

Name: Bhavesh Kewalramani

Roll No.: A-25

Section: A

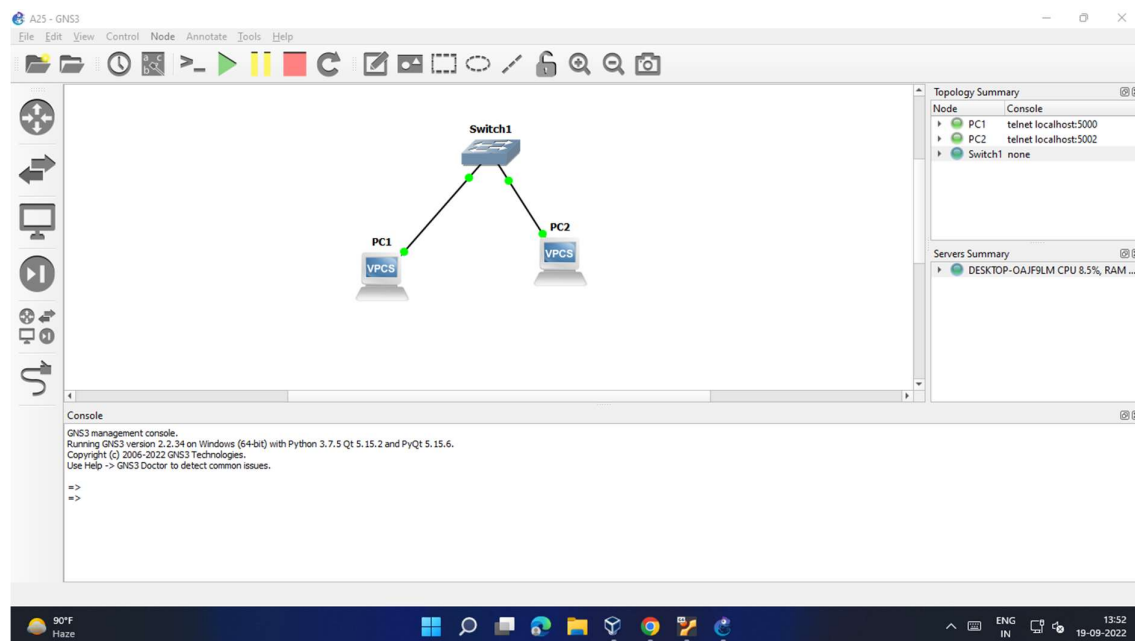
Semester: VII

Shift: I

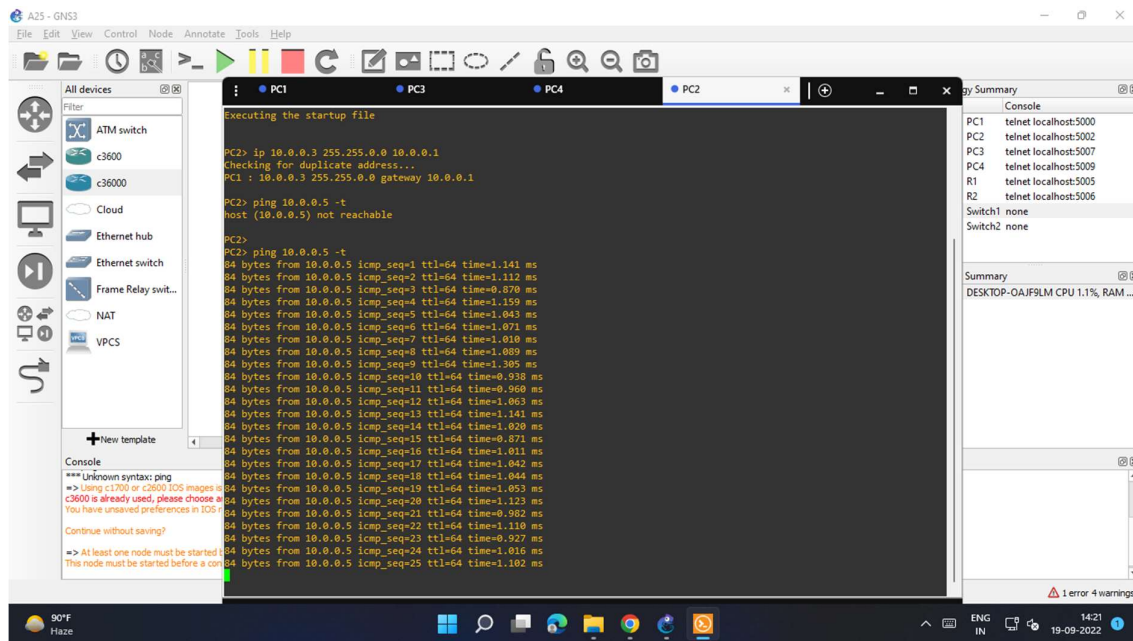
Batch: A1

Aim: Creation of simple network topology using open source network virtualization tool GNS3. Simulating basic network topologies (Tree and Ring) through the GNS3 software.

