# Network Term Project Part 2

Name: Yusuf Topçuoğlu Number: 2099398
Name: İbrahim Atacan Kerpiç Number: 2099174

## How to Run

Submission has two parts, codes and report.
Code part consist of 3 python files: “source.py”, “destination.py”, “broker.py”.

First, files need to be uploaded to the machines in the given topology.

Upload “destination.py” to node “d”
Upload “source.py” to node “s”
Upload “input.txt” to node “s”
Upload “broker.py” to node “b”

Execution order:

Firstly, broker.py must be executed in b node:

```
> python3 broker.py
```

Then, destination must be executed:

```
> python3 destination.py
```

Lastly, source must be executed with a parameter as a file name:

```
> python3 source.py input.txt
```

When source is executed, it sends the file to the broker end the source script is finished. Then the broker sends the file to the destination over r1 and r2 using our reliable data transfer protocol on top of UDP.

# Routing Tables and Commands

## 1. Broker

### Output of “ip r” command in Broker:

```
default via 172.16.0.1 dev eth0
10.0.0.0/8 via 10.10.4.2 dev eth2
10.10.1.0/24 dev eth3  proto kernel  scope link  src 10.10.1.2
10.10.2.0/24 dev eth1  proto kernel  scope link  src 10.10.2.1
10.10.3.0/31 via 10.10.2.2 dev eth1
10.10.4.0/24 dev eth2  proto kernel  scope link  src 10.10.4.1
172.16.0.0/12 dev eth0  proto kernel  scope link  src 172.17.2.6
```

### Output of “route” command in Broker:

| Destination | Gateway | Genmask | Flags | Metric | Ref | Use Iface |
|---|---|---|---|---|---|---|
| default | 172.16.0.1 | 0.0.0.0 | UG | 0 | 0 | eth0 |
| 10.0.0.0 | r2-link-3 | 255.0.0.0 | UG | 0 | 0 | eth2 |
| 10.10.1.0 | * | 255.255.255.0 | U | 0 | 0 | eth3 |
| 10.10.2.0 | * | 255.255.255.0 | U | 0 | 0 | eth1 |
| 10.10.3.0 | r1-link-1 | 255.255.255.254 | UG | 0 | 0 | eth1 |
| 10.10.4.0 | * | 255.255.255.0 | U | 0 | 0 | eth2 |
| 172.16.0.0 | * | 255.240.0.0 | U | 0 | 0 | eth0 |

*Table 1:Kernel IP routing table in Broker*

### Routing commands applied to Broker:

- sudo ip r add 10.10.3.0/24 via 10.10.2.2 dev eth1
- sudo ip r add 10.10.5.0/24 via 10.10.4.2 dev eth2
- sudo ip r delete 10.10.3.0/31 via 10.10.2.2

### Output of “ip r” command after the configurations in Broker:

```
default via 172.16.0.1 dev eth0
10.0.0.0/8 via 10.10.4.2 dev eth2
10.10.1.0/24 dev eth3  proto kernel  scope link  src 10.10.1.2
10.10.2.0/24 dev eth1  proto kernel  scope link  src 10.10.2.1
10.10.3.0/24 via 10.10.2.2 dev eth1
10.10.4.0/24 dev eth2  proto kernel  scope link  src 10.10.4.1
10.10.5.0/24 via 10.10.4.2 dev eth2
172.16.0.0/12 dev eth0  proto kernel  scope link  src 172.17.2.6
```

**Output of “route” command after the configurations in Broker:**

| Destination | Gateway | Genmask | Flags | Metric | Ref | Use Iface |
|---|---|---|---|---|---|---|
| default | 172.16.0.1 | 0.0.0.0 | UG | 0 | 0 | eth0 |
| 10.0.0.0 | r2-link-3 | 255.0.0.0 | UG | 0 | 0 | eth2 |
| 10.10.1.0 | * | 255.255.255.0 | U | 0 | 0 | eth3 |
| 10.10.2.0 | * | 255.255.255.0 | U | 0 | 0 | eth1 |
| 10.10.3.0 | r1-link-1 | 255.255.255.0 | UG | 0 | 0 | eth1 |
| 10.10.4.0 | * | 255.255.255.0 | U | 0 | 0 | eth2 |
| 10.10.5.0 | r2-link-3 | 255.255.255.0 | UG | 0 | 0 | eth2 |
| 172.16.0.0 | * | 255.240.0.0 | U | 0 | 0 | eth0 |

