ENCS 304

Computer Networks

Assignment 1

Basic Network Topologies

Name	Shubh Singhal
Roll No.	2301010172
Programme	B.Tech Computer Science & Engineering
Section	C
Date	27 February 2026

Introduction

This assignment explores the fundamental network topologies used in computer networking — Star (Switch-based), Star (Hub-based), Ring-like (using multiple switches with STP), and a Failure Test on the Ring topology. All simulations were performed using Cisco Packet Tracer. Ping tests were conducted between devices to verify connectivity, and observations were recorded for each topology.

An important distinction to understand before examining each topology is how a **switch** and a **hub** handle frame forwarding. On the *very first transmission*, both devices behave identically — they flood (broadcast) the frame to all connected ports because neither device yet knows which port leads to the destination. However, from the *second transmission onwards*, a switch uses its dynamically built MAC address table to forward frames *only* to the correct destination port, eliminating unnecessary traffic. A hub, by contrast, *always* broadcasts every frame to all ports regardless of how many transmissions have occurred — it has no learning capability. This fundamental difference makes switches significantly more efficient than hubs in any network with more than two devices.

Task 1 – Star Topology (Switch)

A star topology was built in Cisco Packet Tracer using a 2960-24TT switch (Switch1) as the central device, with four PCs (PC0–PC3) connected to it. All devices were assigned IP addresses in the **192.168.10.x/24** subnet. Ping tests were performed from PC0 to each of the remaining PCs.

Network Topology Screenshots

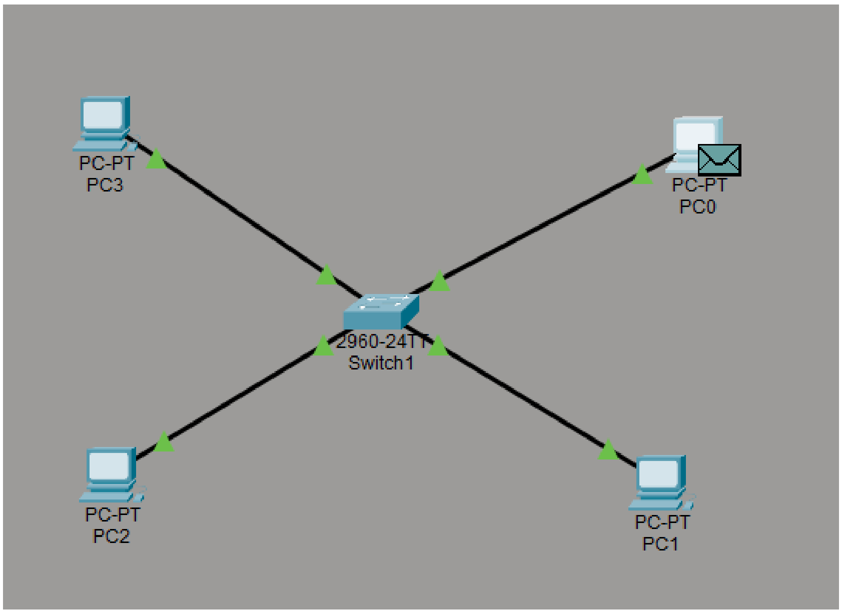

Fig 1a: PC0 initiates ping — hub broadcasts frame to all ports (first transmission)

