



**Department of Computer
Science and Engineering**

Advanced Computer Networks

UE16CS346

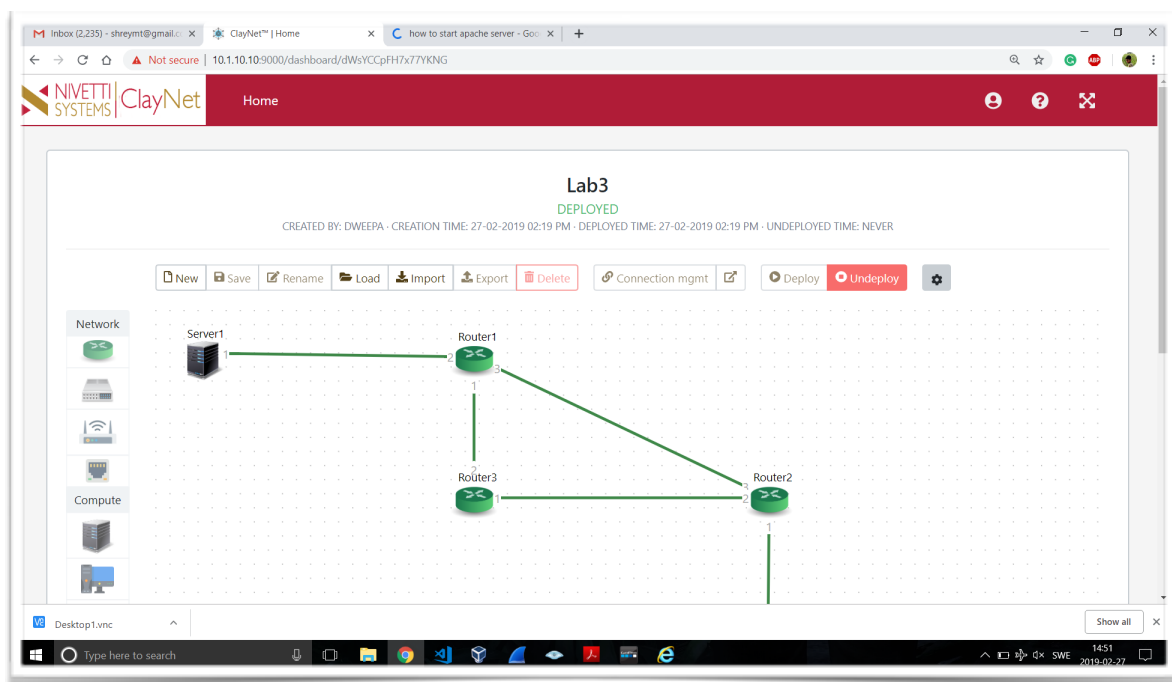
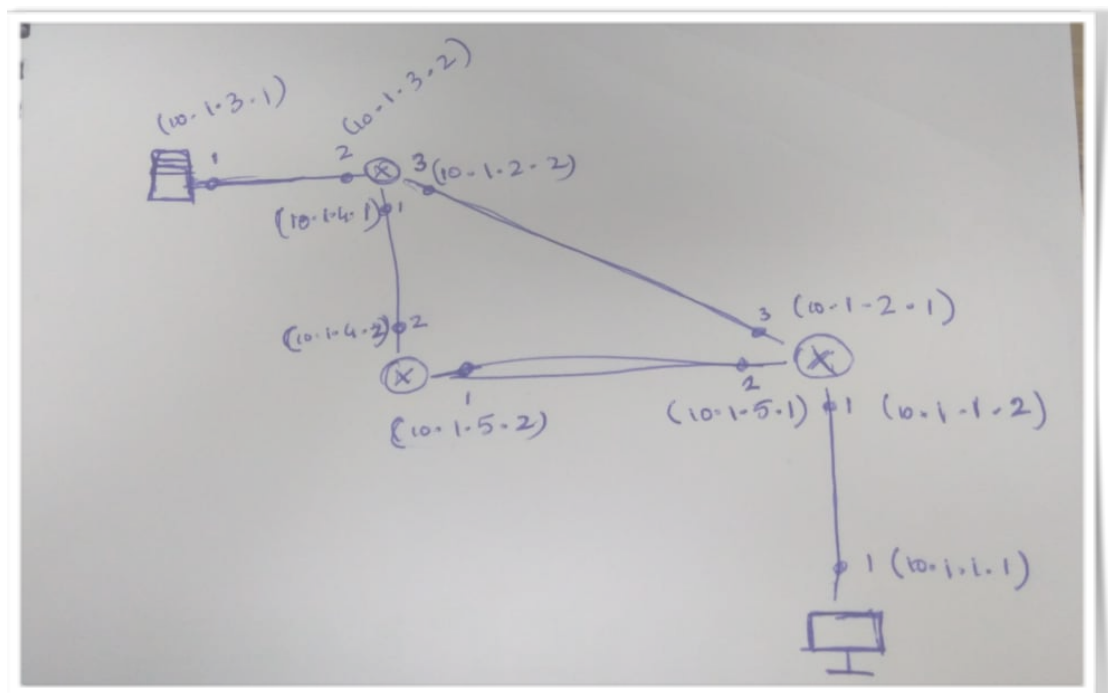
Lab Assignment 3