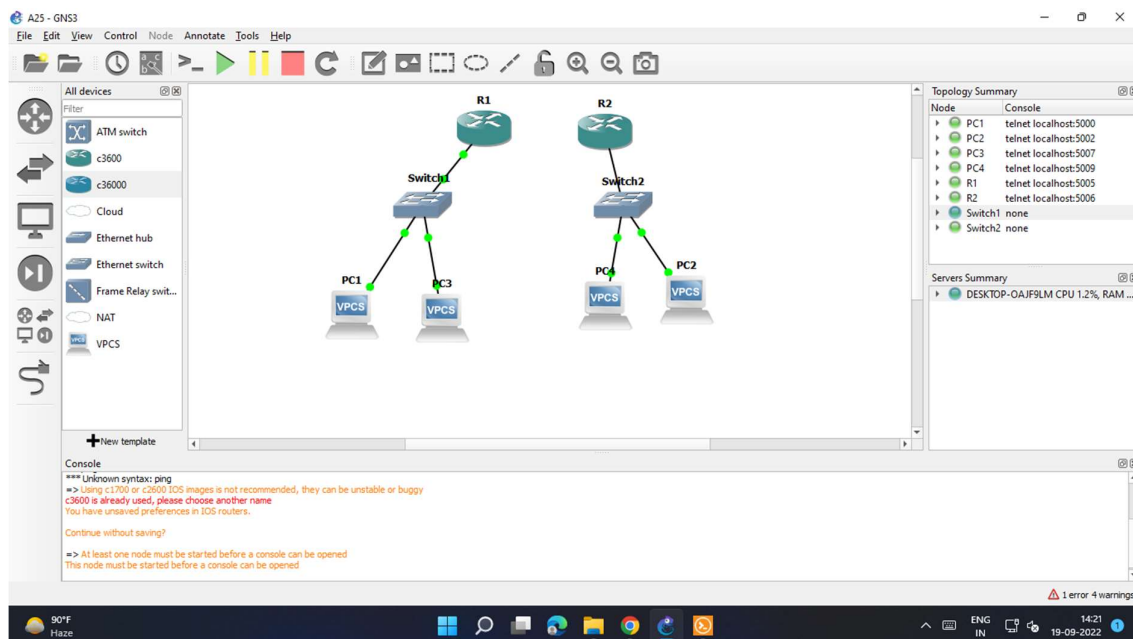
Outputs:



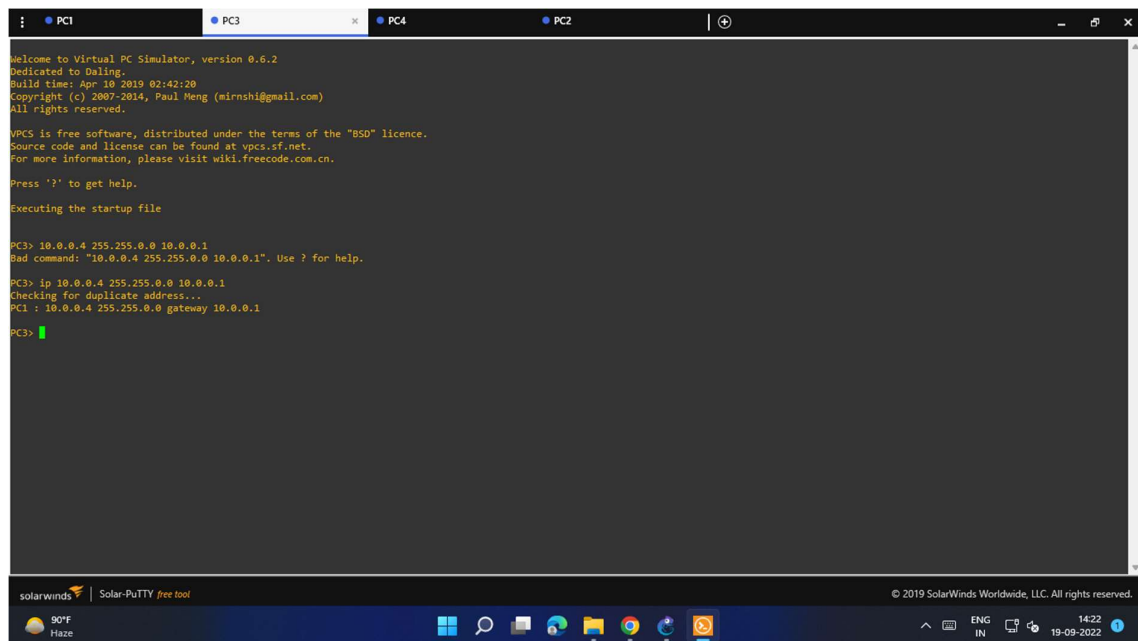
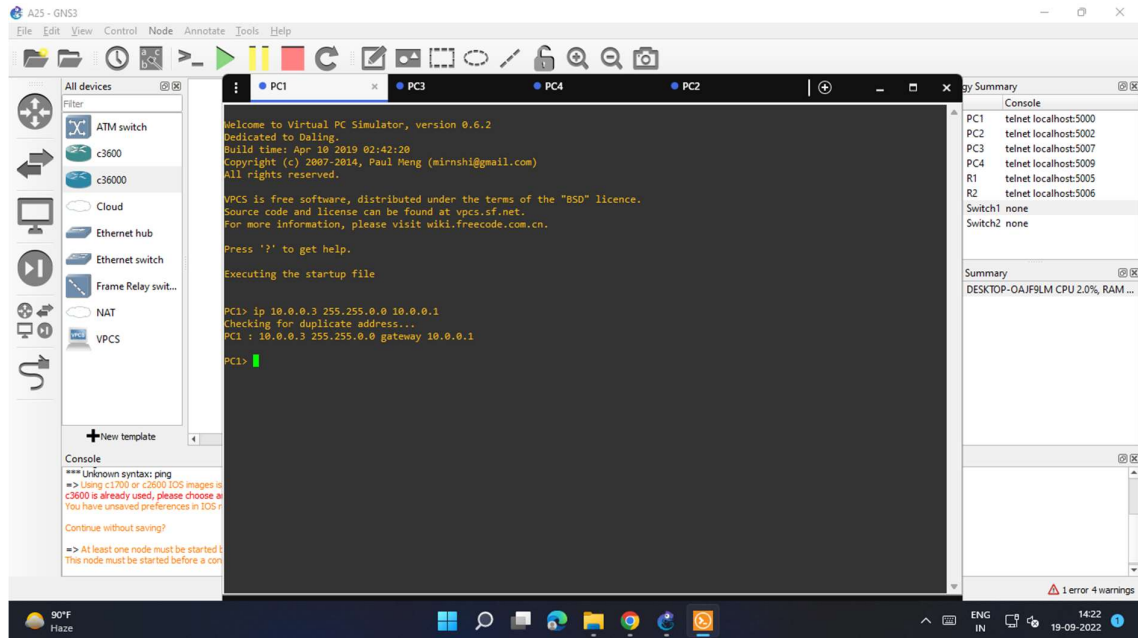
Using Switch to connect two Personal Computers. Here Switch1 is the Switch and PC1 and PC2 are Personal Computers