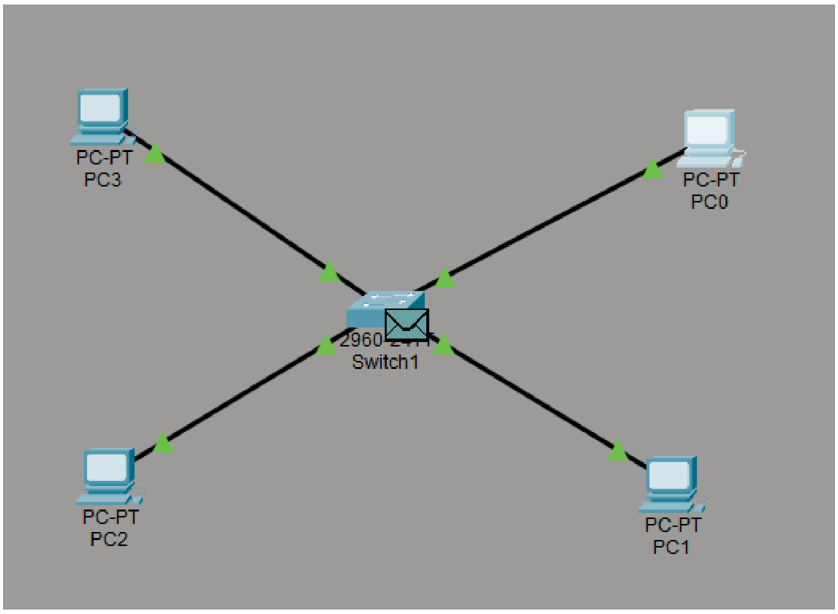

Fig 1b: Frame forwarded — switch learning MAC addresses in progress

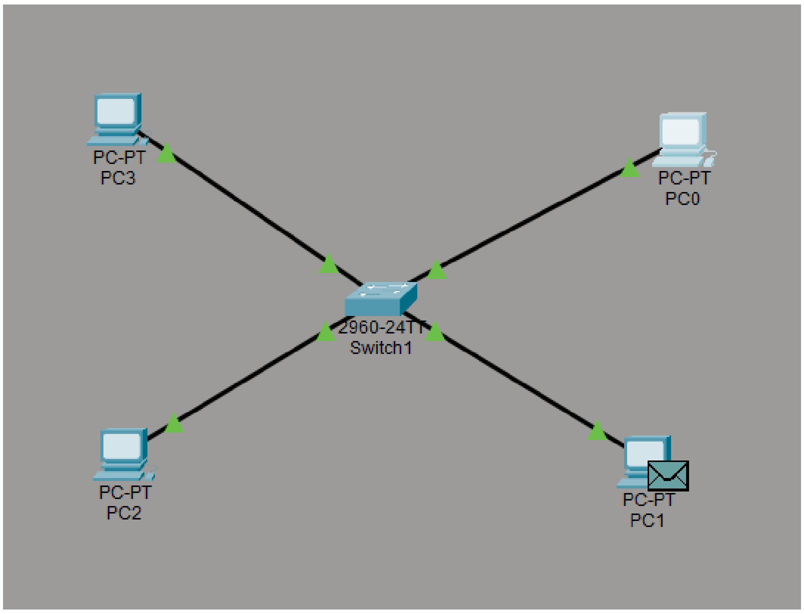

Fig 1c: Reply from destination PC — unicast delivery after MAC learning

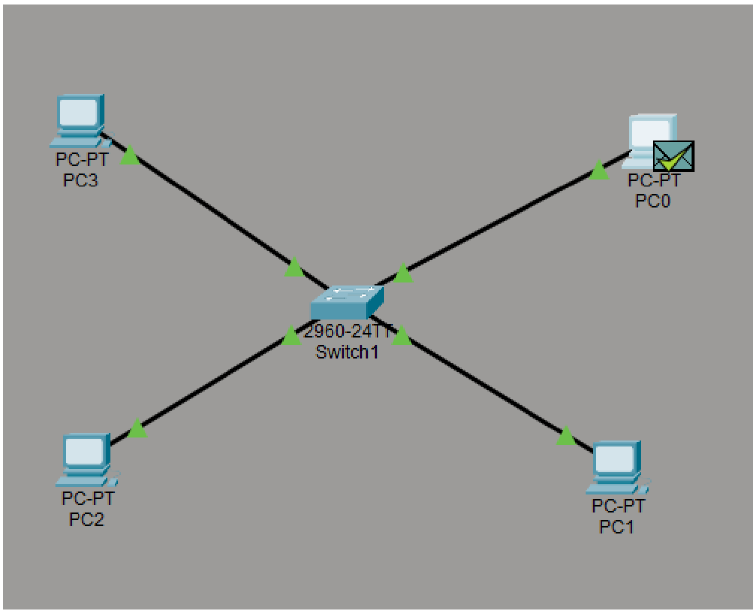

Fig 1d: Final state — switch forwards packets only to intended destination

Ping Results

```
PC0 > ping 192.168.10.2
Reply from 192.168.10.2: bytes=32 time<1ms TTL=128 (x4)
Packets: Sent=4, Received=4, Lost=0 (0% loss) | Avg=0ms

PC0 > ping 192.168.10.3
Reply from 192.168.10.3: bytes=32 time<1ms TTL=128 (x4)
Packets: Sent=4, Received=4, Lost=0 (0% loss) | Avg=0ms

PC0 > ping 192.168.10.4
Reply from 192.168.10.4: bytes=32 time<1ms TTL=128 (x4)
Packets: Sent=4, Received=4, Lost=0 (0% loss) | Avg=0ms
```

Observation

All four devices communicated successfully through the switch with zero packet loss. On the first ping to each destination, the switch flooded the frame to all ports (since the MAC table was initially empty). After learning the MAC addresses, the switch began forwarding frames exclusively to the correct destination port — a behaviour known as selective unicast forwarding. This drastically reduces unnecessary traffic on the network compared to a hub. The consistently sub-millisecond round-trip times confirm the efficiency of a switch-based star topology.

Task 2 – Star Topology (Hub)

The same physical star layout was recreated, but the central switch was replaced with a **Hub-PT (Hub0)**. Four PCs (PC0–PC3) were connected to the hub with IP addresses in the **192.168.10.x/24** range. The same ping tests were repeated to compare hub behaviour against the switch-based topology.

