MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

Mini Project Report
of
Computer Networks LAB

STATIC ROUTING DEMO

SUBMITTED BY

NAGAM VENKATA MANOJ KUMAR - 200905262
(B section, Roll.No : 48, 5th sem, CSE)

SURAPURAJU ROHITH - 200905290
(B section, Roll.No : 52, 5th sem, CSE)

ABSTRACT

Our Mini-Project involves the demo of static routing. We have designed a demo that gives us all the details regarding path travelled from source to destination routers. User can give his source router and his destination router. Along with that user can also provide intermediate routers according to his choice.

Here are few assumptions which are made.

1. We are assuming that the network is having 4 routers which are connected with each other in a ring topology manner.
2. The above network is fixed.
3. IP address of the nodes that are connected to the router will not change in the future.

Firstly, the number of networks that are connected to each router in the ring topology manner are taken.

Secondly, IP addresses of all the networks are taken as input. Validation of each IP addresses provided is done by the program.

Now, the program automatically detects for the intermediate routers and asks to provide the details if any. Finally, after collecting all the necessary details it will display all the information regarding source router, destination router, intermediate routers, static route followed and success state of the process.

The complete program used for the static routing demo is developed in C language.

TABLE OF CONTENTS

INTRODUCTION

3.1 STATIC ROUTING

3.1.1 What is Static Routing?

Static routing is a form of routing that occurs when a router uses a manually configured routing entry, rather than information from dynamic routing traffic. In many cases, static routes are manually configured by a network administrator. Static routing leaves the network administrator with full control over the routing behavior of the network.

Unlike dynamic routing, static routes are fixed and do not change if the network is changed or reconfigured. Static routing can be used for small networks that require only one or two routes. This is often more efficient since a link is not being wasted by exchanging dynamic routing information.

3.1.2 Advantages of Static Routing

Static Routing is very easy to configure on small networks. The path that static routing takes to the destination is very predictable. Even if there is a change in the network design and layout, there won't be any changes in the router. The users always know where the path is going to be.

Static routing requires a smaller number of resources. Extra resources such as CPU and memory are not needed here. Static routing also provides better security and protection.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

3.2.1 Hardware Requirements

Hardware requirements needed for implementing this static routing demo are a laptop working in a good condition or any desktop which is working well.

3.2.2 Software Requirements

Software requirements needed for the implementation of this static routing demo are gcc compiler pre-installed in the device used for the demo. This gcc compiler is used for compiling and executing C program. This code can be compiled in either Linux or windows terminal. This code can also be executed in any IDEs.

PROBLEM DEFINITION

This project is to implement the static routing demo and giving the information about the routing details. In order to implement the static routing demo, we need four routers connected with each other in ring topology and each router is connected to multiple networks

For each of these networks our project requires user to provide the valid IP addresses. Validation of the given IP addresses is conducted by the project. It now requires the source and destination of the routing to be done. It also prompts if any intermediate routers are required for continuing the process. After taking all the necessary inputs it will provide the detailed information about the static routing demo.

OBJECTIVES

Our objectives for this mini-project are as followed

1. To show the demo of static routing where the routers are connected in a ring topology manner.
2. Validating the IP addresses of the networks provided by the user that are connected to the routers.
3. To prompt the user whether to give shortest path for routing between the routers if there is a direct connection between those two routers.
4. Ask for any intermediate routers if required.
5. Route through the path that is decided by the user, by asking through which intermediate routers user want to travel.
6. To display all the information about routing which is done in the process.
7. Finally, to give a clear demo of how static routing is implemented and provide good understanding about it.

METHODOLOGY

There are different types of methodologies that we used in building this static routing demo project. The main method we used for this project implementation is connecting 4 routers in the ring topology manner (i.e., all the four routers are connected to each other in the form of a ring).

