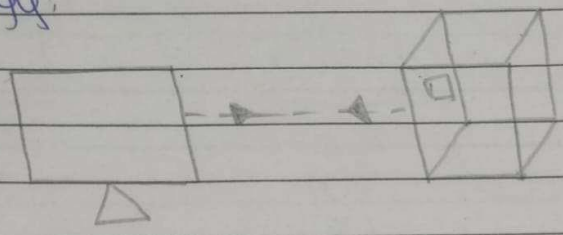
PC to Server:

Mangal
Date
Page

PC to Server

Aim: To set up a point to point b/w a PC and a Server, facilitating direct communication to observe the data exchange.

Topology:

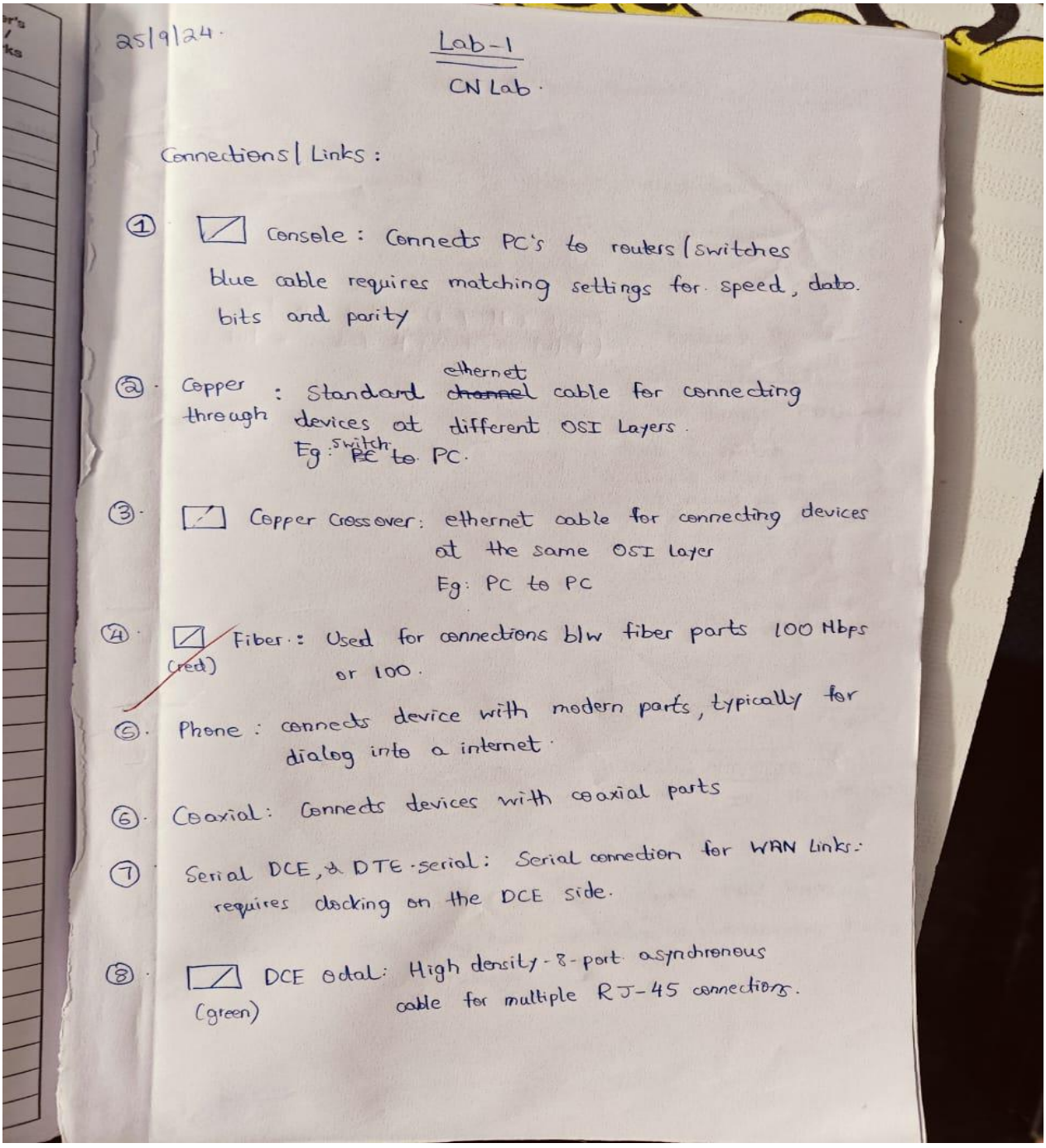


| PC-PT | Server-PT |
|---|---|
| PC0 | Server0 |
| 10.0.0.1 | 10.0.0.2 |

Observation: Direct communication allows PC to communicate with server, which is typically in small networks for tasks such as file sharing, service request or testing server responses to client queue.

Hubs and Switches:

Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message



25/9/24.