Network Topology Screenshots

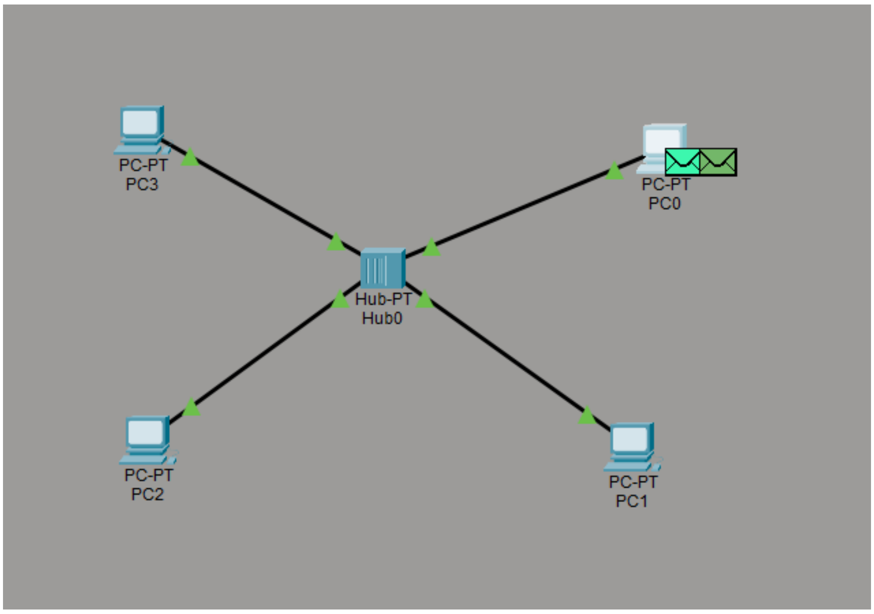

Fig 2a: PC0 sends ping — hub broadcasts frame to ALL connected devices

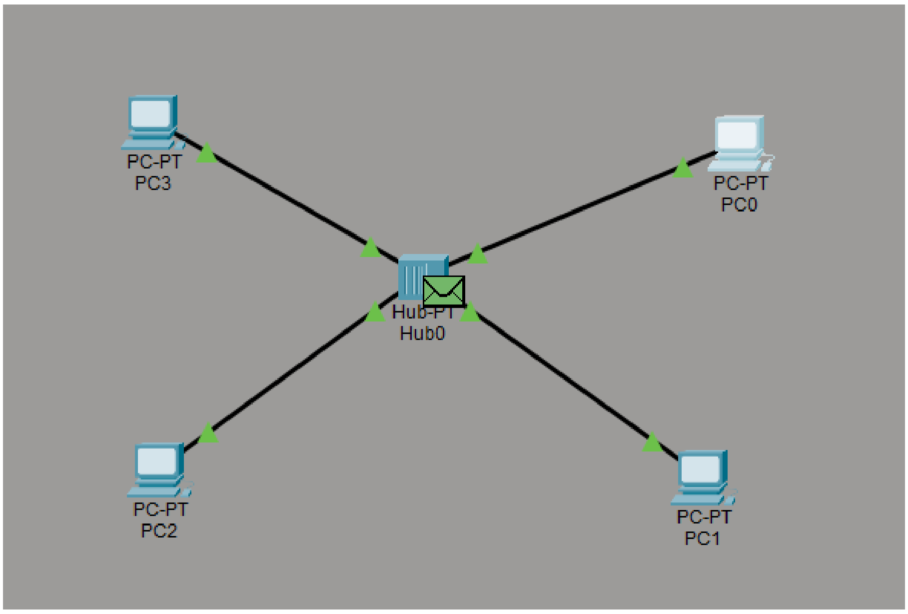

Fig 2b: Frame propagation — all devices receive the broadcast (including non-targets)

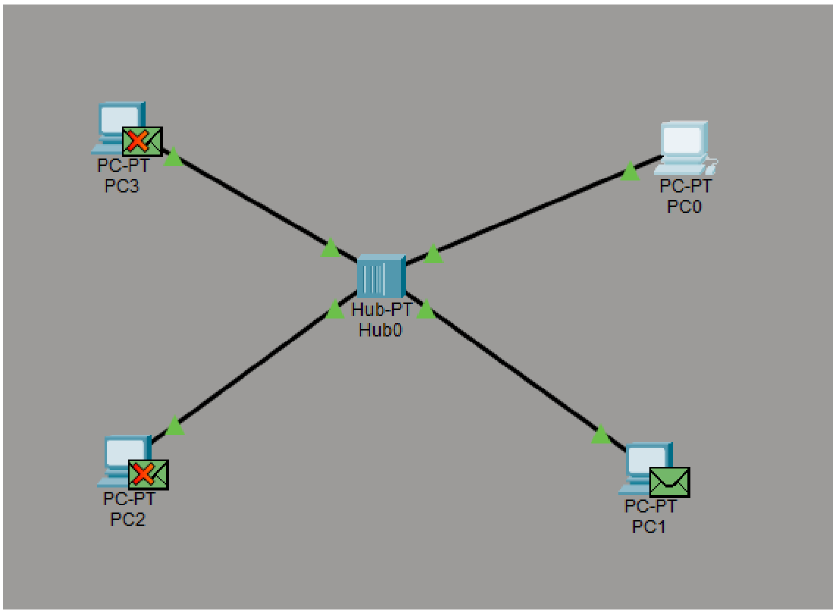

Fig 2c: Destination responds — reply also broadcast to all ports by hub

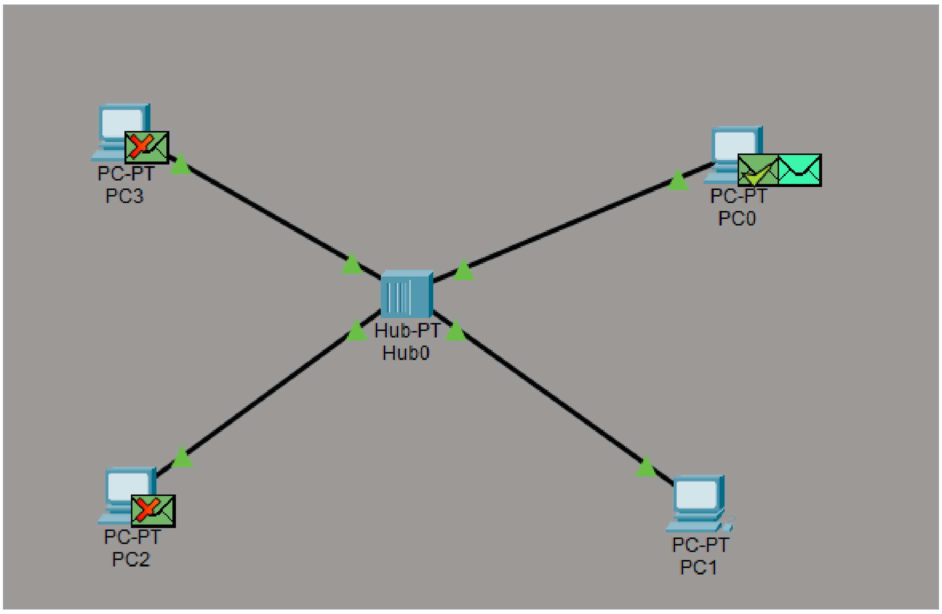

Fig 2d: Subsequent pings — hub continues to broadcast every frame (no MAC learning)