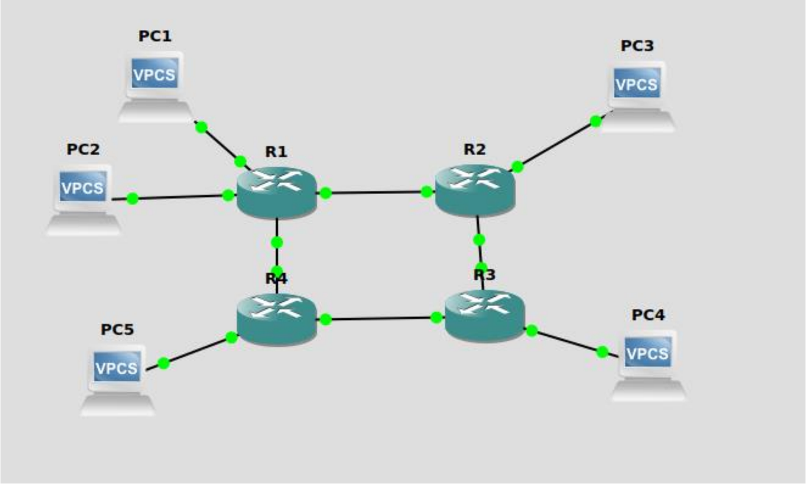


Figure 1.1: Example diagram for ring topology connection.

Secondly, we have conducted the validation of different IP addresses provided by the user through the function *validate_ip*. This function will take character string as the input. Then the string is cut into different sub strings using a dot delimiter (i.e '.').

Each of these sub strings are given as input to the function *validate_number*. This function will validate whether the given input a number or character. It will consider only digits and rejects the characters. Finally, it analyses the whole address and accept if it is valid.

Another method which is used for printing the total intermediate routers used for the transmission of data is done by the function *concat*. This function takes two integers as input. Convert them into string and concat both of them. Returns back the integer format of the concatenated string.

Finally, in the main function we take all the required inputs from the users. These inputs are passed to the above functions as parameters for validating purposes. Also, the inputs passed to functions like *concat* will help in updating the list of intermediate routers added newly every time.

IMPLEMENTATION DETAILS

This project is implemented in a terminal in ubuntu which has a pre-installed GCC complier for executing C programs. This project is completely coded in C language.

We have used different types of structures for multiple purposes.

1. *arr* : This is a 1D array of structure node which is having 2D array of character to store the given IP addresses for each router.
2. *n* : This is a 2D array of size 16 and type int is used to save the links between routers and intern save the structure (i.e., ring topology) in which they are connected.
3. *sc* : Pointer of type char to store IP addresses given by the user for networks. This is used for calculating whether the IP address id valid.
4. *sd* : To store the data of inter mediate step.
5. *im* : Used to give the final static route followed in the process.

We have also used two variables *check0* and *count0* where *check0* is used to declare the success state of the process. *count0* is used as the index for *im* in the program.

This program will need different types of inputs for successful routing purpose. At the beginning, we give the total no of networks connected to different routers. Then, we will provide the IP addresses of different networks connected to different routers. Validation of IP addresses will be done.

Now it prompts the user to give the source router from where user wants to start the travel. It again checks whether the given source address is among the priorly given network addresses. After checking, it automatically detects to which router does this source network belong to. Then, it asks for destination address and the same process mentioned above will be performed.