*Table 2:Kernel IP routing table after configurations in Broker*

**Traceroute from broker to 10.10.3.2:**

traceroute to 10.10.3.2 (10.10.3.2), 30 hops max, 60 byte packets
1  r1-link-1 (10.10.2.2)  0.593 ms  0.552 ms  0.519 ms
2  d-link-2 (10.10.3.2)  1.211 ms  1.184 ms  1.151 ms

**Traceroute from broker to 10.10.5.2:**

traceroute to 10.10.5.2 (10.10.5.2), 30 hops max, 60 byte packets
1  r2-link-3 (10.10.4.2)  0.588 ms  0.542 ms  0.508 ms
2  d-link-4 (10.10.5.2)  1.270 ms  1.226 ms  1.194 ms

## 2) Destination

**Output of “ip r” command in Destination:**

```
default via 172.16.0.1 dev eth0
10.0.0.0/8 via 10.10.3.1 dev eth1
10.10.3.0/24 dev eth1  proto kernel  scope link  src 10.10.3.2
10.10.4.0/31 via 10.10.3.1 dev eth1
10.10.4.0/22 via 10.10.5.1 dev eth2
10.10.5.0/24 dev eth2  proto kernel  scope link  src 10.10.5.2
172.16.0.0/12 dev eth0  proto kernel  scope link  src 172.17.2.7
```

### Output of “route” command in Destination:

| Destination | Gateway | Genmask | Flags | Metric | Ref | Use Iface |
|---|---|---|---|---|---|---|
| default | 172.16.0.1 | 0.0.0.0 | UG | 0 | 0 | eth0 |
| 10.0.0.0 | r1-link-2 | 255.0.0.0 | UG | 0 | 0 | eth1 |
| 10.10.3.0 | * | 255.255.255.0 | U | 0 | 0 | eth1 |
| 10.10.4.0 | r1-link-2 | 255.255.255.254 | UG | 0 | 0 | eth1 |
| 10.10.4.0 | r2-link-4 | 255.255.252.0 | UG | 0 | 0 | eth2 |
| 10.10.5.0 | * | 255.255.255.0 | U | 0 | 0 | eth2 |
| 172.16.0.0 | * | 255.240.0.0 | U | 0 | 0 | eth0 |

*Table 3: Kernel IP routing table in Destination*

### Routing commands applied to Destination:

- sudo ip r add 10.10.2.0/24 via 10.10.3.1 dev eth1
- sudo ip r add 10.10.4.0/24 via 10.10.5.1 dev eth2
- sudo ip r delete 10.10.4.0/31 via 10.10.3.1

### Output of “ip r” command after the configurations in Destination:

```
default via 172.16.0.1 dev eth0
10.0.0.0/8 via 10.10.3.1 dev eth1
10.10.2.0/24 via 10.10.3.1 dev eth1
10.10.3.0/24 dev eth1  proto kernel  scope link  src 10.10.3.2
10.10.4.0/24 via 10.10.5.1 dev eth2
10.10.4.0/22 via 10.10.5.1 dev eth2
10.10.5.0/24 dev eth2  proto kernel  scope link  src 10.10.5.2
172.16.0.0/12 dev eth0  proto kernel  scope link  src 172.17.2.7
```

### Output of “route” command after the configurations in Destination:

| Destination | Gateway | Genmask | Flags | Metric | Ref | Use Iface |
|---|---|---|---|---|---|---|
| default | 172.16.0.1 | 0.0.0.0 | UG | 0 | 0 | eth0 |
| 10.0.0.0 | r1-link-2 | 255.0.0.0 | UG | 0 | 0 | eth1 |
| 10.10.2.0 | r1-link-2 | 255.255.255.0 | UG | 0 | 0 | eth1 |
| 10.10.3.0 | * | 255.255.255.0 | U | 0 | 0 | eth1 |
| 10.10.4.0 | r2-link-4 | 255.255.255.0 | UG | 0 | 0 | eth2 |
| 10.10.4.0 | r2-link-4 | 255.255.252.0 | UG | 0 | 0 | eth2 |
| 10.10.5.0 | * | 255.255.255.0 | U | 0 | 0 | eth2 |
| 172.16.0.0 | * | 255.240.0.0 | U | 0 | 0 | eth0 |