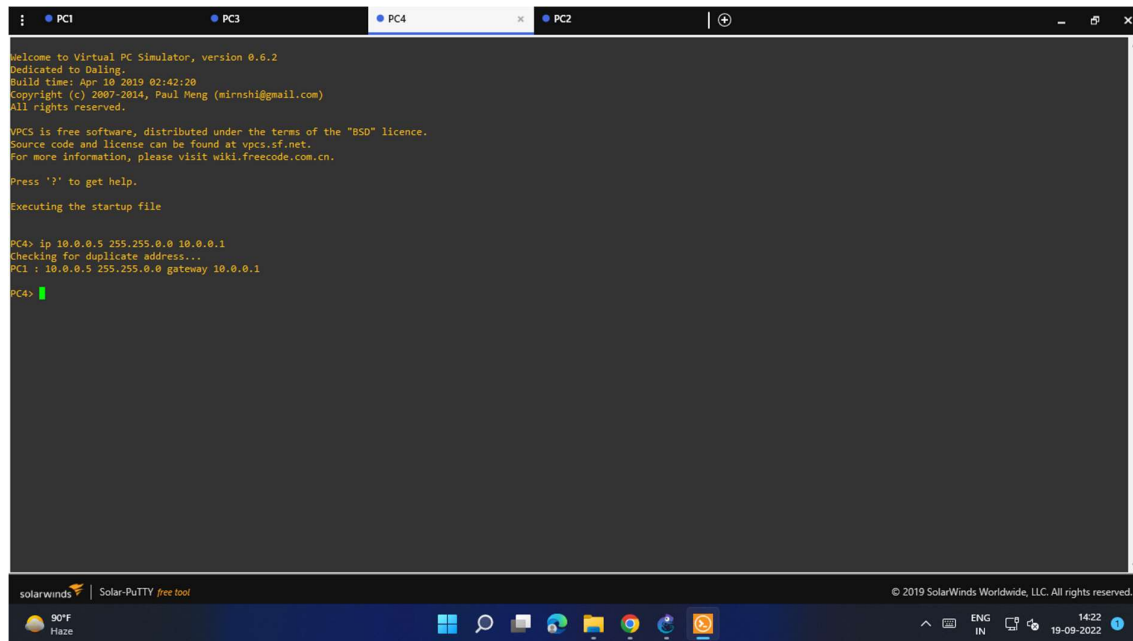


Assigning the IP address to each Personal Computer and using ping command to check if they are reachable or not