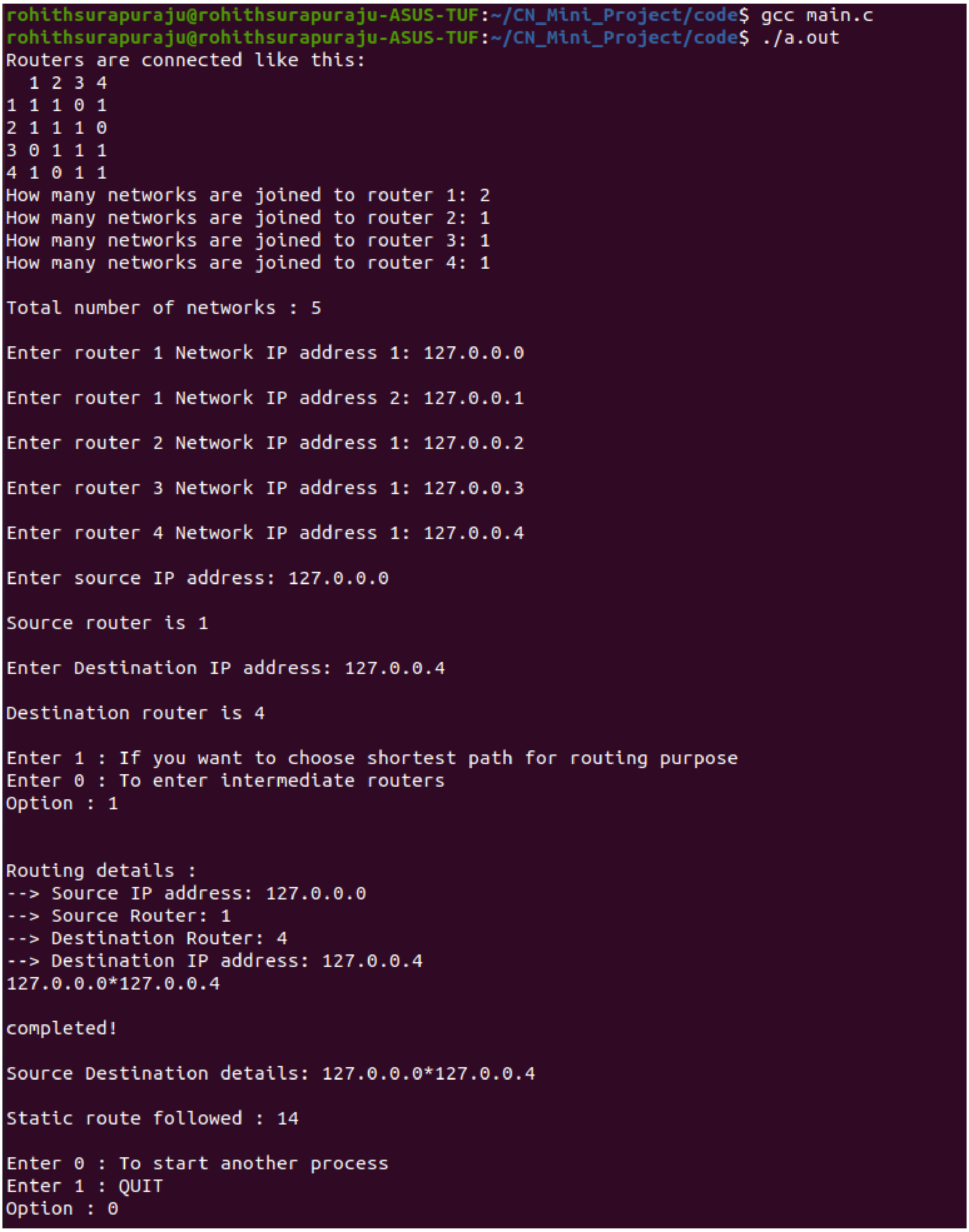
It now asks whether the user requires shortest path if there is a direct edge between the source and destination router. Otherwise, it prompts to give the intermediate routers for continuing the process. Finally, it displays all the information about static routing done in the program. User can continue new process with different source and destination addresses thereafter.

OUTPUT

We have included different types of cases below.

Case 1:

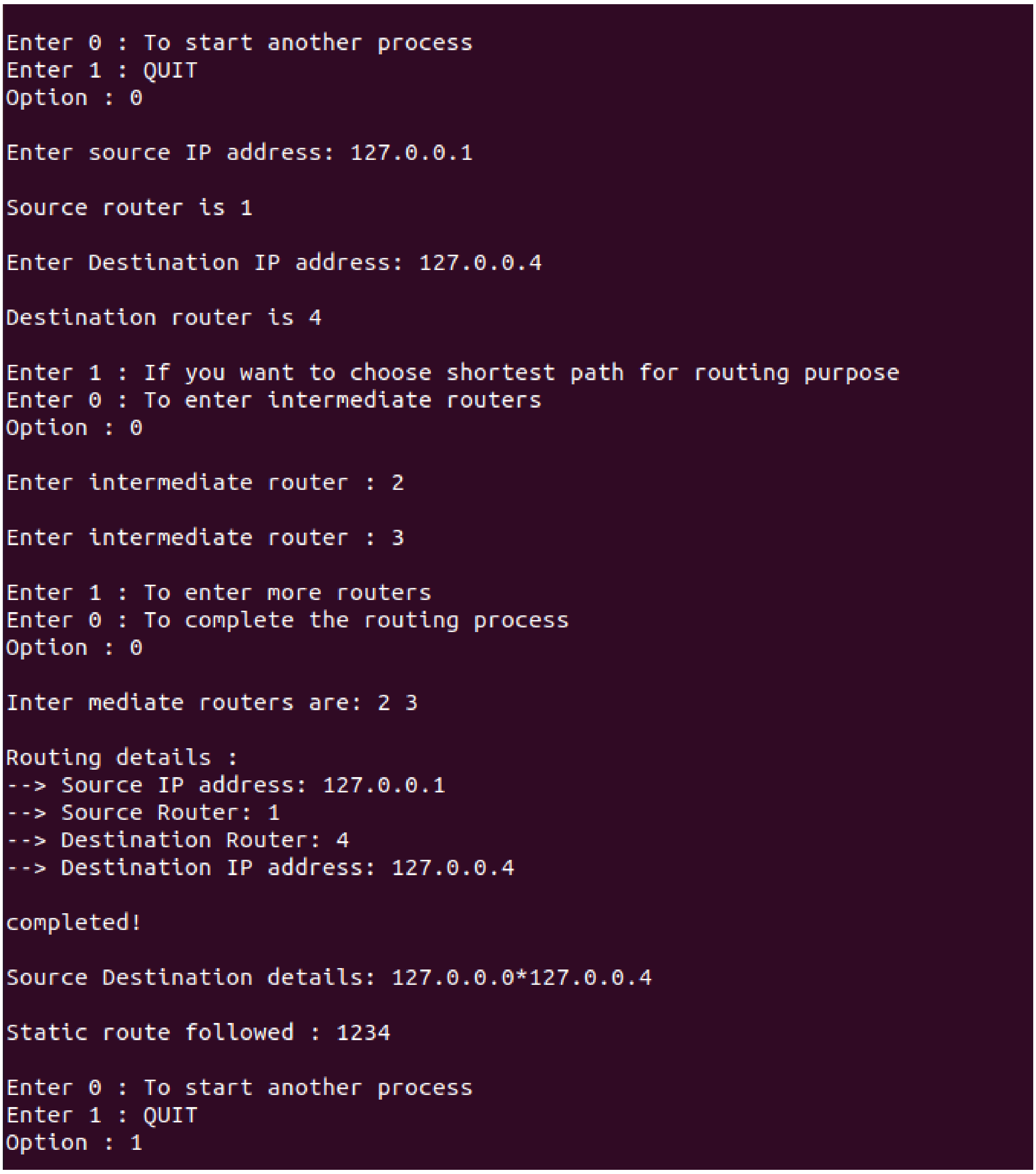
From router 1 to 4 directly.



```
rohithsurapuraju@rohithsurapuraju-ASUS-TUF:~/CN_Mini_Project/code$ gcc main.c
rohithsurapuraju@rohithsurapuraju-ASUS-TUF:~/CN_Mini_Project/code$ ./a.out
Routers are connected like this:
  1 2 3 4
1 1 1 0 1
2 1 1 1 0
3 0 1 1 1
4 1 0 1 1
How many networks are joined to router 1: 2
How many networks are joined to router 2: 1
How many networks are joined to router 3: 1
How many networks are joined to router 4: 1

Total number of networks : 5

Enter router 1 Network IP address 1: 127.0.0.0

Enter router 1 Network IP address 2: 127.0.0.1

Enter router 2 Network IP address 1: 127.0.0.2

Enter router 3 Network IP address 1: 127.0.0.3

Enter router 4 Network IP address 1: 127.0.0.4

Enter source IP address: 127.0.0.0

Source router is 1

Enter Destination IP address: 127.0.0.4

Destination router is 4

Enter 1 : If you want to choose shortest path for routing purpose
Enter 0 : To enter intermediate routers
Option : 1


Routing details :
--> Source IP address: 127.0.0.0
--> Source Router: 1
--> Destination Router: 4
--> Destination IP address: 127.0.0.4
127.0.0.0*127.0.0.4

completed!

Source Destination details: 127.0.0.0*127.0.0.4

Static route followed : 14

Enter 0 : To start another process
Enter 1 : QUIT
Option : 0
```

