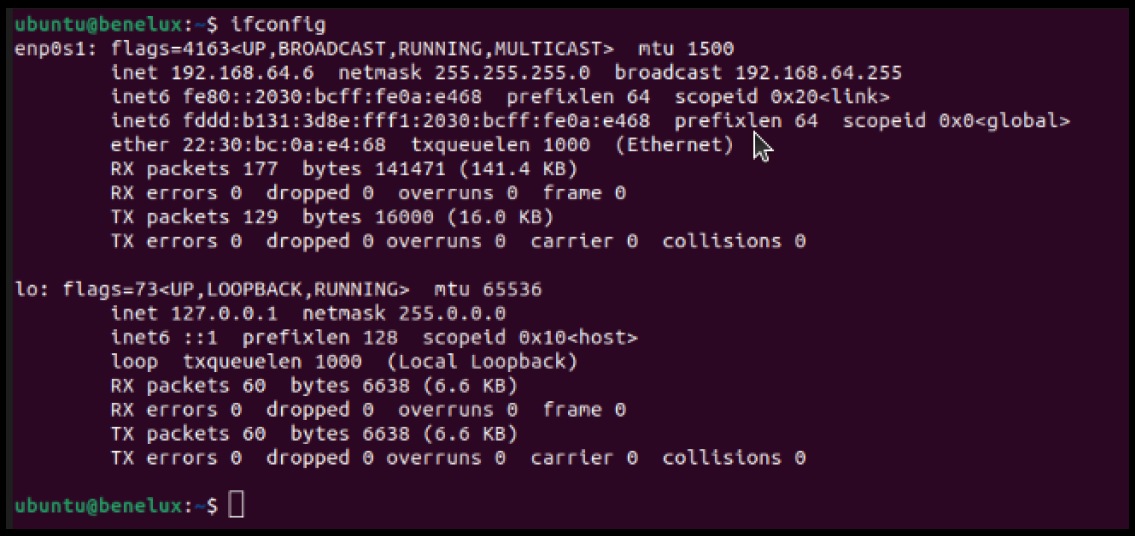
1. Basics of terminal:

1) **ifconfig**:- CLI command to list the network interfaces
   General syntax



$ ifconfig [interface] [options]

Network interfaces in my system are listed

1) enp0s1 - An Ethernet interface on the first PCI bus, slot 1.These names provide more consistent and predictable interface names based on the hardware configuration.
   - **IP Address (inet)**: 192.168.64.6
   - **Subnet Mask (netmask)**: 255.255.255.0
   - **Broadcast Address**: 192.168.64.255
   - **MAC Address (ether)**: 22:30:bc:0a:e4:68

2). lo - A lookback network interface in linux, its a virtual network interface used by the system to communicate with itself.

- IP Address (inet): 127.0.0.1
- **Subnet Mask (netmask)**: 255.0.0.0
- **Broadcast Address**: (not applicable for loopback)
- **MAC Address**: (not applicable for loopback)

ifconfig is primarily used for the system administration utility in Unix-like operating systems used for network interface configuration

2) Assign a Static IP Address

```
ubuntu@benelux:~$ ifconfig
enp0s1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.64.6  netmask 255.255.255.0  broadcast 192.168.64.255
        inet6 fe80::2030:bcff:fe0a:e468  prefixlen 64  scopeid 0x20<link>
        inet6 fddd:b131:3d8e:fff1:2030:bcff:fe0a:e468  prefixlen 64  scopeid 0x0<global>
        ether 22:30:bc:0a:e4:68  txqueuelen 1000  (Ethernet)
        RX packets 177  bytes 141471 (141.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 129  bytes 16000 (16.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 60  bytes 6638 (6.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 60  bytes 6638 (6.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ubuntu@benelux:~$
```

Given for the enp0s (local host) ip address:- 192.168.64.6 with the subnet mask :- 255.255.255.0 (192.168.64.6/24)

so,Binary form of both

Binary form	1st octets	2nd octets	3rd octets	4th octets
ip address	192	168	64	6
	11000000	10101000	01000000	00000110
subnet mask	255	255	255	00000000
	11111111	11111111	11111111	00000000

To find the starting address in the following subnet mask, we simply do binary “and” operation between the IP address and the subnet mask:

Binary form				
subnet mask	11111111	11111111	11111111	00000000
ip address	11000000	10101000	01000000	00000110
First ip address	11000000	10101000	10000000	00000000