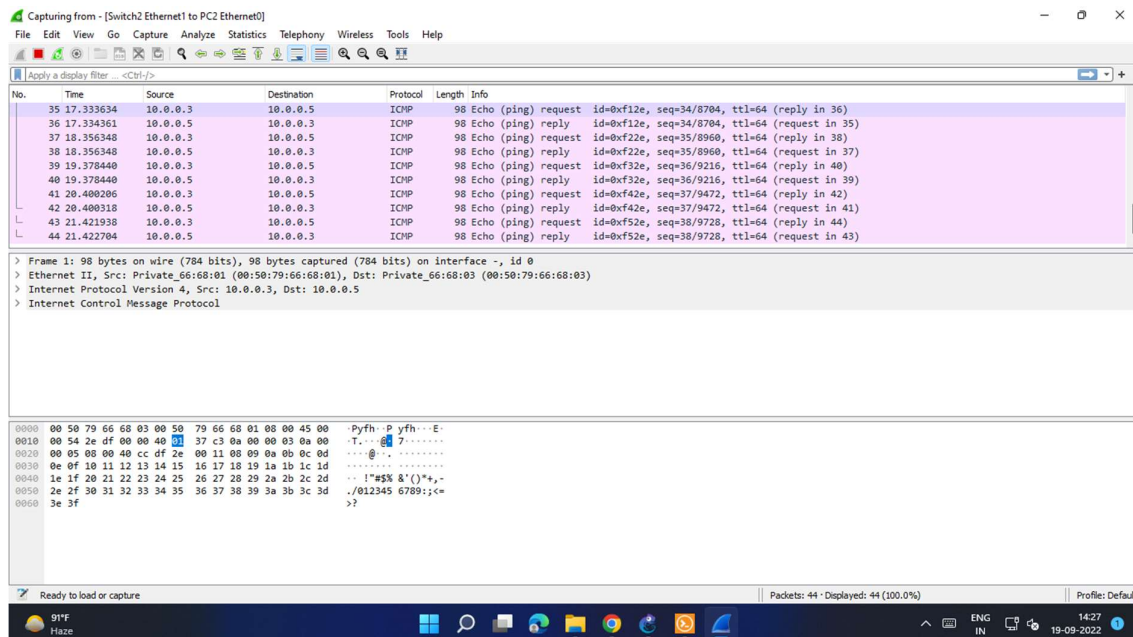


Using Router (R1and R2) to connect the two Personal Computers (PC1-PC3,PC2-PC4) with the help of Switch (Switch1, Switch2)

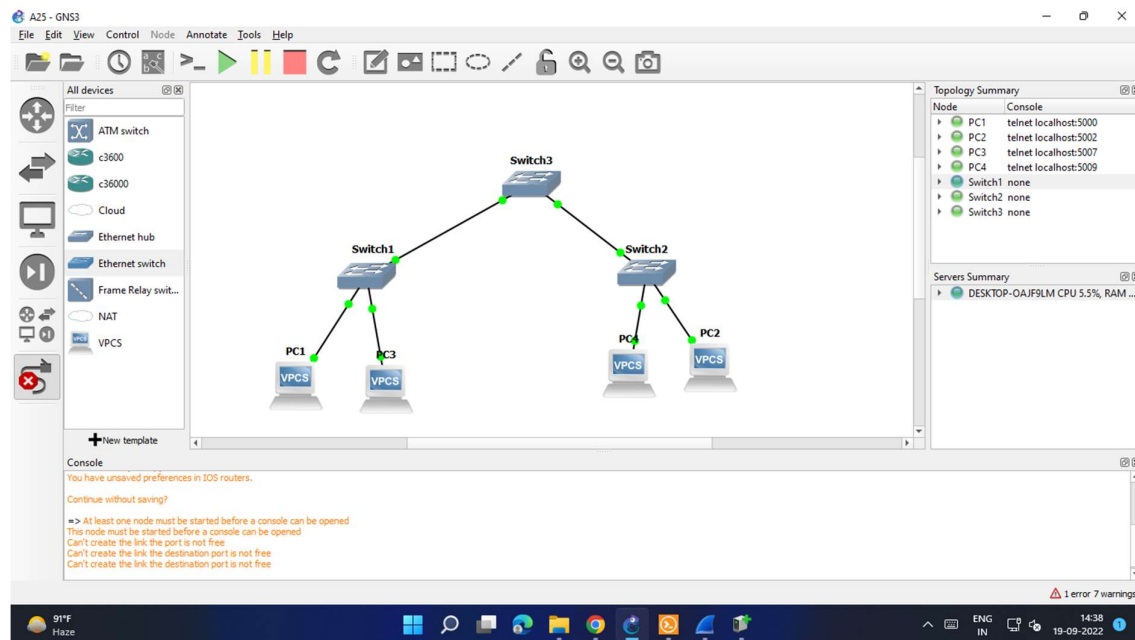




Assigning the IP address to each Personal Computer and using ping command to check if they are reachable or not



Using Wireshark Software to keep a check on the requests and replies made by one Personal Computer to Another