Ping Results

```
PC0(1) > ping 192.168.10.22
Reply from 192.168.10.22: bytes=32 time<1ms TTL=128 (x4)
Packets: Sent=4, Received=4, Lost=0 (0% loss) | Max=1ms, Avg=0ms

PC0(1) > ping 192.168.10.23
Reply from 192.168.10.23: bytes=32 time<1ms TTL=128 (x4)
Packets: Sent=4, Received=4, Lost=0 (0% loss) | Avg=0ms

PC0(1) > ping 192.168.10.24
Reply from 192.168.10.24: bytes=32 time<1ms TTL=128 (x4)
Packets: Sent=4, Received=4, Lost=0 (0% loss) | Max=1ms, Avg=0ms
```

Observation

All devices communicated successfully through the hub with zero packet loss. However, unlike the switch, the hub broadcasts every single frame to all connected ports — on the first transmission *and* on every subsequent transmission. It has no MAC address table and no learning capability. This means every device must process every frame, even frames not addressed to it, wasting bandwidth and CPU cycles. In a larger or busier network, this behaviour would cause significant congestion and collision domain issues, since all hub-connected devices share a single collision domain. The switch topology is clearly superior in efficiency for any real-world deployment.

Key Difference — Switch vs Hub: *Both devices broadcast frames on the first transmission when the destination MAC is unknown. After that, a switch learns and delivers frames only to the target device, while a hub permanently broadcasts every frame to all ports. This makes a switch far more scalable and efficient.*

Task 3 – Ring-like Topology (Switch Loop with STP)

Three 2960-24TT switches (Switch0, Switch1, Switch2) were interconnected in a triangular loop to form a ring-like topology. Six PCs (PC0–PC5) were distributed across the three switches. The redundant inter-switch links create loops that would cause broadcast storms without loop prevention. Cisco's **Spanning Tree Protocol (STP)** was enabled to automatically block one redundant link per loop, keeping the network loop-free while maintaining path redundancy.

Network Topology Screenshots

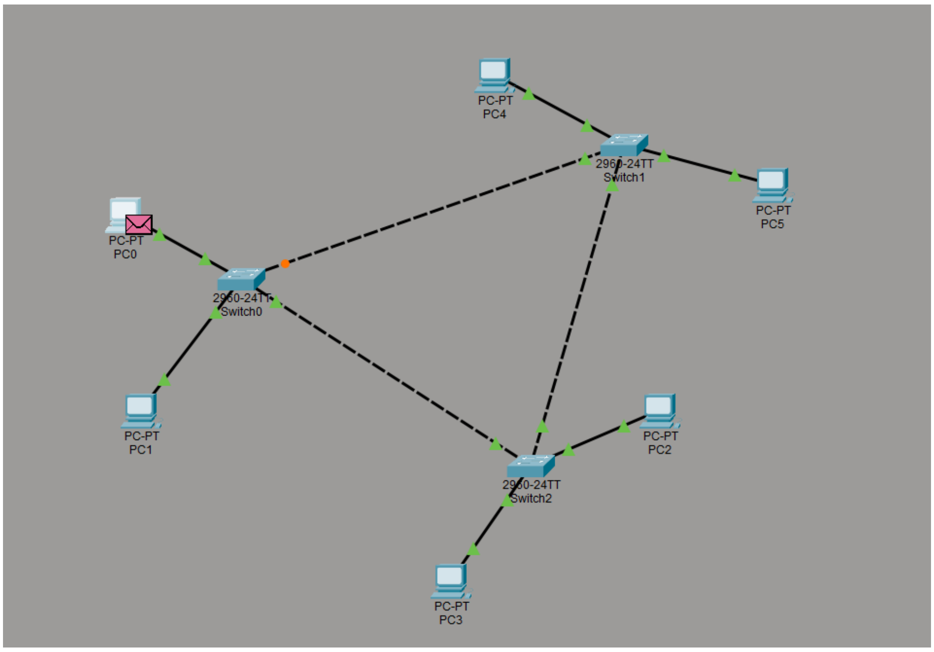

Fig 3a: PC0 initiates ping — STP has already blocked redundant links (shown as dashed)