Dweepa Prasad	01FB16ECS138
Ishita Bhandari	01FB16ECS143
Shashank Prabhakar	01FB16ECS356
Shrey Tiwari	01FB16ECS368

Introduction

- This experiment focuses on bringing forth a flavour of how network setup occurs in real life. It also drives home the point as to why static routing is hard and not very reliable.
- To carry out this experiment we use the ClayNet software and set up the topology as shown.
- As can be seen in the topology, one packet can take up many paths while traversing the network from one system to the other.
- As a part of the experiment, we are also required to analyse the network parameters such as bandwidth and throughput. We do this using the "iperf" command.

• Topology



Design

- First, we draw the topology and identify all the networks present.
- We assign each network a unique network address.
- Then we configure every network interface in the network and assign it an ip address
- After we finish the configuration of IP addresses, we move on to creating the static routing entries.
- We create a logical table for each router as shown
- We then deploy the topology and move on entering the static routing table entries

** Router Near Hosts*

Destination	Next Hop	Name
10.1.3.0/24	10.1.2.2	R1.1
10.1.3.0/24	10.1.5.2	R1.2
10.1.4.0/24	10.1.2.2	R1.3
10.1.4.0/24	10.1.5.2	R1.4

** Router Near Server:*

Destination	Next Hop	Name
10.1.1.0/24	10.1.2.1	R1.1
10.1.1.0/24	10.1.4.2	R1.2
10.1.5.0/24	10.1.2.1	R1.3
10.1.5.0/24	10.1.4.2	R1.4

** Other Router:*

Destination	Next Hop	Name
10.1.1.0/24	10.1.4.1	R1.1
10.1.1.0/24	10.1.5.1	R1.2
10.1.2.0/24	10.1.4.1	R1.3
10.1.2.0/24	10.1.5.1	R1.4
10.1.3.0/24	10.1.4.1	R1.5
10.1.3.0/24	10.1.5.1	R1.6

Browser tabs: Inbox (2.235) - shreyamt@gmail.com, ClayNet™ | Home, clayroot@ClayNet: ~, how to start apache server - Google

