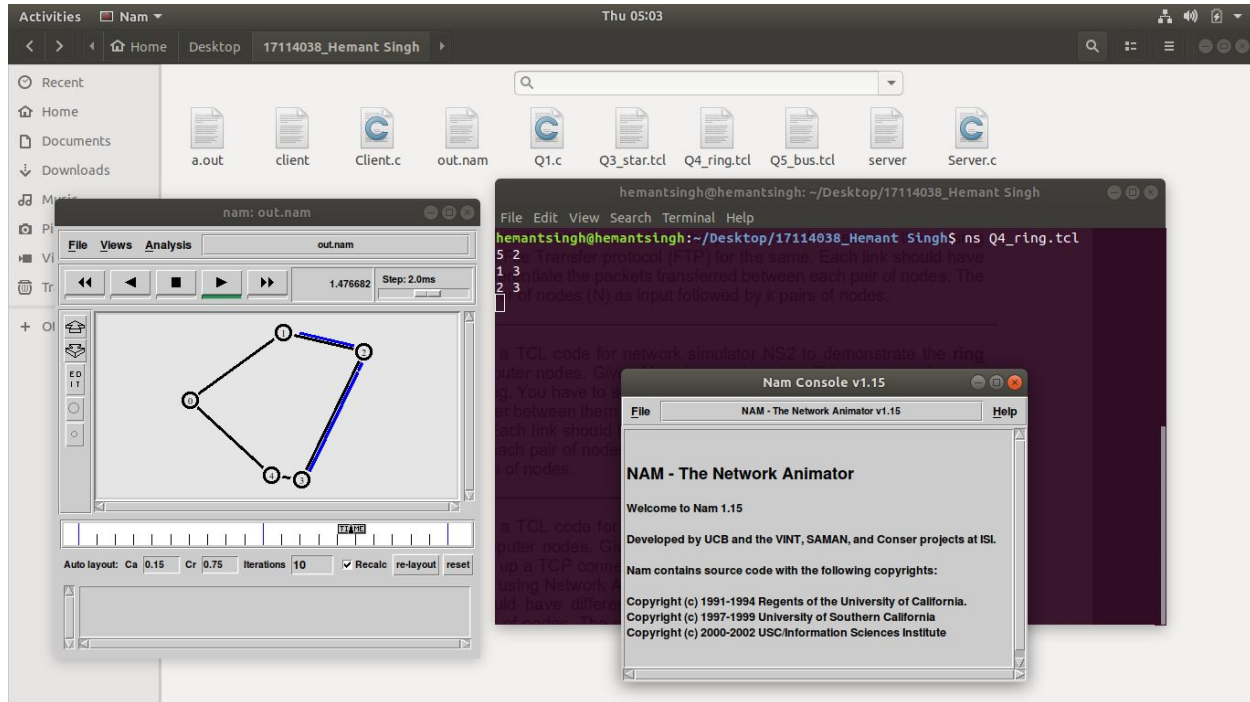
Problem 4 :

Write a TCL code for network simulator NS2 to demonstrate the ring topology among a set of computer nodes. Given N nodes, each node will be connected to two other nodes in the form of a ring. You have to set up a TCP connection between k pairs of nodes and demonstrate packet transfer between them using Network Animator (NAM). Use File Transfer protocol (FTP) for the same. Each link should have different color of packets to differentiate the packets transferred between each pair of nodes. The program should take the number of nodes (N) as input followed by k pairs of nodes.

Algorithms used :

- All data flows in one direction, reducing the chance of packet collisions.
- A network server is not needed to control network connectivity between each workstation.
- Data can transfer between workstations at high speeds.

Screenshot :



Problem 5 :

Write a TCL code for network simulator NS2 to demonstrate the bus topology among a set of computer nodes. Given N nodes, each node will be connected to a common link. You have to set up a TCP connection between k pairs of nodes and demonstrate packet transfer between them using Network Animator (NAM). Use File Transfer protocol (FTP) for the same. Each link should have different color of packets to differentiate the packets transferred between each pair of nodes. The program should take the number of nodes (N) as input followed by k pairs of nodes.

Algorithms used :

- Easy to connect a computer or peripheral to a linear **bus**.
- Requires less cable length than a star **topology**.
- Entire **network** shuts down if there is a break in the main cable.
- Terminators are required at both ends of the backbone cable.

Screenshot :