*Table 4: Kernel IP routing table after configurations in Destination*

**Traceroute from Destination to 10.10.2.1 :**

traceroute to 10.10.2.1 (10.10.2.1), 30 hops max, 60 byte packets
1  r1-link-2 (10.10.3.1)  0.614 ms  0.567 ms  0.532 ms
2  B-link-1 (10.10.2.1)  1.084 ms  1.053 ms  1.021 ms

**Traceroute from Destination to 10.10.4.1 :**

traceroute to 10.10.4.1 (10.10.4.1), 30 hops max, 60 byte packets
1  r2-link-4 (10.10.5.1)  0.600 ms  0.556 ms  0.526 ms
2  B-link-3 (10.10.4.1)  1.168 ms  1.139 ms  1.108 ms

# Experiments

## Experiment 1

Firstly, we applied configurations to all links between broker and destination in order to configure loss, corrupt, duplicate and delay values. To do this, we execute following commands in broker, r1, r2 and destination virtual machines.

```
sudo tc qdisc add dev eth1 root netem loss 0.5% corrupt 0% duplicate 0% delay 3ms reorder 0% 0%
sudo tc qdisc add dev eth2 root netem loss 0.5% corrupt 0% duplicate 0% delay 3ms reorder 0% 0%
```

Then, we run the destination.py, broker.py and source.py respectively and obtained the first file transfer time after whole packet is sent.

Secondly, we changed configurations by following commands and obtained the second file transfer time:

```
sudo tc qdisc change dev eth1 root netem loss 10% corrupt 0% duplicate 0% delay 3ms reorder 0% 0%
sudo tc qdisc change dev eth2 root netem loss 10% corrupt 0% duplicate 0% delay 3ms reorder 0% 0%
```

Lastly, we changed configurations by following commands and obtained the final file transfer time:

```
sudo tc qdisc change dev eth1 root netem loss 20% corrupt 0% duplicate 0% delay 3ms reorder 0% 0%
sudo tc qdisc change dev eth2 root netem loss 20% corrupt 0% duplicate 0% delay 3ms reorder 0% 0%
```

## Experiment 2

Similarly, we applied following configurations respectively and obtained the results for each of them.

```
sudo tc qdisc change dev eth1 root netem loss 0% corrupt 0.2% duplicate 0% delay 3ms reorder 0% 0%
sudo tc qdisc change dev eth2 root netem loss 0% corrupt 0.2% duplicate 0% delay 3ms reorder 0% 0%
```

```
sudo tc qdisc change dev eth1 root netem loss 0% corrupt 10% duplicate 0% delay 3ms reorder 0% 0%
sudo tc qdisc change dev eth2 root netem loss 0% corrupt 10% duplicate 0% delay 3ms reorder 0% 0%
```

```
sudo tc qdisc change dev eth1 root netem loss 0% corrupt 20% duplicate 0% delay 3ms reorder 0% 0%
sudo tc qdisc change dev eth2 root netem loss 0% corrupt 20% duplicate 0% delay 3ms reorder 0% 0%
```

## Experiment 3

Similarly, we applied following configurations respectively and obtained the results for each of them.

```
sudo tc qdisc change dev eth1 root netem loss 0% corrupt 0% duplicate 0% delay 3ms reorder 1% 50%
sudo tc qdisc change dev eth2 root netem loss 0% corrupt 0% duplicate 0% delay 3ms reorder 1% 50%
```

```
sudo tc qdisc change dev eth1 root netem loss 0% corrupt 0% duplicate 0% delay 3ms reorder 10% 50%
sudo tc qdisc change dev eth2 root netem loss 0% corrupt 0% duplicate 0% delay 3ms reorder 10% 50%
```

```
sudo tc qdisc change dev eth1 root netem loss 0% corrupt 0% duplicate 0% delay 3ms reorder 35% 50%
sudo tc qdisc change dev eth2 root netem loss 0% corrupt 0% duplicate 0% delay 3ms reorder 35% 50%
```