Address bar: 10.1.10.10:9000/dashboard/dWsYCCpFH7x77YKNG

ClayNet™ | NIVETTI SYSTEMS

Router1

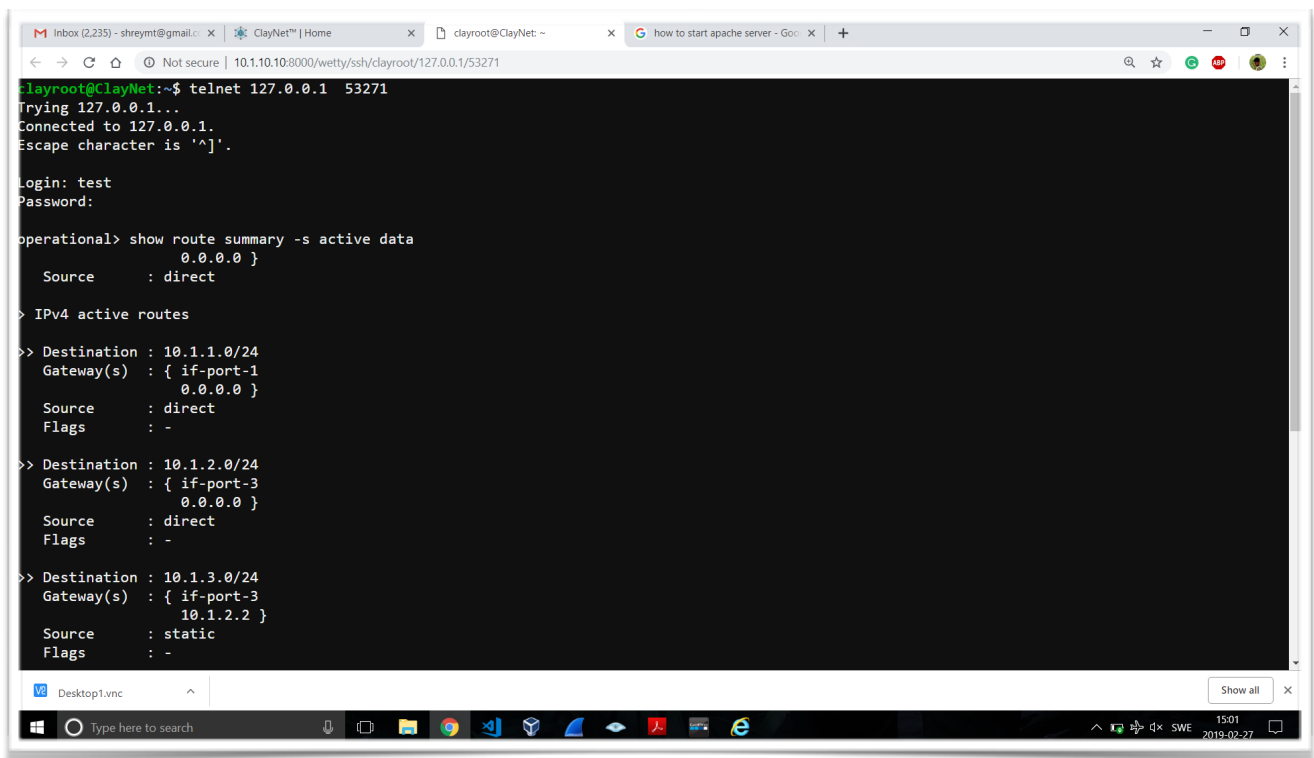
Name: Router1

Port	IP Address	Netmask	Port	IP Address	Netmask
1	10.1.4.1	/ 24	5	0.0.0.0	/ 0
2	10.1.3.2	/ 24	6	0.0.0.0	/ 0
3	10.1.2.2	/ 24	7	0.0.0.0	/ 0
4	0.0.0.0	/ 0	8	0.0.0.0	/ 0