Case 2:

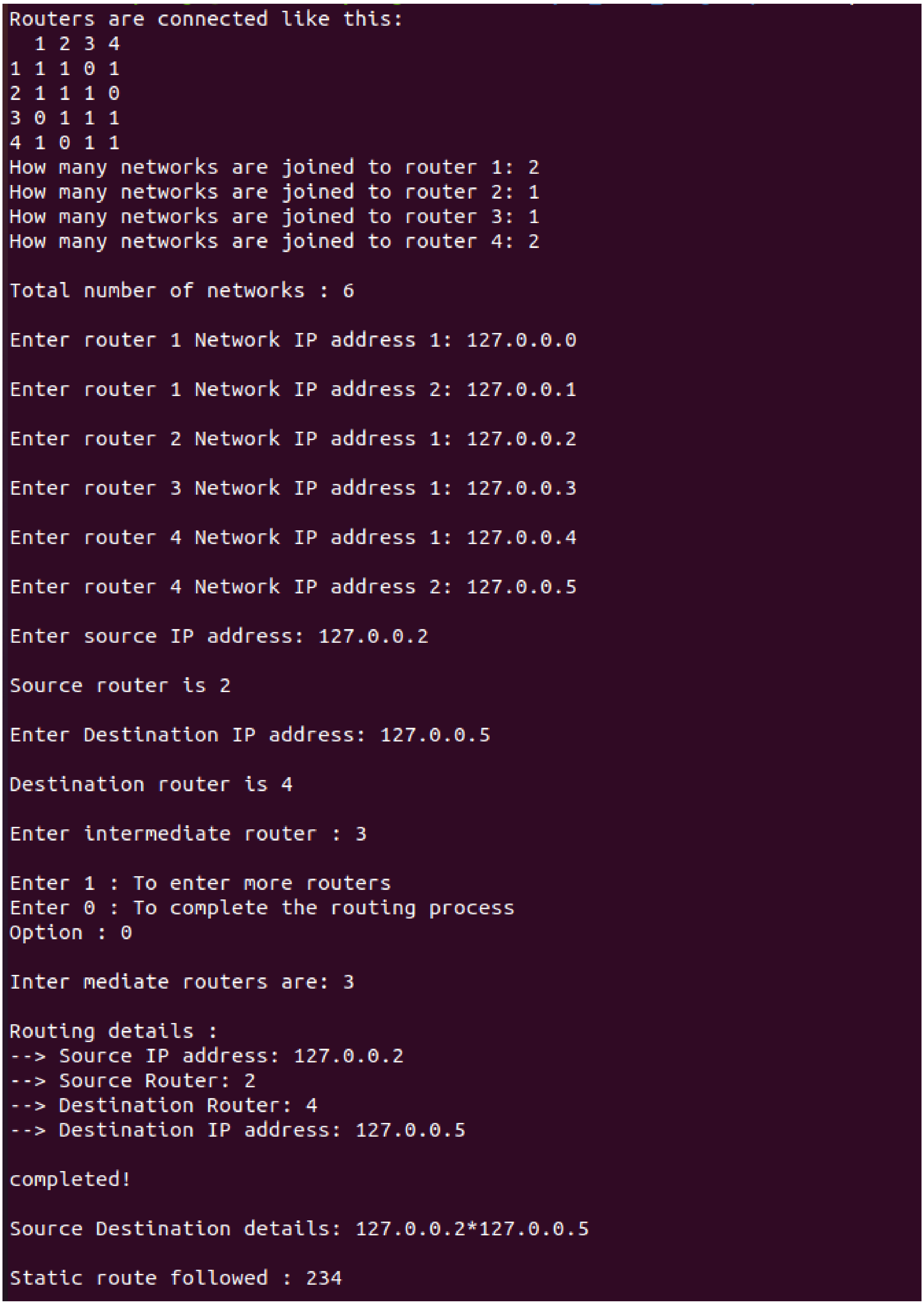
From router 1 to 4 through intermediate routers 2 and 3.



```
Enter 0 : To start another process
Enter 1 : QUIT
Option : 0

Enter source IP address: 127.0.0.1

Source router is 1

Enter Destination IP address: 127.0.0.4

Destination router is 4

Enter 1 : If you want to choose shortest path for routing purpose
Enter 0 : To enter intermediate routers
Option : 0

Enter intermediate router : 2

Enter intermediate router : 3

Enter 1 : To enter more routers
Enter 0 : To complete the routing process
Option : 0

Inter mediate routers are: 2 3

Routing details :
--> Source IP address: 127.0.0.1
--> Source Router: 1
--> Destination Router: 4
--> Destination IP address: 127.0.0.4

completed!

Source Destination details: 127.0.0.0*127.0.0.4

Static route followed : 1234

Enter 0 : To start another process
Enter 1 : QUIT
Option : 1
```