First ip address in digits :- 192.168.64.0

for last IP address by applying the "**or**" operation on it with the bitwise binary inverse of the subnet mask to the first IP address

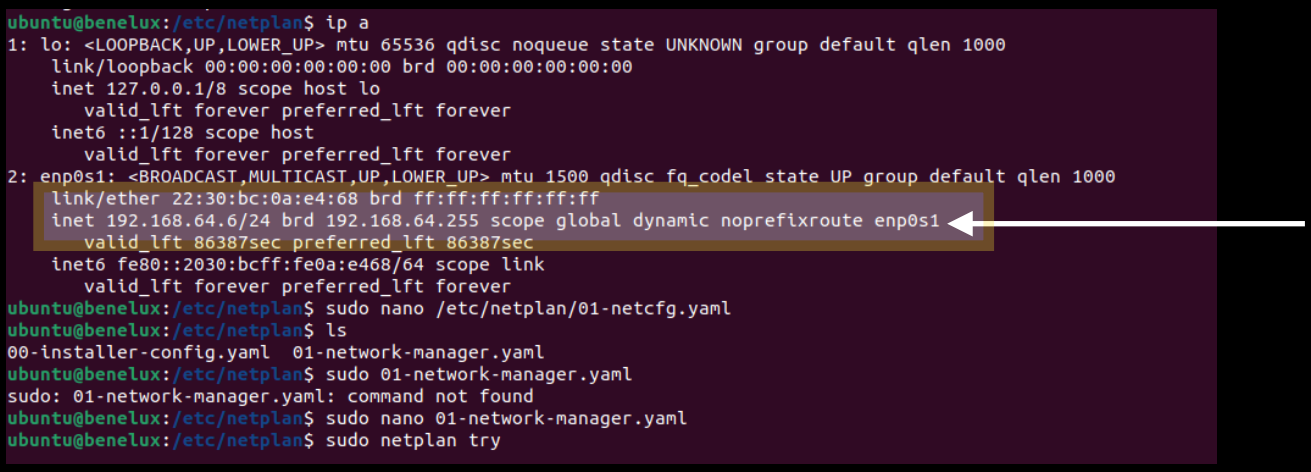
Binary form				
ip address	192	168	64	6
	11000000	10101000	01000000	00000110
inverse subnet	00000000	00000000	00000000	11111111
last ip address	11000000	10101000	01000000	11111111

last ip address :- 192.168.64.255

therefore usable ip address range is **192.168.64.1-192.168.64.254**

Using Netplan with proper yaml script.
Static ip 192.168.64.69 is been assign to enp0s(local host)

Before yaml script
Source :-https://www.freecodecamp.org/news/setting-a-static-ip-in-ubuntu-linux-ip-address-tutorial/

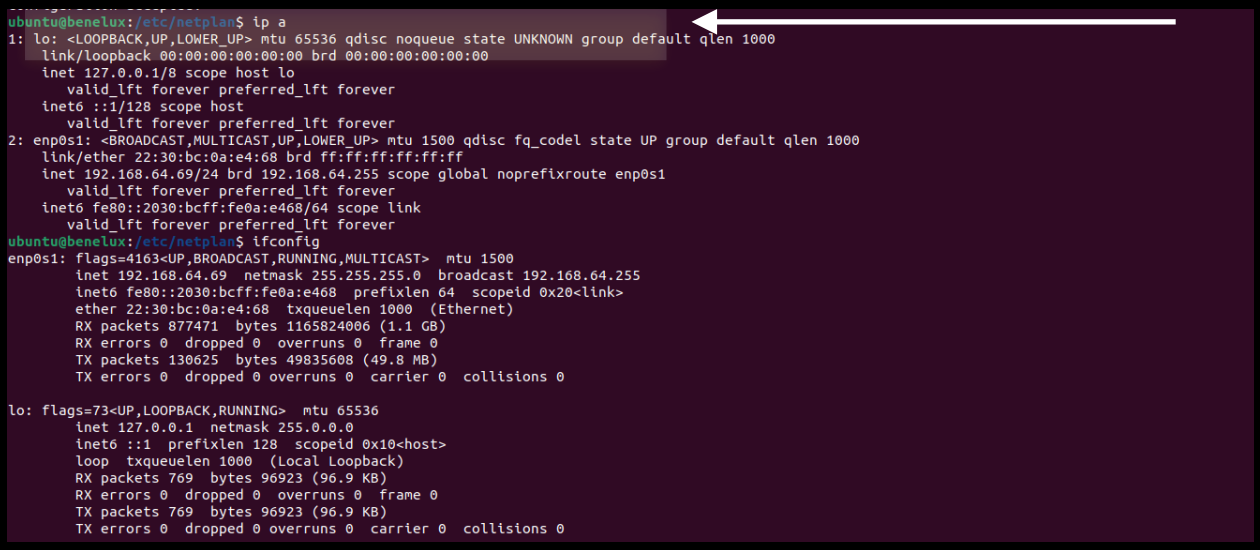


After yaml script

3)Ping

General syntax
$ ping [hostname or ip address]