Close

Entering Static routing table entries

1. Create Topology as mentioned below and create necessary connections.
2. Set the server IP and client IP using the remote desktop viewer.
3. Configure the routers using the following commands:
 - a. Login > username:test and password:test@12345
 - b. Operational > configure
 - c. Configure> create parameter-group ip-route r1.1
 - d. Configure> set enable yes
 - e. Configure> set router data
 - f. Configure> set destination [10.1.1.2/24](#)
 - g. Configure> set next-hop gateway 10.1.5.1
 - h. Save
 - i. Exit
4. Build a routing table for each router by creating routes for R1 via R2 as well as R3 etc. by using the above commands and changing router destination and next hop.
5. After configuring the router, client and server, proceed to ping the server from the client. If the topology is configured correctly, ping is successful.
6. Observe TTL values.



```
clayroot@ClayNet:~$ telnet 127.0.0.1 53271
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^J'.

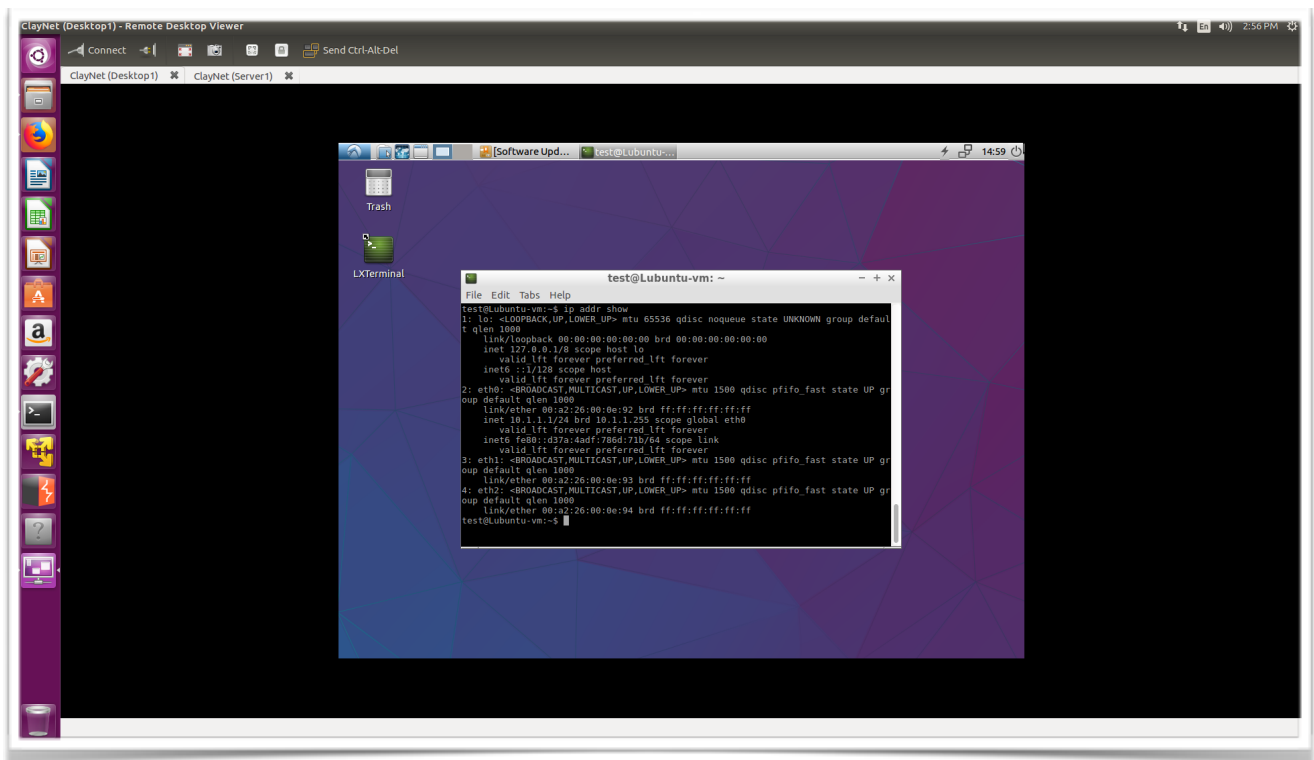
login: test
Password:

operational> show route summary -s active data
      0.0.0.0 }
Source      : direct

> IPv4 active routes
>> Destination : 10.1.1.0/24
Gateway(s)    : { if-port-1
                  0.0.0.0 }
Source        : direct
Flags         : -

>> Destination : 10.1.2.0/24
Gateway(s)    : { if-port-3
                  0.0.0.0 }
Source        : direct
Flags         : -

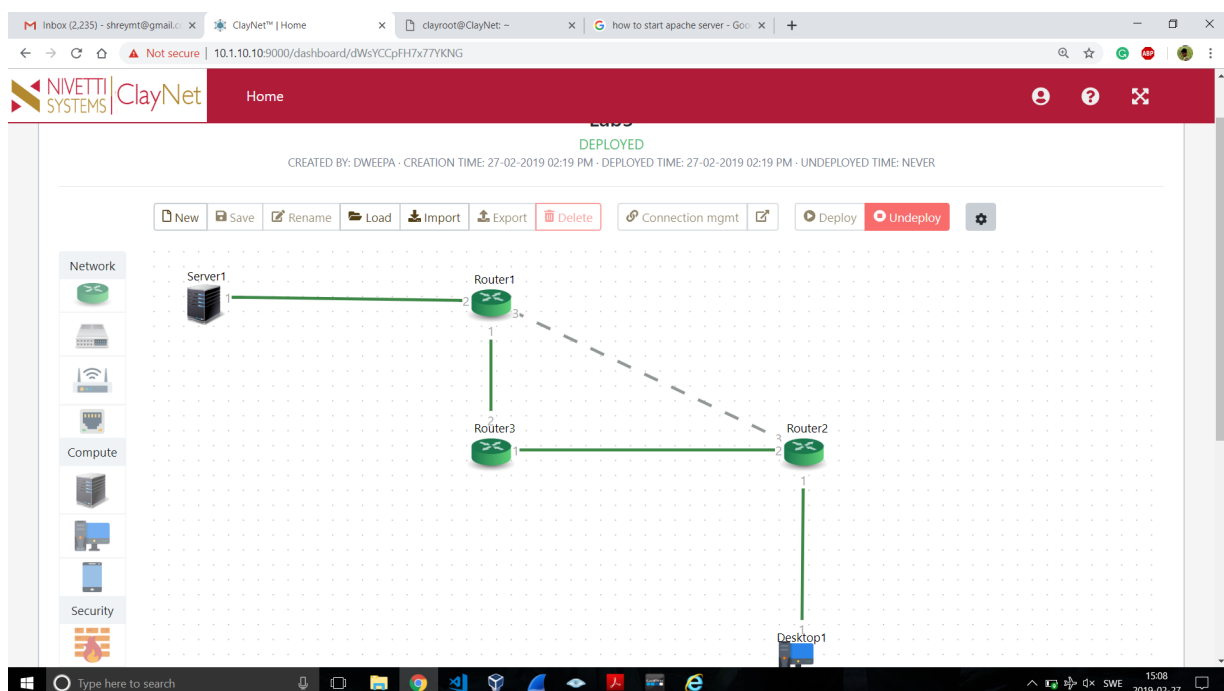
>> Destination : 10.1.3.0/24
Gateway(s)    : { if-port-3
                  10.1.2.2 }
Source        : static
Flags         : -
```



Assignment

- Observe TTL values after cutting a link

1. The below topology is observed after cutting the link between R2 and R3.
2. Ping the server from the client.
3. Observe TTL values.
4. TTL value is one less since destination is reached by going through all three routers rather than shortcut.



- **Analyse network parameters using iperf**

1. Start iperf on server using `$iperf -s`
2. Start iperf on client using `$iperf -c 10.1.1.1`
3. Observe bandwidth values