Case 3:

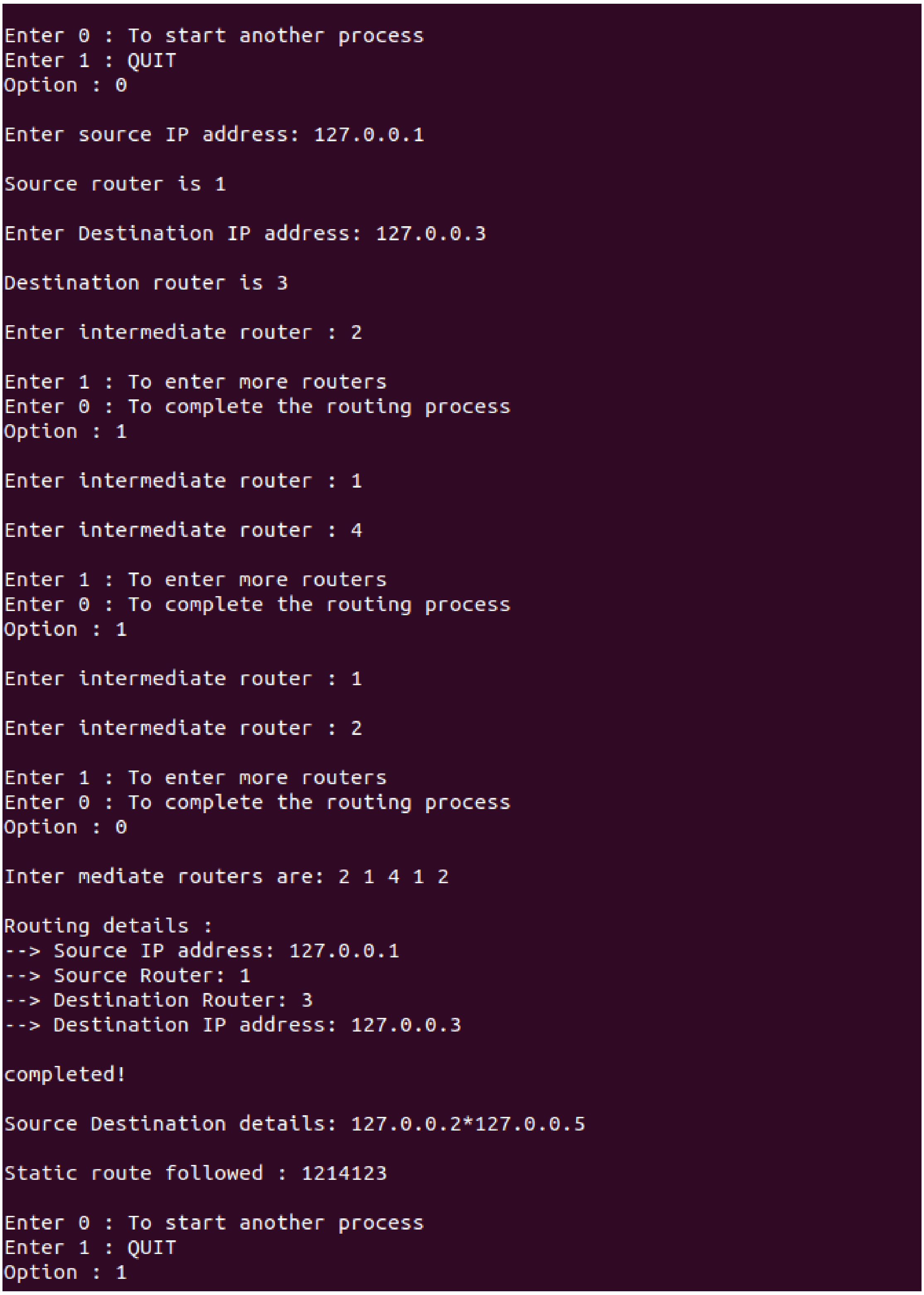
From router 2 to 4 through intermediate router 3.



```
Routers are connected like this:
  1 2 3 4
1 1 1 0 1
2 1 1 1 0
3 0 1 1 1
4 1 0 1 1
How many networks are joined to router 1: 2
How many networks are joined to router 2: 1
How many networks are joined to router 3: 1
How many networks are joined to router 4: 2

Total number of networks : 6

Enter router 1 Network IP address 1: 127.0.0.0

Enter router 1 Network IP address 2: 127.0.0.1

Enter router 2 Network IP address 1: 127.0.0.2

Enter router 3 Network IP address 1: 127.0.0.3

Enter router 4 Network IP address 1: 127.0.0.4

Enter router 4 Network IP address 2: 127.0.0.5

Enter source IP address: 127.0.0.2

Source router is 2

Enter Destination IP address: 127.0.0.5

Destination router is 4

Enter intermediate router : 3

Enter 1 : To enter more routers
Enter 0 : To complete the routing process
Option : 0

Inter mediate routers are: 3

Routing details :
--> Source IP address: 127.0.0.2
--> Source Router: 2
--> Destination Router: 4
--> Destination IP address: 127.0.0.5

completed!

Source Destination details: 127.0.0.2*127.0.0.5

Static route followed : 234
```