Lab-1

CN Lab.

Connections | Links :

① Console : Connects PC's to routers | switches blue cable requires matching settings for speed, data bits and parity

② Copper : Standard ethernet ~~channel~~ cable for connecting through devices at different OSI Layers.
Eg: Switch to PC.

③ Copper Crossover: ethernet cable for connecting devices at the same OSI layer
Eg: PC to PC

④ Fiber : Used for connections blw fiber parts 100 Mbps or 100. (red)

⑤ Phone : connects device with modern parts, typically for dialog into a internet.

⑥ Coaxial : Connects devices with coaxial parts

⑦ Serial DCE, & DTE-serial : Serial connection for WAN Links: requires clocking on the DCE side.

⑧ DCE octal: High density-8-port asynchronous cable for multiple RJ-45 connections. (green)

# Week-1

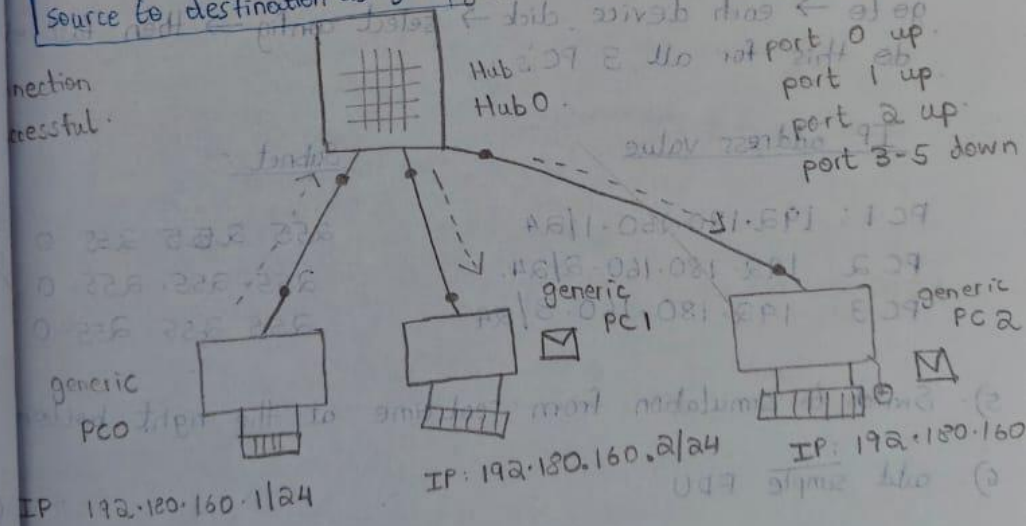
Switch & Hub.

Created a network by selecting end devices. Add. a generic server & PC to the workspace, and used a copper cross over cable.

Sent a simple text message in realtime

4. : To demonstrate the transmission for simple PDU b/w 2 devices connected using a hub & switch.

Aim: Create a topology & simulate sending a simple PDU from source to destination using hub & switch as connecting devices & demonstrate ping msg




nection
ressful.

reate a topology & simulate sending a simple PDU from source to destination using hub and switche as connecting devic and demonstrate ping message.

3) Connect the 3 PCD 12 to switch parts using copper straight through.

4) Setup devices by providing ip's
goto config → fast ethernet → input i/p.

IP address : 192.180.160.1
IP address : 192.180.160.2
IP address : 192.180.160.3

5) Switch to simulation mode, choose visible event/protocol.

6) Add PDU to source and destination to PC2.

7) Run simulation auto capture/play.
ICMP (ping) are sent to only correct device with correct mac.

8). Result switch uses MAC address to intelligently forward ping (ICMP) packets to the destination (PC2) rather broadcasting to all connected devices like a hub. would

| Switches. | Hubs. |
| --- | --- |
| • Direct → MAC. | Broadcast model. |
| • Intelligent. | Not intelligent. |
| • Works in full duplex | Works in half duplex |
| • ~~Only o~~ Multiple devices can send data at same time | only one device can send data at a time. |

7). Result:

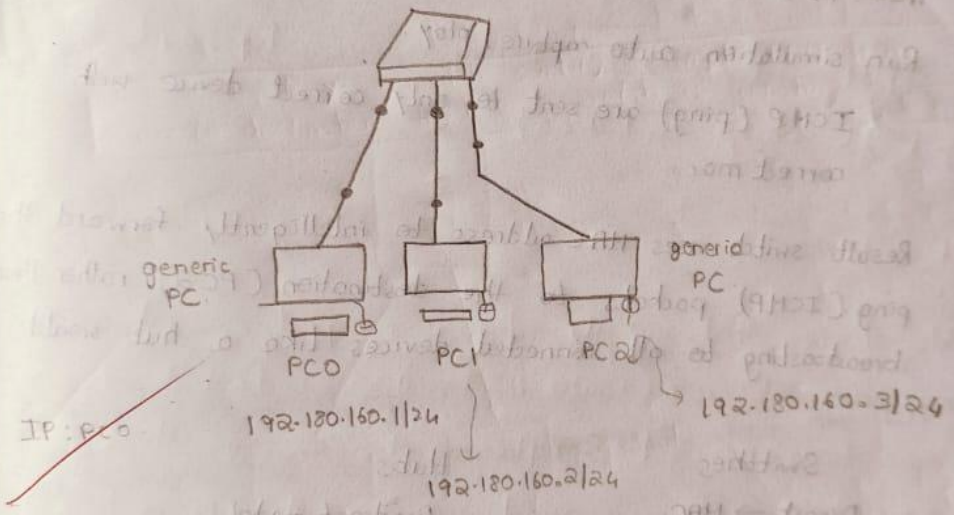
→ From PC0 to p2 & p1 the PDU (protocol data unit).

→ return msgl PDU is sent from PC2 to PC0 that is receiver.

→ But only PC2 will accept the PDU & hence X mark on P1

since it is hub broadcast happens to all connected to a hub. only intended PC will be received.

* Connecting PC to switch:




Observation:

1) Open cisco packet tracer

2) Add device
   switch → network devices
   Network devices - (generic switch).
   end devices
   generic PC.

3) Connect the 3 PCD 12 to switch parts using copper straight through.

4) Setup devices by providing ip's
goto config → fast ethernet → input i/p.

IP address : 192.180.160.1
IP address : 192.180.160.2
IP address : 192.180.160.3

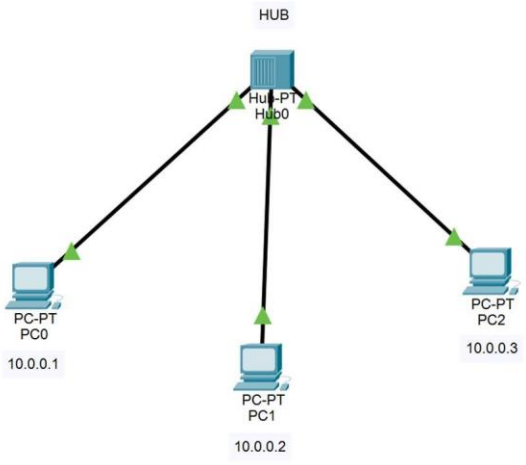
5) Switch to simulation mode, choose visible event/protocol.

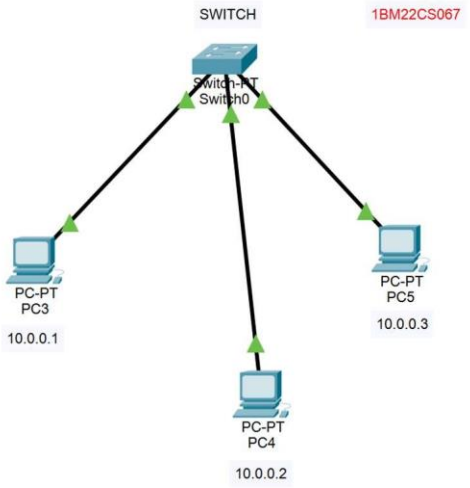
6) Add PDU to source and destination to PC2.

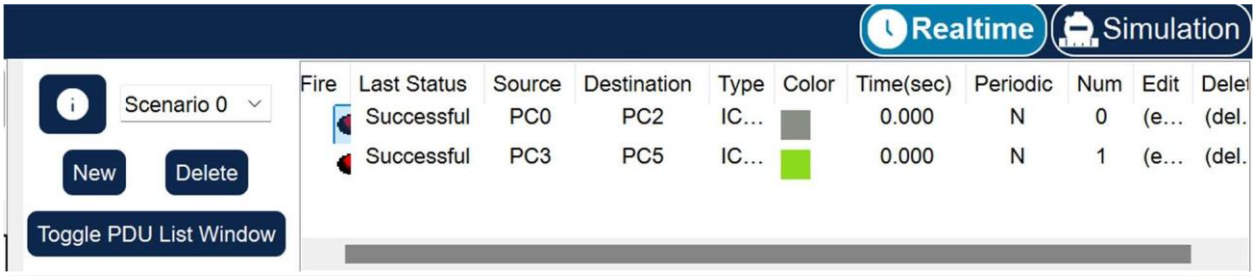
7) Run simulation auto capture/play.
ICMP (ping) are sent to only correct device with correct mac.

8). Result switch uses MAC address to intelligently forward ping (ICMP) packets to the destination (PC2) rather broadcasting to all connected devices like a hub. would

| Switches. | Hubs. |
| --- | --- |
| • Direct → MAC. | Broadcast model. |
| • Intelligent. | Not intelligent. |
| • Works in full duplex | Works in half duplex |
| • ~~Only o~~ Multiple devices can send data at same time | only one device can send data at a time. |




Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```