There are two ways in which a Remote server can be ping
a) with a hostname
eg) $ ping facebook.com
b)with a ip address



eg) ping 142.251.42.110

switch in ping

```
➜  ~ ping -c 5 google.com
PING google.com (142.250.192.78): 56 data bytes
64 bytes from 142.250.192.78: icmp_seq=0 ttl=112 time=54.282 ms
64 bytes from 142.250.192.78: icmp_seq=1 ttl=112 time=43.756 ms
64 bytes from 142.250.192.78: icmp_seq=2 ttl=112 time=37.672 ms
64 bytes from 142.250.192.78: icmp_seq=3 ttl=112 time=32.693 ms
64 bytes from 142.250.192.78: icmp_seq=4 ttl=112 time=37.840 ms

--- google.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 32.693/41.249/54.282/7.400 ms
```

1) -c to specify number of ping

```
➜  ~ ping -q google.com
PING google.com (142.250.192.78): 56 data bytes
^C
--- google.com ping statistics ---
5 packets transmitted, 4 packets received, 20.0% packet loss
round-trip min/avg/max/stddev = 37.208/40.080/47.695/4.410 ms
```

2) -q quiet mode

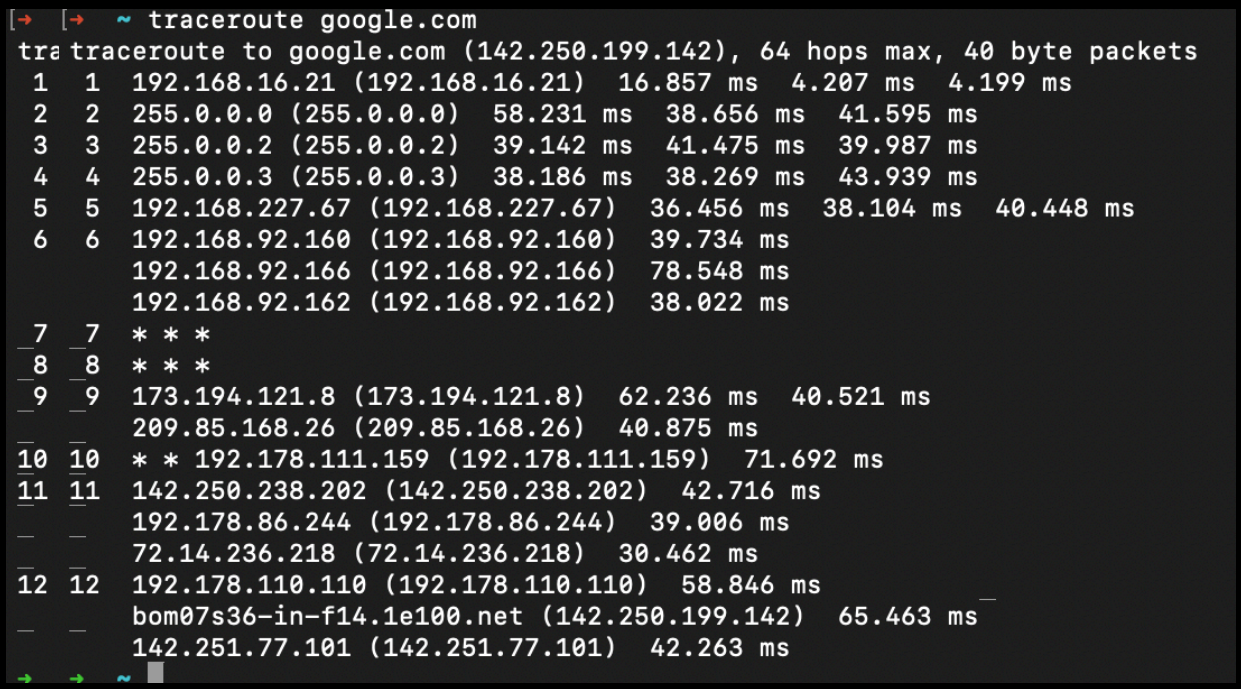
Some important terms
Ping :-Ping is a network utility tool. Based on ICMP
Packets:-packet is a small unit of data transmitted over a network.
Packet loss :-packet loss means that the ICMP Echo Request packets sent to the target host did not receive a corresponding ICMP Echo Reply

```
➜  ~ ping facebook.com
PING facebook.com (163.70.140.35): 56 data bytes
64 bytes from 163.70.140.35: icmp_seq=0 ttl=51 time=45.633 ms
64 bytes from 163.70.140.35: icmp_seq=1 ttl=51 time=40.200 ms
64 bytes from 163.70.140.35: icmp_seq=2 ttl=51 time=34.579 ms
64 bytes from 163.70.140.35: icmp_seq=3 ttl=51 time=35.278 ms
64 bytes from 163.70.140.35: icmp_seq=4 ttl=51 time=36.518 ms
^C
--- facebook.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 34.579/38.442/45.633/4.085 ms
```

