

Objective:

To configure and verify static routing on Cisco routers using Packet Tracer. The goal is to allow routers to communicate with each other and route packets between different networks using static routes.

Topology:

- **Router1 (R1):** Connects to Network 192.168.1.0/24
- **Router2 (R2):** Connects to Network 192.168.2.0/24
- **Router3 (R3):** Connects to Network 192.168.3.0/24
- All routers are connected using serial or Ethernet links.

Equipment:

- 3 Cisco Routers
- 3 PCs (1 connected to each router for testing)
- Serial or Ethernet cables
- Cisco Packet Tracer software

Step 1: Network Setup

1. **Create the Topology in Packet Tracer**
 - Use the "Router" tool to add three routers (R1, R2, R3).
 - Use the "PC" tool to add one PC to each router.
 - Use Ethernet or Serial connections to connect the routers in a triangle-like topology. Each PC is connected to its respective router using an Ethernet connection.
2. **Assign IP Addresses**
 - **R1:**
 - Interface G0/0: 192.168.1.1 /24 (connected to PC1)
 - Interface S0/0/0: 10.1.1.1 /30 (link between R1 and R2)
 - Interface S0/0/1: 10.1.1.5 /30 (link between R1 and R3)
 - **R2:**
 - Interface G0/0: 192.168.2.1 /24 (connected to PC2)
 - Interface S0/0/0: 10.1.1.2 /30 (link between R1 and R2)
 - **R3:**
 - Interface G0/0: 192.168.3.1 /24 (connected to PC3)
 - Interface S0/0/0: 10.1.1.6 /30 (link between R1 and R3)

Step 2: Configuration on Routers

1. Configuring Router1 (R1)

- Access **R1** via the CLI in Packet Tracer.
- Enter privileged EXEC mode:

```
bash
Copy code
R1> enable
```

- Enter global configuration mode:

```
bash
Copy code
R1# configure terminal
```

- Configure the IP address on the Ethernet interface:

```
bash
Copy code
R1(config)# interface gigabitethernet 0/0
R1(config-if)# ip address 192.168.1.1 255.255.255.0
R1(config-if)# no shutdown
```

- Configure the serial interface between **R1** and **R2**:

```
bash
Copy code
R1(config)# interface serial 0/0/0
R1(config-if)# ip address 10.1.1.1 255.255.255.252
R1(config-if)# no shutdown
```

- Configure the serial interface between **R1** and **R3**:

```
bash
Copy code
R1(config)# interface serial 0/0/1
R1(config-if)# ip address 10.1.1.5 255.255.255.252
R1(config-if)# no shutdown
```

- Exit interface configuration mode:

```
bash
Copy code
R1(config-if)# exit
```

- Configure static routes on **R1**:

```
bash
Copy code
R1(config)# ip route 192.168.2.0 255.255.255.0 10.1.1.2
R1(config)# ip route 192.168.3.0 255.255.255.0 10.1.1.6
```

2. Configuring Router2 (R2)

- Access **R2** via the CLI.
- Enter privileged EXEC mode:

```
bash
Copy code
R2> enable
```

- Enter global configuration mode:

```
bash
```

```
Copy code
R2# configure terminal
```

- Configure the IP address on the Ethernet interface:

```
bash
Copy code
R2(config)# interface gigabitethernet 0/0
R2(config-if)# ip address 192.168.2.1 255.255.255.0
R2(config-if)# no shutdown
```

- Configure the serial interface between **R2** and **R1**:

```
bash
Copy code
R2(config)# interface serial 0/0/0
R2(config-if)# ip address 10.1.1.2 255.255.255.252
R2(config-if)# no shutdown
```

- Exit interface configuration mode:

```
bash
Copy code
R2(config-if)# exit
```

- Configure static routes on **R2**:

```
bash
Copy code
R2(config)# ip route 192.168.1.0 255.255.255.0 10.1.1.1
R2(config)# ip route 192.168.3.0 255.255.255.0 10.1.1.1
```

3. Configuring Router3 (R3)

- Access **R3** via the CLI.
- Enter privileged EXEC mode:

```
bash
Copy code
R3> enable
```

- Enter global configuration mode:

```
bash
Copy code
R3# configure terminal
```

- Configure the IP address on the Ethernet interface:

```
bash
Copy code
R3(config)# interface gigabitethernet 0/0
R3(config-if)# ip address 192.168.3.1 255.255.255.0
R3(config-if)# no shutdown
```

- Configure the serial interface between **R3** and **R1**:

```
bash
Copy code
R3(config)# interface serial 0/0/0
R3(config-if)# ip address 10.1.1.6 255.255.255.252
R3(config-if)# no shutdown
```

- Exit interface configuration mode:

```
bash
Copy code
R3(config-if)# exit
```

- Configure static routes on **R3**:

```
bash
Copy code
R3(config)# ip route 192.168.1.0 255.255.255.0 10.1.1.5
R3(config)# ip route 192.168.2.0 255.255.255.0 10.1.1.5
```

Step 3: Verifying Static Routes

After configuring static routing on all routers, use the following commands to verify routing and connectivity:

1. Verify IP Address Configuration

```
bash
Copy code
R1# show ip interface brief
R2# show ip interface brief
R3# show ip interface brief
```

2. Verify Routing Table

Use this command to check if the static routes are correctly installed in the routing table:

```
bash
Copy code
R1# show ip route
R2# show ip route
R3# show ip route
```

3. Testing Connectivity

Use the `ping` command to check the connectivity between PCs on different networks:

```
bash
Copy code
R1# ping 192.168.2.1
R1# ping 192.168.3.1
```

Step 4: Saving Configuration

Once everything is verified, save the configurations on all routers to avoid losing them after a reboot.

```
bash
Copy code
R1# copy running-config startup-config
R2# copy running-config startup-config
R3# copy running-config startup-config
```

Conclusion:

In this lab, you successfully configured static routing on three Cisco routers using Packet Tracer. You learned how to manually configure IP addresses, establish static routes, and verify network connectivity between different routers and networks. Static routing is useful in small networks where routing paths do not change frequently.