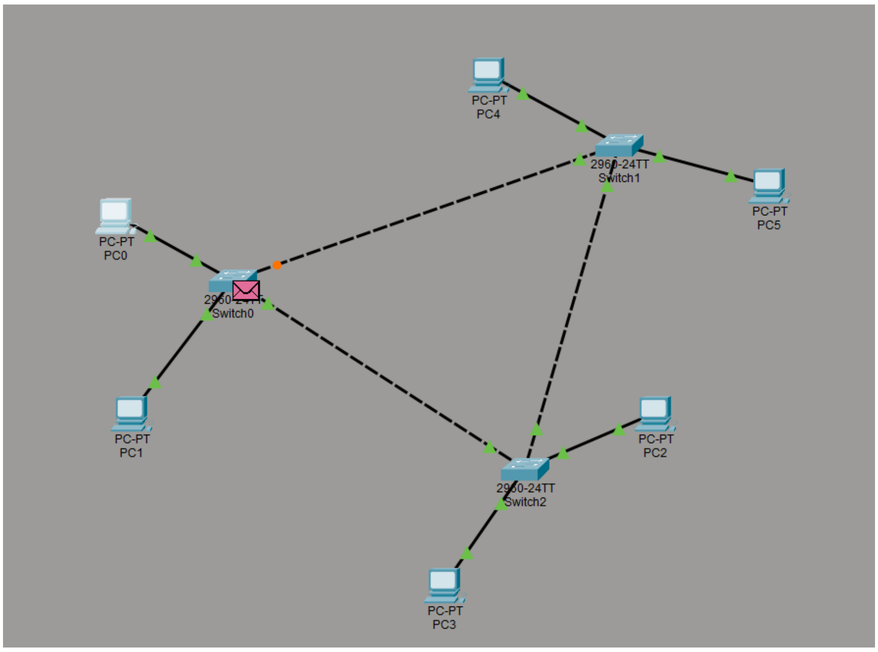

Fig 3b: Frame traverses active path through Switch0 → Switch2

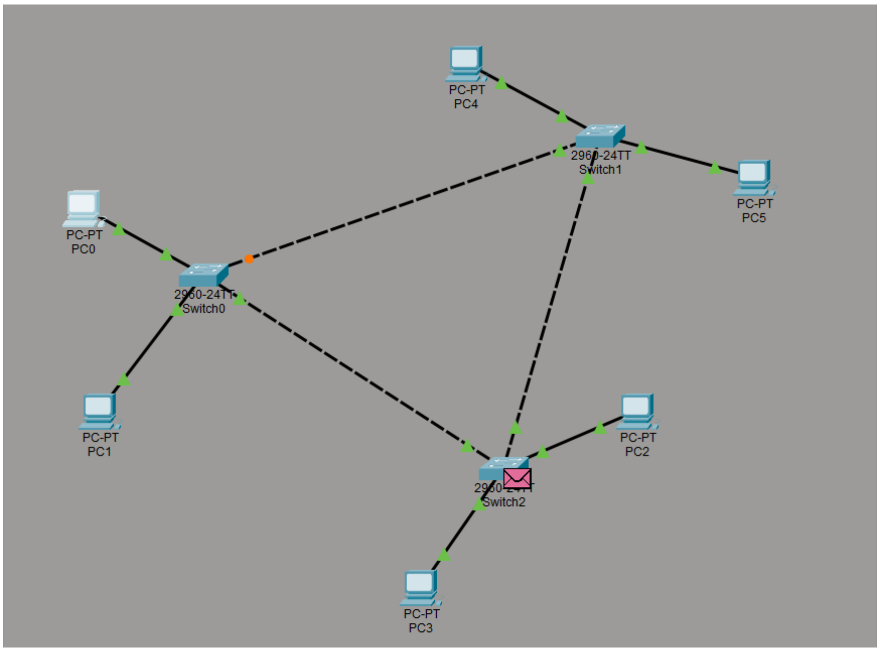

Fig 3c: Frame reaches Switch2, forwarded to destination PC

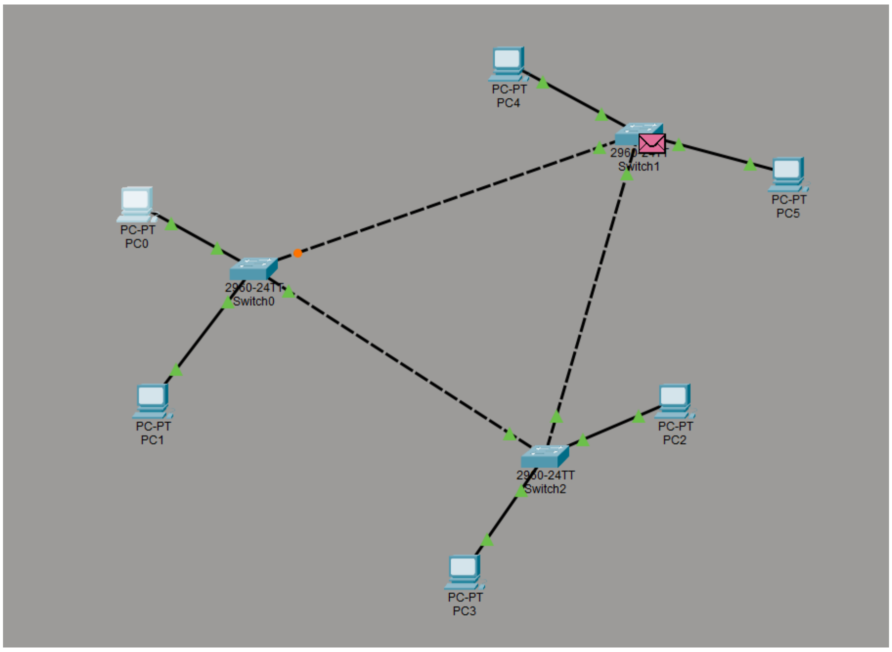

Fig 3d: Reply packet travelling back along the STP-selected active path

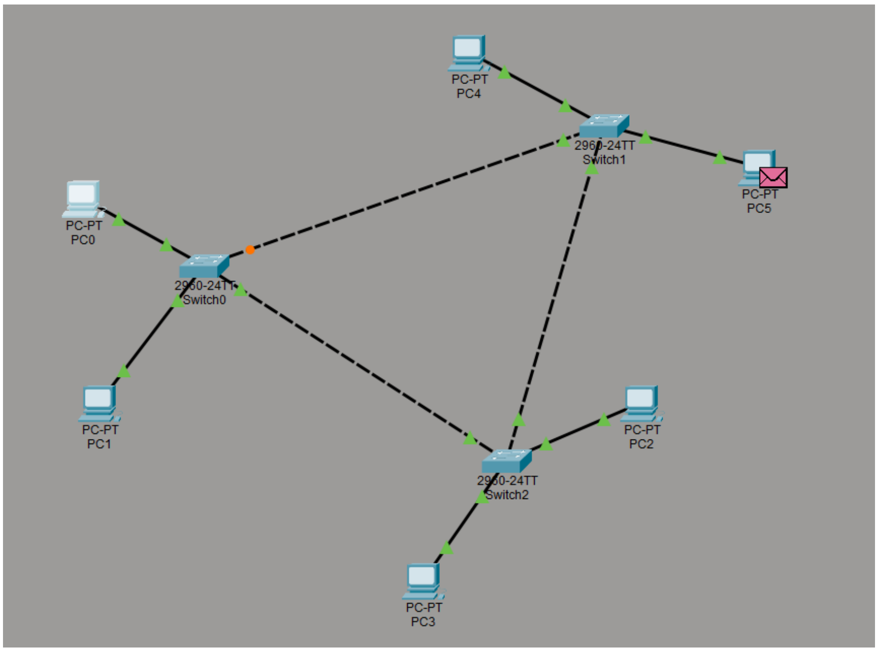

Fig 3e: Ping to second destination — traffic routed across multiple switches

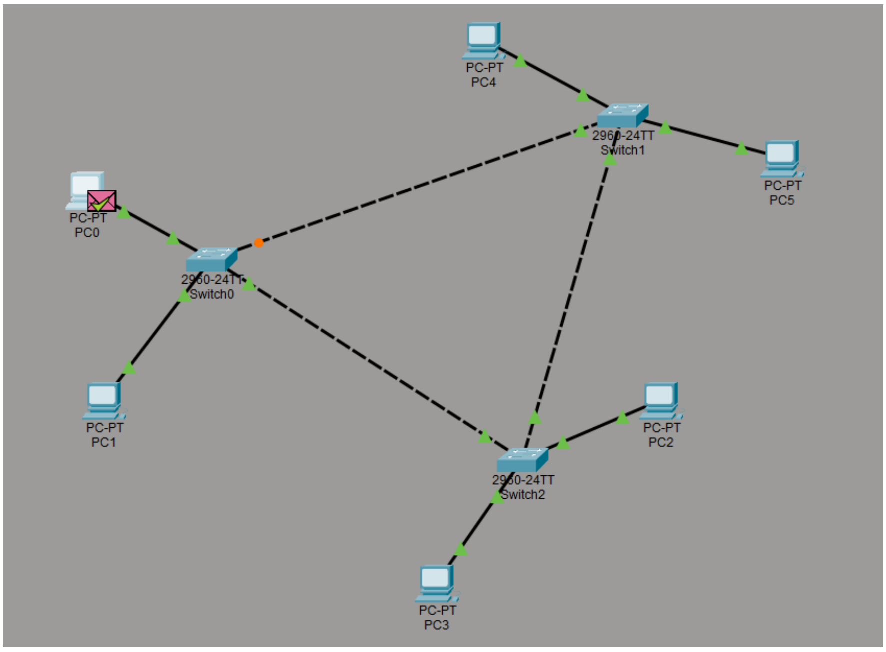

Fig 3f: Network stable — STP maintains loop-free topology throughout

Ping Results

```
PC0(2) > ping 192.168.10.231
Reply from 192.168.10.231: bytes=32 time<1ms TTL=128 (x4)
Packets: Sent=4, Received=4, Lost=0 (0% loss) | Avg=0ms

PC0(2) > ping 192.168.10.221
Reply from 192.168.10.221: bytes=32 time<1ms TTL=128 (x4)
Packets: Sent=4, Received=4, Lost=0 (0% loss) | Max=1ms, Avg=0ms
```

Observation

Devices communicated successfully across multiple switches with zero packet loss. STP automatically elected a root bridge and blocked one redundant link per loop (shown as dashed lines in the screenshots), preventing broadcast storms while keeping alternate paths available. The network remained stable with no loops detected. Traffic was efficiently routed via the STP-selected active links, and all pings completed within sub-millisecond round-trip times.

Task 4 – Failure Test (Ring Topology)

To demonstrate fault tolerance, one active link in the ring topology was deliberately disconnected. This simulates a real-world cable failure or switch port failure. STP was expected to detect the topology change, unblock the previously blocked redundant port, and recalculate a new loop-free path — restoring connectivity without manual intervention.