Minimum round trip time = 34.579 ms
Maximum round trip time = 45.633 ms
Average round trip time =38.442 ms
Standard deviation of round trip time =4.085 ms

4) Traceroute

General syntax
$ traceroute [options] [destination]
Traceroute is used primarily for diagnostic purposes
for eg;
Hops:- refers to each step or point that a data packet passes through on its journey from the source to the destination. Each hop represents a router or a gateway that forwards the packet closer to its final destination.
Here node yellow is source sending packets towards green node which is destination node.In between the white nodes represented can be local gateway for our subnet or router. At each step along the path, traceroute identifies the hop's IP as well as the latency to that hop.
Here a lots of information can be gathered with the above output.
traceroute command send's three packet to destinations and the round trip time of each packet is listed.



Our first hop does't tell us much because its local gateway for our subnet.The first few hops (1-6) involve IP addresses within the local network (192.168.x.x and 255.x.x.x). These indicate internal routing within the private network.

Hops 7 and 8 show * * *, indicating that the requests timed out, and no response was received from these routers. These router might be busy with another network request. As traceroute is low priority request its bypasses its without tracing router data.

From hop 9 onwards, the packets traverse through various public IP addresses:

Hop 9: 173.194.121.8 and 209.85.168.26, which are associated with Google's network.
Hop 10: The * * 192.178.111.159 line indicates another timeout for the first two attempts, but the third attempt succeeded.
Hop 11: 142.250.238.202, another IP within Google's network.
Hop 12: The final destination is reached at bom07s36-in-f14.1e100.net (142.250.199.142) as name suggest this packet might went to Mumbai and another associated IP (142.251.77.101)

The response times (in milliseconds) generally increase as the packet moves further from the source to the destination, as expected.

2. Basic Network Topology Module

1. Star Topology

Description: In a star topology, all devices (nodes) are connected to a central hub or switch. Each device communicates directly with the hub, which manages and controls the flow of data within the network.

