

CENG435 Term Project Part 1

Ahmet Kürşad Şaşmaz

Ali Şen

I. INTRODUCTION

In this CENG435 Term Project, participants worked on an already designed network topology. Shortest path from source to destination was found by calculating the average RTT values between every node. After, topology was reset according to shortest path. Finally, end to end delay was tested with respect to network emulation delay.

II. SET UP THE ENVIRONMENT

A. Network Topology

There was 5 nodes on the topology. Names of the nodes were **s,r1,r2,r3,d**. Connection link table is given below.

s	r1
s	r2
s	r3
d	r1
d	r2
d	r3
r1	r2
r3	r2

B. Initial Configure Scripts

Initial configure scripts were for node **r1** and node **r2**. Before the server up, configure scripts were run on related nodes.

C. Server/Client Files

Every node had capability to behave server and client at the same time. In order to provide this, threading paradigm was used. Every node had different purposes.

1) s:

- Get data from **r1** and send back.
- Get data from **r2** and send back.
- Get data from **r3** and send back.

2) r1:

- Get data from **r2** and send back to **r2** and send copies of data to **s** and **d**, and keep a log of data and send time.
- Get data from **s** and keep a log of data and receive time.
- Get data from **d** and keep a log of data and receive time.

3) r2:

- Create data packets and send it to **s**, **d**, **r1**, **r3**, and keep a log of data and send time.
- Get data from **s** and keep a log of data and receive time.
- Get data from **d** and keep a log of data and receive time.
- Get data from **r1** and keep a log of data and receive time.
- Get data from **r3** and keep a log of data and receive time.

4) r3:

- Get data from **r2** and send back to **r2** and send copies of data to **s** and **d**, and keep a log of data and send time.
- Get data from **s** and keep a log of data and receive time.
- Get data from **d** and keep a log of data and receive time.

5) d:

- Get data from **r1** and send back.
- Get data from **r2** and send back.
- Get data from **r3** and send back.

D. Synchronization of Server Times

In order to get the time packet send and packet receive time correctly, the servers had to be synchronized with respect to time. To provide this;

sudo ntpdate -qu north-america.pool.ntp.org
command was executed in every node.

III. DISCOVERING THE NETWORK TOPOLOGY

A. Starting The Servers

Because **s** and **d** were mirroring nodes, these were initialized first.

Then **r1** and **r3** were initialized.

Because **r2** was data initializer node, **r2** was initialized last. Totally, 1000 packets were used for every link.

B. Collecting The RTT Values

Logs were brought together to be compared. About one third of packets were lost. Matched data packets were sorted out, and average of the difference between receive time and send time was calculated for every link. The table of average RTT value of links is below.

s	r1	100.3ms
s	r2	104.6ms
s	r3	0.5ms
d	r1	85.4ms
d	r2	102.8ms
d	r3	0.5ms
r1	r2	258.0ms
r3	r2	224.5ms

C. Finding The Shortest Path

- nodeSet = {}
- s = 0
- d = inf
- r1 = inf
- r2 = inf
- r3 = inf
- nodeSet = {s}

- $r1 = 100.3$
- $r2 = 104.6$
- $r3 = 0.5$
- $\text{nodeSet} = \{s, r3\}$
- $d = 0.5$
- $\text{nodeSet} = \{s, r3, d\}$
- $\text{nodeSet} = \{s, r3, d, r1\}$
- $\text{nodeSet} = \{s, r3, d, r1, r2\}$

Shortest path table is below.

s	r1
s	r2
s	r3
r3	d

IV. RESET THE ENVIRONMENT

A. Network Topology

There was 5 nodes on the topology. Names of the nodes were **s, r1, r2, r3, d**. Connection link table is given below.

s	r1
s	r2
s	r3
r3	d

B. Configure Scripts

Because there were 3 different experiments, 3 different configuration files for every link between the nodes **s, r3** and **d**. To configure the links, two different configuration scripts were prepared for **s** and **r3**. Before the server up for specific experiment, configure scripts for related experiment were run on related nodes.

Experiment	Network Emulation Delay	Jitter
1	20ms	5ms
2	40ms	5ms
3	50ms	5ms

C. Server/Client Files

Because the shortest path was **s-r3-d**. Purposes of the nodes were changed.

1) **s**:

- Create data packets and send it to **r3**, and keep a log of data and send time.

2) **r3**:

- Get data from **s** and send it to **d**.

3) **d**:

- Get data from **r3**, and keep a log of data and receive time.

V. EXPERIMENTS

A. Starting The Servers

For every specific experiment, below were done. Because **d** was listening node, it was initialized first.

Then **r3** was initialized.

Because **s** was data initializer node, **s** was initialized last.

Totally, 3000 packets were sent, and send time and receive time were recorded by **s** and **d**

Finally, the scripts were terminated.

B. Collecting The End to End Delay Values

Logs were brought together to be compared. About one sixth of packets were lost. Matched data packets were sorted out, differences of the receive time and the send time were calculated for every experiment, and plotted on a graph with % 95 confidence interval by using Excel.

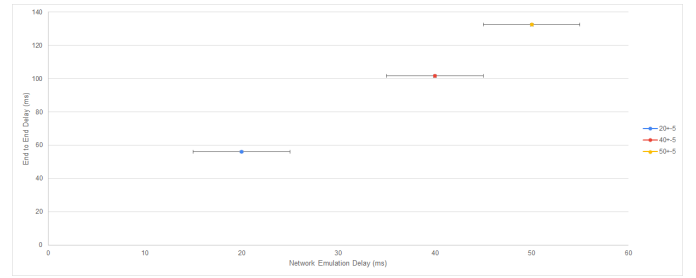


Fig. 1. Network Emulation Delay vs End to End Delay

Figure 1 Network Emulation Delay vs End to End Delay.

Network Emulation Delay	μ	σ	Error
20ms \pm 5ms	56ms	19ms	1ms
40ms \pm 5ms	102ms	21ms	1ms
50ms \pm 5ms	132ms	19ms	1ms

According to the graph, as network emulation delay between the links increases, end to end delay from source to destination increases almost linearly. When d stands for network emulation delay and p stands for total processing time of nodes, theoretical end to end delay formula becomes $2d \pm 5ms + p$. According to this formula, when network emulation delay increases, end to end delay increases twice. It can be seen that the results of the experiments were consistent with the theoretical calculations.

VI. DISCUSSIONS

In real life, most of the links are used by many users most of the time packets affected by the delay but in our topology nobody our link except us because we create the servers and clients and they are under control of us. That is why there would be no delay or the same delays for other router unless we configure the routers. This means that all packets would give the same amount of delay. It doesn't matter which router our packets pass through, to prevent this we configure them. The advantage of packet switching is sending and receiving data at the same time when the link is being used. But on the other hand, we observe some packet loss which we didn't want in real life and that is the disadvantage of such a topology.