Creating a Tree Topology using Personal Computers and Switches

The screenshot shows a SolarWinds Solar-PuTTY terminal window with a tab for PC2. The terminal displays the output of a series of ping commands from 10.0.0.5 to 10.0.0.5. The first five commands result in timeouts, while the subsequent 28 commands show successful replies from 10.0.0.5 with varying TTL and time values.

```

10.0.0.5 icmp_seq=759 timeout
10.0.0.5 icmp_seq=760 timeout
10.0.0.5 icmp_seq=761 timeout
10.0.0.5 icmp_seq=762 timeout
10.0.0.5 icmp_seq=763 timeout
10.0.0.5 icmp_seq=764 timeout
10.0.0.5 icmp_seq=765 timeout
84 bytes from 10.0.0.5: icmp_seq=766 ttl=64 time=1.021 ms
84 bytes from 10.0.0.5: icmp_seq=767 ttl=64 time=1.159 ms
84 bytes from 10.0.0.5: icmp_seq=768 ttl=64 time=1.205 ms
84 bytes from 10.0.0.5: icmp_seq=769 ttl=64 time=1.258 ms
84 bytes from 10.0.0.5: icmp_seq=770 ttl=64 time=1.442 ms
84 bytes from 10.0.0.5: icmp_seq=771 ttl=64 time=1.475 ms
84 bytes from 10.0.0.5: icmp_seq=772 ttl=64 time=1.191 ms
84 bytes from 10.0.0.5: icmp_seq=773 ttl=64 time=1.348 ms
84 bytes from 10.0.0.5: icmp_seq=774 ttl=64 time=1.621 ms
84 bytes from 10.0.0.5: icmp_seq=775 ttl=64 time=1.321 ms
84 bytes from 10.0.0.5: icmp_seq=776 ttl=64 time=1.166 ms
84 bytes from 10.0.0.5: icmp_seq=777 ttl=64 time=1.352 ms
84 bytes from 10.0.0.5: icmp_seq=778 ttl=64 time=1.318 ms
84 bytes from 10.0.0.5: icmp_seq=779 ttl=64 time=1.424 ms
84 bytes from 10.0.0.5: icmp_seq=780 ttl=64 time=1.441 ms
84 bytes from 10.0.0.5: icmp_seq=781 ttl=64 time=1.319 ms
84 bytes from 10.0.0.5: icmp_seq=782 ttl=64 time=1.527 ms
84 bytes from 10.0.0.5: icmp_seq=783 ttl=64 time=1.519 ms
84 bytes from 10.0.0.5: icmp_seq=784 ttl=64 time=1.380 ms
84 bytes from 10.0.0.5: icmp_seq=785 ttl=64 time=1.569 ms
84 bytes from 10.0.0.5: icmp_seq=786 ttl=64 time=1.346 ms
84 bytes from 10.0.0.5: icmp_seq=787 ttl=64 time=1.318 ms
84 bytes from 10.0.0.5: icmp_seq=788 ttl=64 time=1.362 ms
84 bytes from 10.0.0.5: icmp_seq=789 ttl=64 time=1.242 ms
84 bytes from 10.0.0.5: icmp_seq=790 ttl=64 time=1.283 ms
84 bytes from 10.0.0.5: icmp_seq=791 ttl=64 time=1.185 ms
84 bytes from 10.0.0.5: icmp_seq=792 ttl=64 time=1.296 ms
84 bytes from 10.0.0.5: icmp_seq=793 ttl=64 time=1.160 ms
84 bytes from 10.0.0.5: icmp_seq=794 ttl=64 time=1.329 ms
84 bytes from 10.0.0.5: icmp_seq=795 ttl=64 time=1.466 ms
84 bytes from 10.0.0.5: icmp_seq=796 ttl=64 time=1.265 ms
84 bytes from 10.0.0.5: icmp_seq=797 ttl=64 time=1.338 ms
84 bytes from 10.0.0.5: icmp_seq=798 ttl=64 time=1.300 ms
  
```

Assigning the IP address to each Personal Computer and using ping command to check if they are reachable or not. Here the ping command is used between PC1 and PC2.

The screenshot shows a Solar-PuTTY terminal window with a dark background and yellow text. The text displays the output of a ping command from PC3 to PC4. It starts with three 'queue is full' messages, followed by a '10.0.0.3 icmp_seq=555 timeout' message. Then, it shows a series of successful ping responses from 10.0.0.3 to 10.0.0.4, each with a sequence number (icmp_seq) and a time value. The sequence numbers range from 556 to 591. The times are in milliseconds (ms). At the bottom of the terminal window, there is a status bar showing 'solarwinds Solar-PuTTY free tool' and '© 2019 SolarWinds Worldwide, LLC. All rights reserved.' Below the terminal window, there is a Windows taskbar with a weather widget showing '91°F Haze' and a system clock showing '14:40 19-09-2022'.

```
queue is full
queue is full
queue is full
10.0.0.3 icmp_seq=555 timeout
64 bytes from 10.0.0.3: icmp_seq=556 ttl=64 time=1.054 ms
64 bytes from 10.0.0.3: icmp_seq=557 ttl=64 time=1.431 ms
64 bytes from 10.0.0.3: icmp_seq=558 ttl=64 time=1.307 ms
64 bytes from 10.0.0.3: icmp_seq=559 ttl=64 time=1.216 ms
64 bytes from 10.0.0.3: icmp_seq=560 ttl=64 time=1.127 ms
64 bytes from 10.0.0.3: icmp_seq=561 ttl=64 time=1.304 ms
64 bytes from 10.0.0.3: icmp_seq=562 ttl=64 time=1.302 ms
64 bytes from 10.0.0.3: icmp_seq=563 ttl=64 time=1.187 ms
64 bytes from 10.0.0.3: icmp_seq=564 ttl=64 time=1.638 ms
64 bytes from 10.0.0.3: icmp_seq=565 ttl=64 time=1.263 ms
64 bytes from 10.0.0.3: icmp_seq=566 ttl=64 time=1.303 ms
64 bytes from 10.0.0.3: icmp_seq=567 ttl=64 time=1.239 ms
64 bytes from 10.0.0.3: icmp_seq=568 ttl=64 time=1.203 ms
64 bytes from 10.0.0.3: icmp_seq=569 ttl=64 time=1.378 ms
64 bytes from 10.0.0.3: icmp_seq=570 ttl=64 time=1.341 ms
64 bytes from 10.0.0.3: icmp_seq=571 ttl=64 time=1.399 ms
64 bytes from 10.0.0.3: icmp_seq=572 ttl=64 time=1.538 ms
64 bytes from 10.0.0.3: icmp_seq=573 ttl=64 time=1.420 ms
64 bytes from 10.0.0.3: icmp_seq=574 ttl=64 time=1.342 ms
64 bytes from 10.0.0.3: icmp_seq=575 ttl=64 time=1.504 ms
64 bytes from 10.0.0.3: icmp_seq=576 ttl=64 time=1.253 ms
64 bytes from 10.0.0.3: icmp_seq=577 ttl=64 time=1.246 ms
64 bytes from 10.0.0.3: icmp_seq=578 ttl=64 time=1.618 ms
64 bytes from 10.0.0.3: icmp_seq=579 ttl=64 time=1.303 ms
64 bytes from 10.0.0.3: icmp_seq=580 ttl=64 time=1.290 ms
64 bytes from 10.0.0.3: icmp_seq=581 ttl=64 time=1.252 ms
64 bytes from 10.0.0.3: icmp_seq=582 ttl=64 time=1.267 ms
64 bytes from 10.0.0.3: icmp_seq=583 ttl=64 time=1.371 ms
64 bytes from 10.0.0.3: icmp_seq=584 ttl=64 time=1.205 ms
64 bytes from 10.0.0.3: icmp_seq=585 ttl=64 time=1.214 ms
64 bytes from 10.0.0.3: icmp_seq=586 ttl=64 time=1.238 ms
64 bytes from 10.0.0.3: icmp_seq=587 ttl=64 time=1.291 ms
64 bytes from 10.0.0.3: icmp_seq=588 ttl=64 time=1.449 ms
64 bytes from 10.0.0.3: icmp_seq=589 ttl=64 time=1.414 ms
64 bytes from 10.0.0.3: icmp_seq=590 ttl=64 time=1.314 ms
64 bytes from 10.0.0.3: icmp_seq=591 ttl=64 time=1.441 ms
```