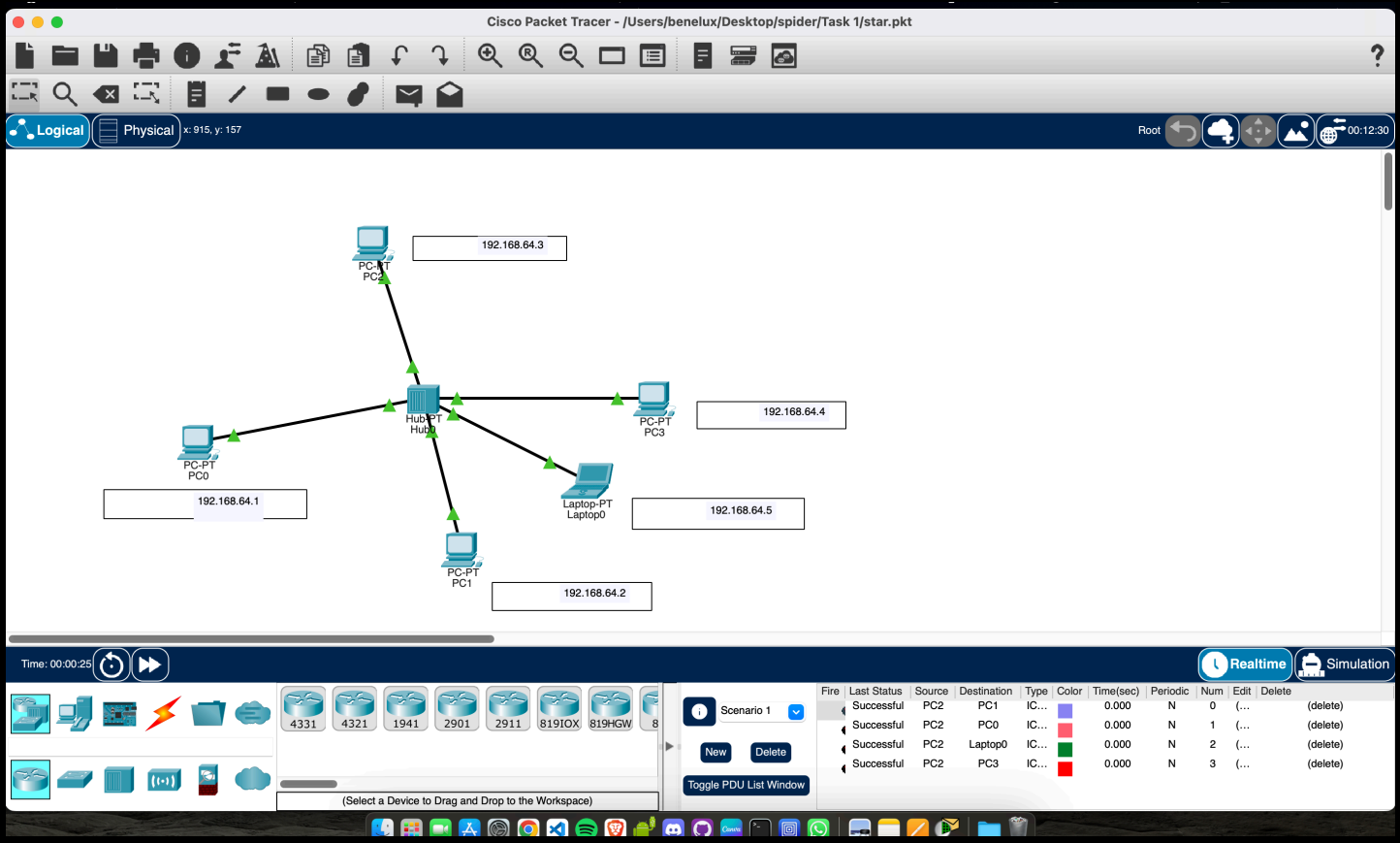
Characteristics:

- **Centralised Management:** The hub or switch facilitates easy management and troubleshooting as all communication flows through it.
- **Scalability:** Adding or removing devices is straightforward without affecting the rest of the network.
- **Reliability:** If one connection fails, only the affected device loses connectivity, while others remain unaffected.
- **Performance:** Each device can communicate independently without affecting others, leading to potentially better performance compared to other topologies.

Performance Evaluation:

- **Ping Test Results:** Conducting ping tests between devices in a star topology typically shows low latency and reliable communication. If there are issues, they are often related to the hub or switch.

Conclusion: Star topologies are ideal for environments where centralized control and easy scalability are important. They offer good performance and reliability for networks with moderate numbers of devices.



2. Bus Topology

Description: A bus topology consists of a single central cable (the bus), to which all devices are connected. Data travels along the bus and is received by all devices, but each device only processes data intended for it.

