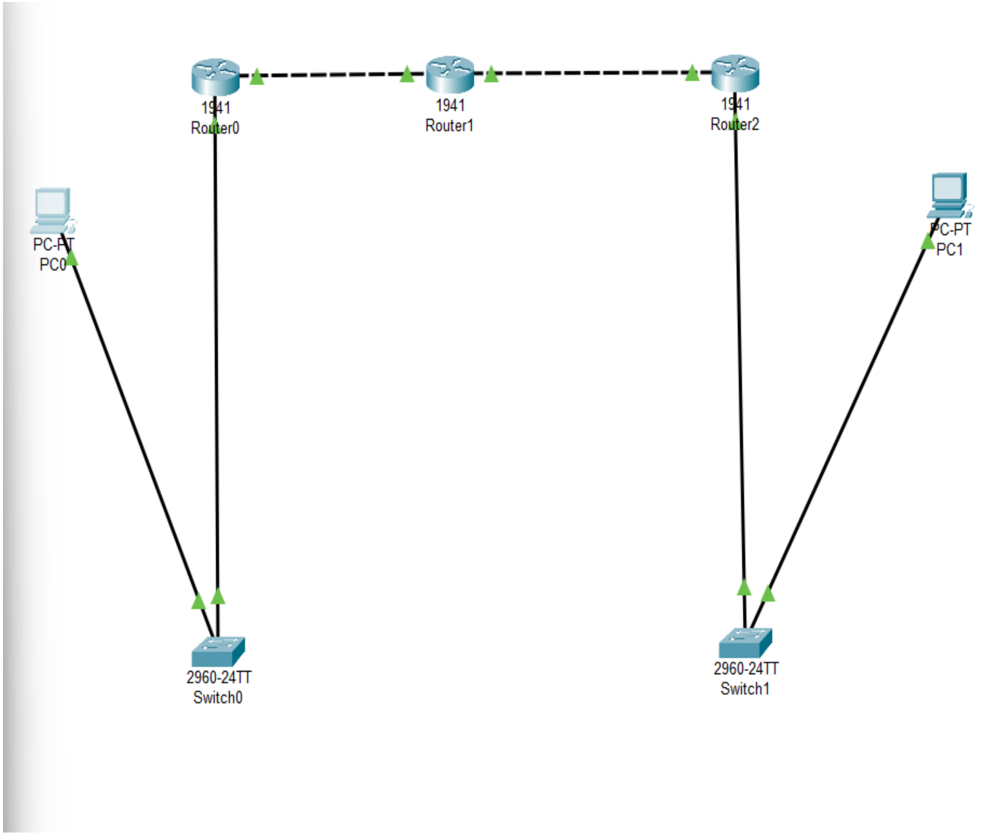
Project Report: Network Troubleshooting Lab

1. Introduction

This project involved designing, configuring, and troubleshooting a multi-router network using Cisco Packet Tracer. The objective was to implement and analyze both static and dynamic routing protocols (OSPF), simulate network failures, and apply a systematic troubleshooting methodology to restore connectivity. The project demonstrates a foundational understanding of network operations, routing logic, and fault analysis.

2. Network Topology

The network was built using three routers, two switches, and two PCs to simulate two distinct local area networks (LANs) connected via a wide area network (WAN)



3. IP Addressing and Basic Configuration

Each device was configured with a static IP address to enable communication. Routers were configured via the Command Line Interface (CLI) to assign IP addresses to their interfaces.

Device	Interface	IP Address	Subnet Mask	Default Gateway
PC0	FastEthernet0	192.168.1.10	255.255.255.0	192.168.1.1
PC1	FastEthernet0	192.168.2.10	255.255.255.0	192.168.2.1
R0	GigE0/0	192.168.1.1	255.255.255.0	N/A
R0	GigE0/1	10.0.0.1	255.255.255.252	N/A
R1	GigE0/1	10.0.0.2	255.255.255.252	N/A

R1	GigE0/0	10.0.0.5	255.255.255.252	N/A
R2	GigE0/0	10.0.0.6	255.255.255.252	N/A
R2	GigE0/1	192.168.2.1	255.255.255.0	N/A

Routing Protocol Configuration

4.1. Static Routing Initially, static routes were configured to establish a baseline for connectivity. This involved manually defining the path on each router to reach non-directly connected networks.

```
C:\>ping 192.168.2.10

Pinging 192.168.2.10 with 32 bytes of data:

Reply from 192.168.2.10: bytes=32 time<1ms TTL=125
Reply from 192.168.2.10: bytes=32 time<1ms TTL=125
Reply from 192.168.2.10: bytes=32 time=11ms TTL=125
Reply from 192.168.2.10: bytes=32 time<1ms TTL=125

Ping statistics for 192.168.2.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 11ms, Average = 2ms
```

4.2. Dynamic Routing (OSPF) To enable the network to adapt to changes automatically, the static routes were removed and the OSPF dynamic routing protocol was configured. All routers were placed in area 0.

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.0.0.0/30 is directly connected, GigabitEthernet0/1
L       10.0.0.2/32 is directly connected, GigabitEthernet0/1
C       10.0.0.4/30 is directly connected, GigabitEthernet0/0
L       10.0.0.5/32 is directly connected, GigabitEthernet0/0
O    192.168.1.0/24 [110/2] via 10.0.0.1, 00:28:48, GigabitEthernet0/1
O    192.168.2.0/24 [110/2] via 10.0.0.6, 00:11:12, GigabitEthernet0/0
```

4. Network Failure Simulation and Troubleshooting

A network failure was simulated by manually shutting down the GigabitEthernet0/0 interface on R1. A systematic troubleshooting process was then initiated.

Symptom: A ping from PC0 to PC1 failed.

Investigation: - A traceroute from PC0 showed the path stopped at Router0. show ip route on R0 confirmed the route to the destination was missing. - show ip ospf neighbour on R1 showed the neighbour relationship with R2 was lost.

Root Cause: The show ip interface brief command on R1 revealed that the GigE0/0 interface was administratively down.

```
R1#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     10.0.0.5        YES manual up                    up
GigabitEthernet0/1     10.0.0.2        YES manual up                    up
Vlan1                  unassigned      YES unset  administratively down down
R1#
```

Resolution: The interface was re-enabled using the no shutdown command, which restored connectivity.

5. Automation for Error Detection

To proactively monitor network health, an automation script could be developed using Python. This script could be scheduled to run every 15 minutes. It would automatically connect to all routers via SSH, run the show ip ospf neighbor command, and parse the output. If a router has fewer neighbors than expected, the script would immediately send an email alert to the network administrator. This would allow for much faster detection of link failures than waiting for a user to report an outage.

6. Conclusion

This project successfully demonstrated the end-to-end process of network configuration, from physical topology design to advanced troubleshooting. Key takeaways include the efficiency of dynamic routing protocols like OSPF over static routing and the importance of a logical, layer-by-layer troubleshooting methodology.