Network Topology Screenshots

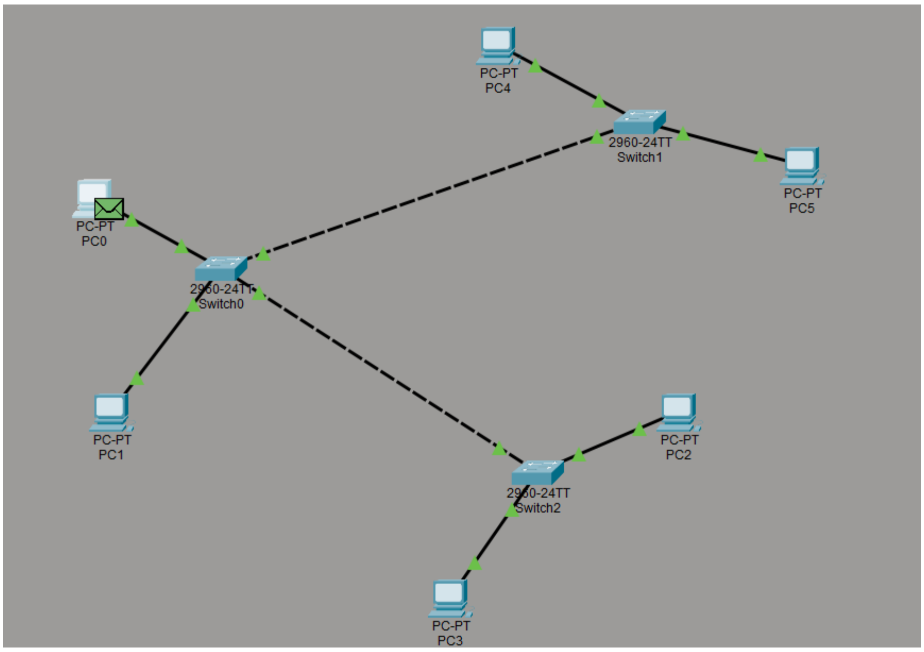

Fig 4a: Active link disconnected — STP detects topology change, begins recalculation