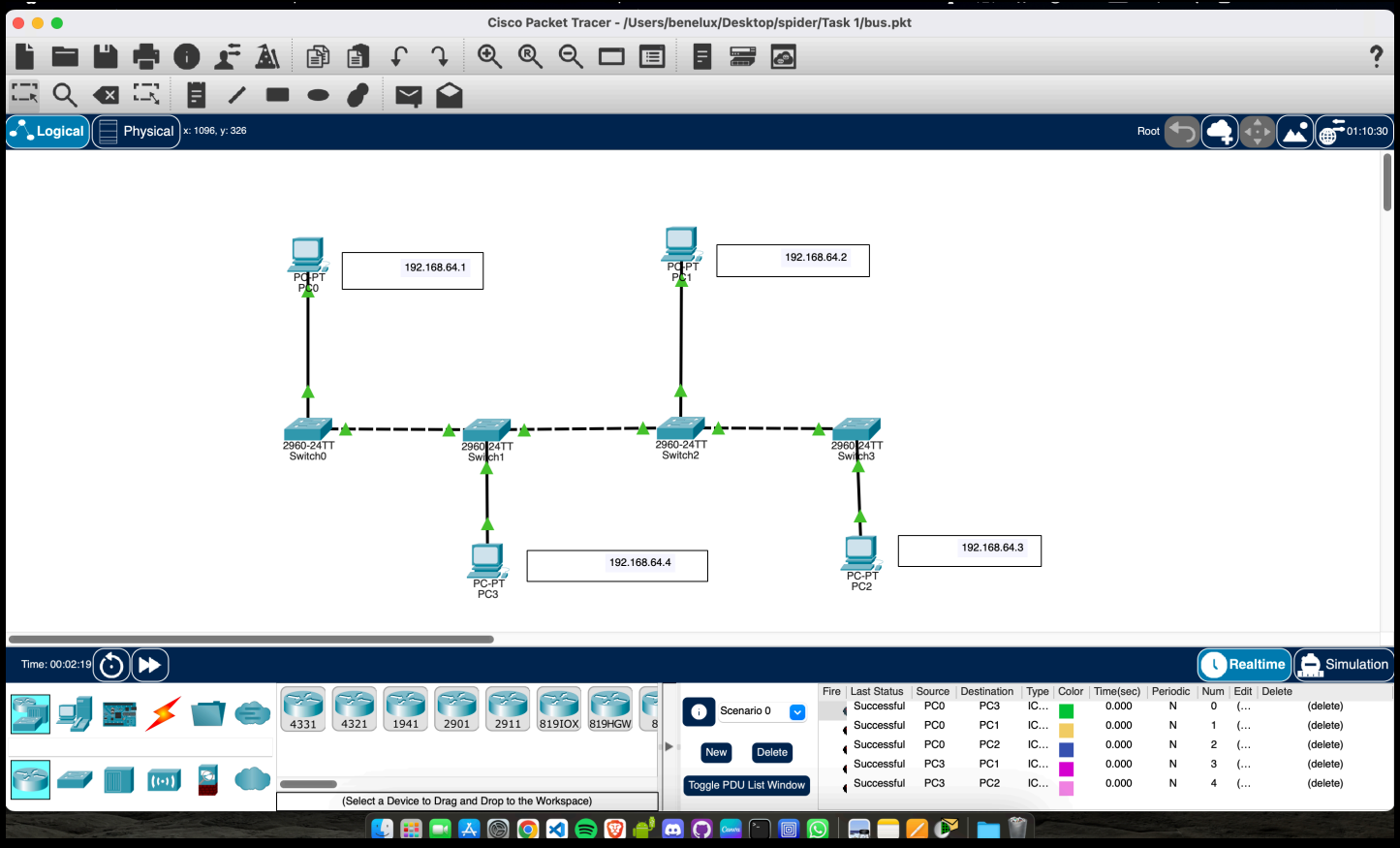
Characteristics:

- **Simplicity:** Easy to set up and requires minimal cabling compared to other topologies.
- **Cost-Effectiveness:** Suitable for small networks with limited budgets.
- **Performance:** As more devices are added, the bus can become a bottleneck, affecting overall network performance.
- **Fault Tolerance:** Failure of the main bus cable can disrupt the entire network.

Performance Evaluation:

- **Ping Test Results:** Ping tests can show varying latency depending on network load. Collisions and data collisions might occur, affecting overall performance.

Conclusion: Bus topologies are economical and simple but may not scale well for larger networks due to potential performance limitations and reliability issues.



3. Ring Topology

Description: In a ring topology, each device is connected directly to two other devices, forming a closed loop. Data travels in one direction around the ring until it reaches its destination.

Characteristics:

- **Unidirectional Data Flow:** Data flows through the network in a single direction, ensuring predictable pathways.
- **Fault Isolation:** If one device or connection fails, it can disrupt the entire network as there are no alternative paths.
- **Performance:** Ring networks can suffer from latency as data must pass through each device in the ring.
- **Reliability:** The network is as reliable as its weakest link; failure of one device affects the entire network.

Performance Evaluation:

- **Ping Test Results:** Ping tests can reveal consistent latency as data travels through each device in the ring.

Conclusion: Ring topologies offer predictable data flow but are less scalable and less fault-tolerant compared to other topologies. They are suitable for small, closed networks where performance requirements are modest.