Here the ping command is used between PC3 and PC4

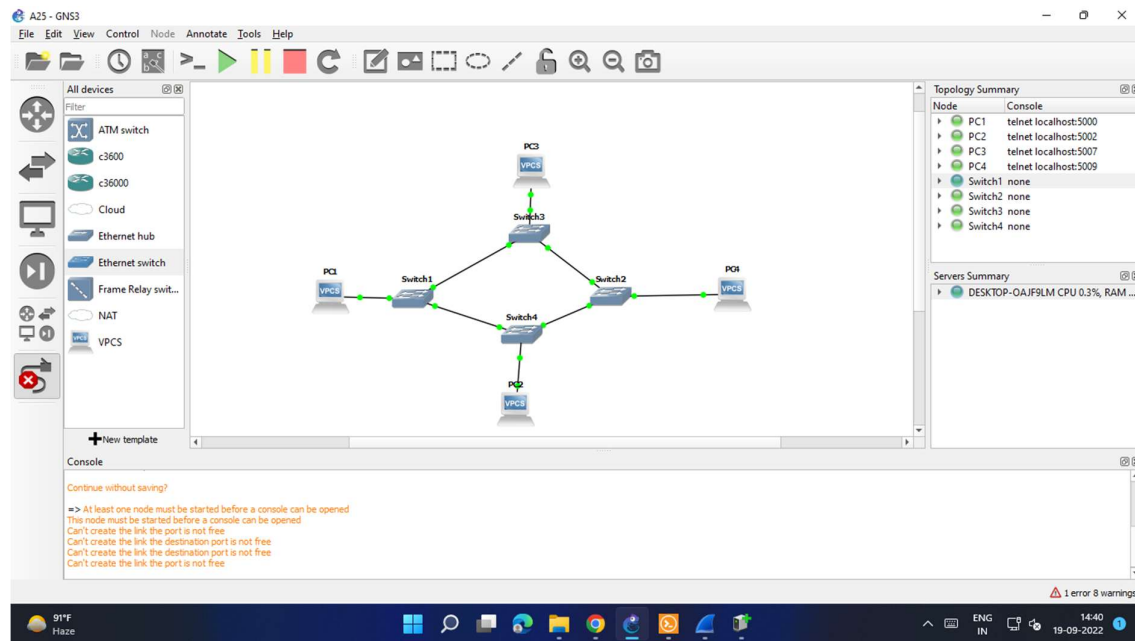
The screenshot shows the Wireshark network traffic capture interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu bar is a toolbar with various icons. The main display area is divided into three panes. The top pane shows a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The bottom pane shows the details of the selected packet (Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0). The bottom pane also shows the raw packet data in hexadecimal and ASCII. The status bar at the bottom indicates 'Ready to load or capture' and 'Packets: 46 · Displayed: 46 (100.0%)'. The system clock shows '14:27 19-09-2022'.

No.	Time	Source	Destination	Protocol	Length	Info
37	18.356348	10.0.0.3	10.0.0.5	ICMP	98	Echo (ping) request id=0xf22e, seq=35/8960, ttl=64 (reply in 38)
38	18.356348	10.0.0.5	10.0.0.3	ICMP	98	Echo (ping) reply id=0xf22e, seq=35/8960, ttl=64 (request in 37)
39	19.378440	10.0.0.3	10.0.0.5	ICMP	98	Echo (ping) request id=0xf32e, seq=36/9216, ttl=64 (reply in 40)
40	19.378440	10.0.0.5	10.0.0.3	ICMP	98	Echo (ping) reply id=0xf32e, seq=36/9216, ttl=64 (request in 39)
41	20.400206	10.0.0.3	10.0.0.5	ICMP	98	Echo (ping) request id=0xf42e, seq=37/9472, ttl=64 (reply in 42)
42	20.400318	10.0.0.5	10.0.0.3	ICMP	98	Echo (ping) reply id=0xf42e, seq=37/9472, ttl=64 (request in 41)
43	21.421938	10.0.0.3	10.0.0.5	ICMP	98	Echo (ping) request id=0xf52e, seq=38/9728, ttl=64 (reply in 44)
44	21.422704	10.0.0.5	10.0.0.3	ICMP	98	Echo (ping) reply id=0xf52e, seq=38/9728, ttl=64 (request in 43)
45	22.442545	10.0.0.3	10.0.0.5	ICMP	98	Echo (ping) request id=0xf62e, seq=39/9984, ttl=64 (reply in 46)
46	22.443277	10.0.0.5	10.0.0.3	ICMP	98	Echo (ping) reply id=0xf62e, seq=39/9984, ttl=64 (request in 45)

> Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0
> Ethernet II, Src: Private_66:68:01 (00:50:79:66:68:01), Dst: Private_66:68:03 (00:50:79:66:68:03)
> Internet Protocol Version 4, Src: 10.0.0.3, Dst: 10.0.0.5
> Internet Control Message Protocol

```
0000  00 50 79 66 68 03 00 50 79 66 68 01 00 00 45 00  :Pyfh..P yfh...E
0010  00 54 2e df 00 00 40 37 c3 0a 00 00 03 0a 00    :T...@ 7.....
0020  00 05 00 00 40 cc df 2e 00 11 00 09 0a 0b 0c 0d  :....@.....
0030  0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d  :.....
0040  1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d  :..."$% &'()*+,-./012345 6789:;<=
0050  2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d  :
0060  3e 3f                                           :>?
```

Again the Wireshark is used to keep a check on requests and replies made



Here Ring Topology has been implemented using Personal Computers and Switches

```

10.0.0.5 icmp_seq=759 timeout
10.0.0.5 icmp_seq=760 timeout
10.0.0.5 icmp_seq=761 timeout
10.0.0.5 icmp_seq=762 timeout
10.0.0.5 icmp_seq=763 timeout
10.0.0.5 icmp_seq=764 timeout
10.0.0.5 icmp_seq=765 timeout
84 bytes from 10.0.0.5 icmp_seq=766 ttl=64 time=1.021 ms
84 bytes from 10.0.0.5 icmp_seq=767 ttl=64 time=1.259 ms
84 bytes from 10.0.0.5 icmp_seq=768 ttl=64 time=1.205 ms
84 bytes from 10.0.0.5 icmp_seq=769 ttl=64 time=1.250 ms
84 bytes from 10.0.0.5 icmp_seq=770 ttl=64 time=1.442 ms
84 bytes from 10.0.0.5 icmp_seq=771 ttl=64 time=1.475 ms
84 bytes from 10.0.0.5 icmp_seq=772 ttl=64 time=1.191 ms
84 bytes from 10.0.0.5 icmp_seq=773 ttl=64 time=1.348 ms
84 bytes from 10.0.0.5 icmp_seq=774 ttl=64 time=1.621 ms
84 bytes from 10.0.0.5 icmp_seq=775 ttl=64 time=1.321 ms
84 bytes from 10.0.0.5 icmp_seq=776 ttl=64 time=1.166 ms
84 bytes from 10.0.0.5 icmp_seq=777 ttl=64 time=1.352 ms
84 bytes from 10.0.0.5 icmp_seq=778 ttl=64 time=1.310 ms
84 bytes from 10.0.0.5 icmp_seq=779 ttl=64 time=1.424 ms
84 bytes from 10.0.0.5 icmp_seq=780 ttl=64 time=1.441 ms
84 bytes from 10.0.0.5 icmp_seq=781 ttl=64 time=1.319 ms
84 bytes from 10.0.0.5 icmp_seq=782 ttl=64 time=1.527 ms
84 bytes from 10.0.0.5 icmp_seq=783 ttl=64 time=1.519 ms
84 bytes from 10.0.0.5 icmp_seq=784 ttl=64 time=1.380 ms
84 bytes from 10.0.0.5 icmp_seq=785 ttl=64 time=1.369 ms
84 bytes from 10.0.0.5 icmp_seq=786 ttl=64 time=1.346 ms
84 bytes from 10.0.0.5 icmp_seq=787 ttl=64 time=1.310 ms
84 bytes from 10.0.0.5 icmp_seq=788 ttl=64 time=1.362 ms
84 bytes from 10.0.0.5 icmp_seq=789 ttl=64 time=1.242 ms
84 bytes from 10.0.0.5 icmp_seq=790 ttl=64 time=1.283 ms
84 bytes from 10.0.0.5 icmp_seq=791 ttl=64 time=1.185 ms
84 bytes from 10.0.0.5 icmp_seq=792 ttl=64 time=1.296 ms
84 bytes from 10.0.0.5 icmp_seq=793 ttl=64 time=1.160 ms
84 bytes from 10.0.0.5 icmp_seq=794 ttl=64 time=1.329 ms
84 bytes from 10.0.0.5 icmp_seq=795 ttl=64 time=1.466 ms
84 bytes from 10.0.0.5 icmp_seq=796 ttl=64 time=1.265 ms
84 bytes from 10.0.0.5 icmp_seq=797 ttl=64 time=1.338 ms
84 bytes from 10.0.0.5 icmp_seq=798 ttl=64 time=1.308 ms