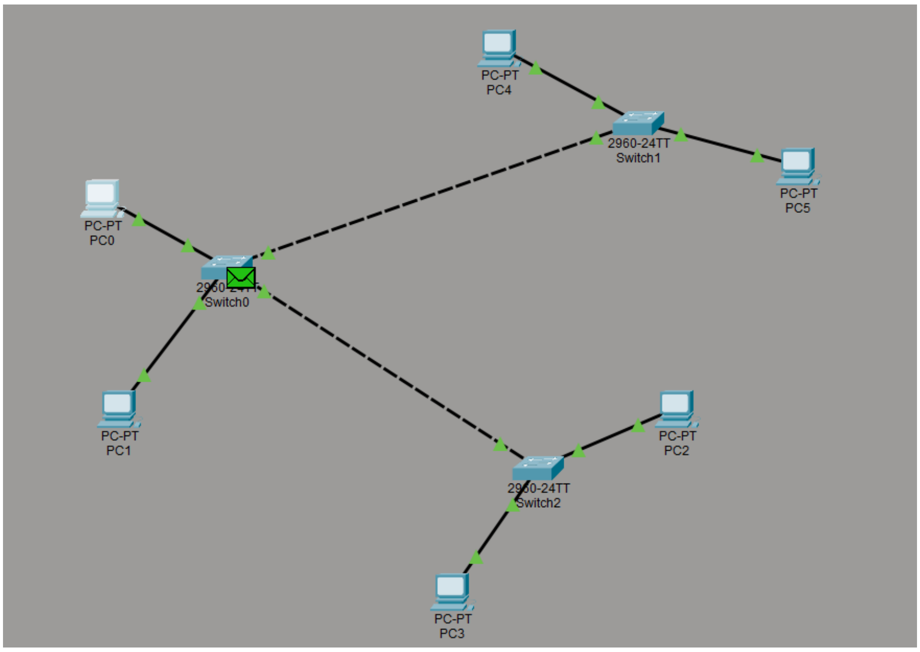

Fig 4b: STP activates previously blocked port to restore connectivity

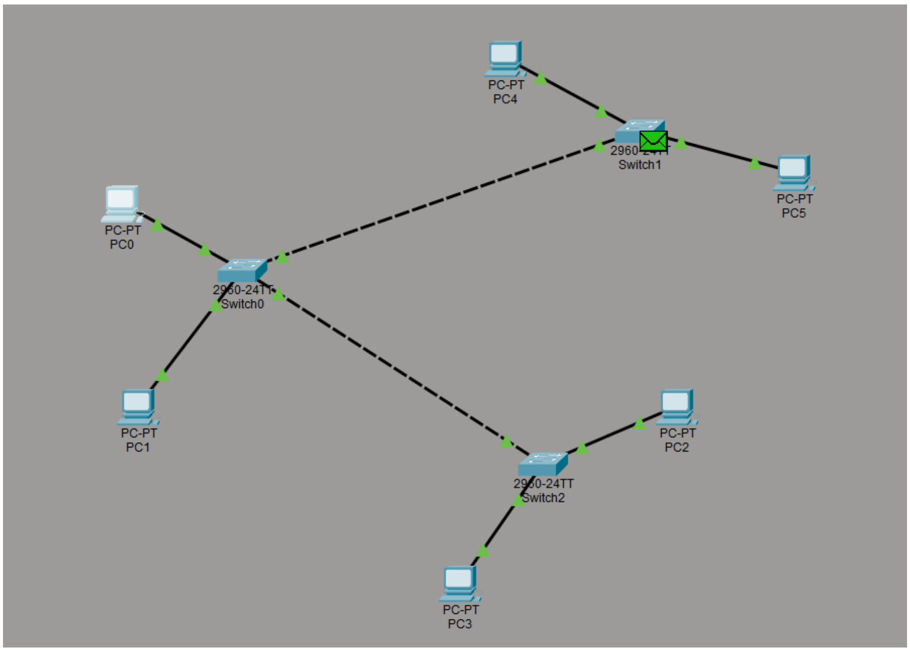

Fig 4c: New active path established — ping test initiated to verify connectivity

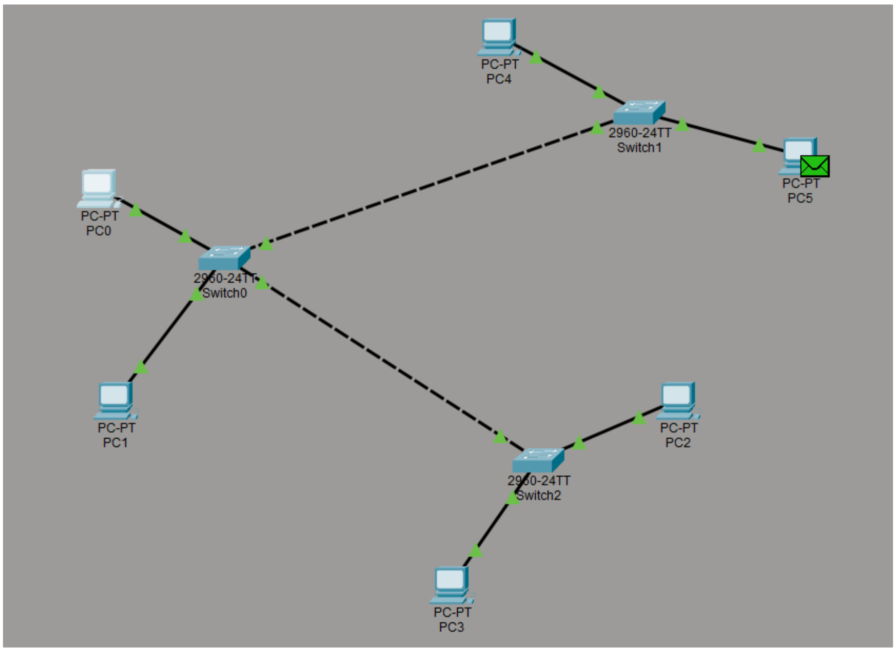

Fig 4d: Reply received — communication restored via alternate path

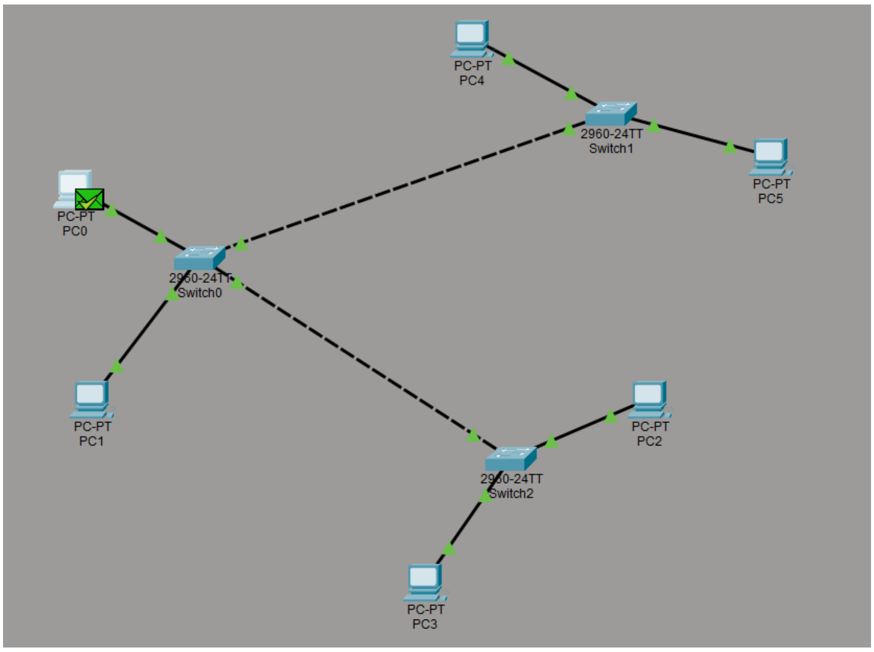

Fig 4e: Network fully stable on alternate path — zero packet loss confirmed

Ping Results

```
PC0(3) > ping 192.168.10.245
Reply from 192.168.10.245: bytes=32 time<1ms TTL=128 (x4)
Packets: Sent=4, Received=4, Lost=0 (0% loss) | Avg=0ms

PC0(3) > ping 192.168.10.243
Reply from 192.168.10.243: bytes=32 time<1ms TTL=128 (x4)
Packets: Sent=4, Received=4, Lost=0 (0% loss) | Max=2ms, Avg=0ms
```

Observation

After the deliberate link disconnection, STP successfully recalculated the topology and activated the previously blocked redundant port, restoring full communication within the STP convergence time. All ping tests completed with zero packet loss, confirming the ring topology's fault tolerance. A slightly elevated maximum RTT of 2ms was observed on one test, attributable to the brief STP recalculation delay. This experiment demonstrates that a ring topology with STP offers significantly higher reliability and resilience compared to a simple star topology, where a single link failure would permanently isolate the affected device.

Conclusion

Topology	Device	Packet Loss	Fault Tolerance	Efficiency
Star	Switch	0%	Low	High (unicast)
Star	Hub	0%	Low	Low (broadcast always)
Ring-like + STP	Switch	0%	Medium	High
Ring Failure Test	Switch	0%	High	High (auto-recovery)

This assignment provided hands-on experience with four fundamental network configurations. The switch-based star topology proved most practical for everyday LANs due to its intelligent frame forwarding and ease of management. The hub-based star, while functionally equivalent in small test environments, revealed inherent inefficiencies that make it unsuitable for modern networks. The ring-like topology with STP demonstrated how redundant links can be exploited for fault tolerance without sacrificing loop-free operation. The failure test conclusively showed that STP-enabled ring networks can self-heal after a link failure with minimal disruption, making them ideal for environments where high availability is critical.