Case 4:

From router 1 to 3 through multiple intermediate routers 2, 1, 4, 1, 2.



```
Enter 0 : To start another process
Enter 1 : QUIT
Option : 0

Enter source IP address: 127.0.0.1

Source router is 1

Enter Destination IP address: 127.0.0.3

Destination router is 3

Enter intermediate router : 2

Enter 1 : To enter more routers
Enter 0 : To complete the routing process
Option : 1

Enter intermediate router : 1

Enter intermediate router : 4

Enter 1 : To enter more routers
Enter 0 : To complete the routing process
Option : 1

Enter intermediate router : 1

Enter intermediate router : 2

Enter 1 : To enter more routers
Enter 0 : To complete the routing process
Option : 0

Inter mediate routers are: 2 1 4 1 2

Routing details :
--> Source IP address: 127.0.0.1
--> Source Router: 1
--> Destination Router: 3
--> Destination IP address: 127.0.0.3

completed!

Source Destination details: 127.0.0.2*127.0.0.5

Static route followed : 1214123

Enter 0 : To start another process
Enter 1 : QUIT
Option : 1
```

CONTRIBUTION SUMMARY

Rohith Surapuraju

- Validation of IP addresses, Taking primary inputs and checking whether source and destination IP addresses are present in the given input, checking for the shortest path and prompting the user necessarily.

Nagam Venakat Manoj Kumar

- Concatenation of strings, after checking the shortest path prompting the user for providing intermediate routers, updating the intermediate router details and error correction.

Design layout and Documentation – both

REFERENCES

1. James F. Kurose and Keith W. Ross COMPUTER NETWORKING: A Top-Down Approach
2. https://en.wikipedia.org/wiki/Static_routing