```

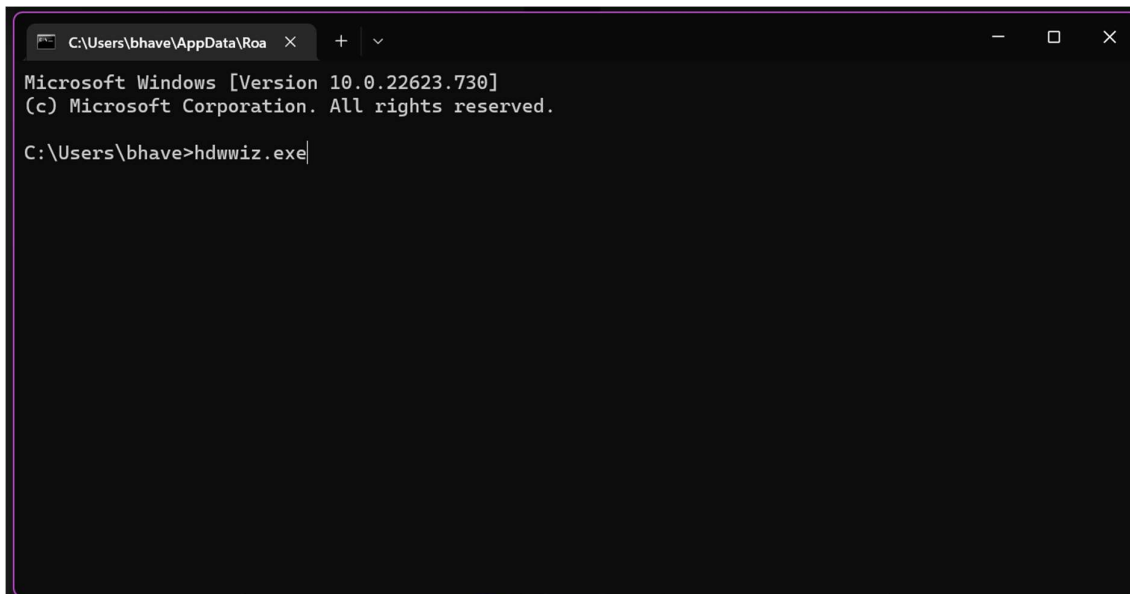
After assigning the IP address the ping command is used to connect two computers

```
queue is full
queue is full
queue is full
10.0.0.3 icmp_seq=555 timeout
64 bytes from 10.0.0.3: icmp_seq=556 ttl=64 time=1.054 ms
64 bytes from 10.0.0.3: icmp_seq=557 ttl=64 time=1.431 ms
64 bytes from 10.0.0.3: icmp_seq=558 ttl=64 time=1.307 ms
64 bytes from 10.0.0.3: icmp_seq=559 ttl=64 time=1.216 ms
64 bytes from 10.0.0.3: icmp_seq=560 ttl=64 time=1.127 ms
64 bytes from 10.0.0.3: icmp_seq=561 ttl=64 time=1.304 ms
64 bytes from 10.0.0.3: icmp_seq=562 ttl=64 time=1.302 ms
64 bytes from 10.0.0.3: icmp_seq=563 ttl=64 time=1.187 ms
64 bytes from 10.0.0.3: icmp_seq=564 ttl=64 time=1.638 ms
64 bytes from 10.0.0.3: icmp_seq=565 ttl=64 time=1.262 ms
64 bytes from 10.0.0.3: icmp_seq=566 ttl=64 time=1.303 ms
64 bytes from 10.0.0.3: icmp_seq=567 ttl=64 time=1.239 ms
64 bytes from 10.0.0.3: icmp_seq=568 ttl=64 time=1.203 ms
64 bytes from 10.0.0.3: icmp_seq=569 ttl=64 time=1.378 ms
64 bytes from 10.0.0.3: icmp_seq=570 ttl=64 time=1.341 ms
64 bytes from 10.0.0.3: icmp_seq=571 ttl=64 time=1.399 ms
64 bytes from 10.0.0.3: icmp_seq=572 ttl=64 time=1.538 ms
64 bytes from 10.0.0.3: icmp_seq=573 ttl=64 time=1.420 ms
64 bytes from 10.0.0.3: icmp_seq=574 ttl=64 time=1.242 ms
64 bytes from 10.0.0.3: icmp_seq=575 ttl=64 time=1.504 ms
64 bytes from 10.0.0.3: icmp_seq=576 ttl=64 time=1.253 ms
64 bytes from 10.0.0.3: icmp_seq=577 ttl=64 time=1.246 ms
64 bytes from 10.0.0.3: icmp_seq=578 ttl=64 time=1.618 ms
64 bytes from 10.0.0.3: icmp_seq=579 ttl=64 time=1.303 ms
64 bytes from 10.0.0.3: icmp_seq=580 ttl=64 time=1.290 ms
64 bytes from 10.0.0.3: icmp_seq=581 ttl=64 time=1.252 ms
64 bytes from 10.0.0.3: icmp_seq=582 ttl=64 time=1.267 ms
64 bytes from 10.0.0.3: icmp_seq=583 ttl=64 time=1.371 ms
64 bytes from 10.0.0.3: icmp_seq=584 ttl=64 time=1.205 ms
64 bytes from 10.0.0.3: icmp_seq=585 ttl=64 time=1.214 ms
64 bytes from 10.0.0.3: icmp_seq=586 ttl=64 time=1.238 ms
64 bytes from 10.0.0.3: icmp_seq=587 ttl=64 time=1.291 ms
64 bytes from 10.0.0.3: icmp_seq=588 ttl=64 time=1.449 ms
64 bytes from 10.0.0.3: icmp_seq=589 ttl=64 time=1.414 ms
64 bytes from 10.0.0.3: icmp_seq=590 ttl=64 time=1.314 ms
64 bytes from 10.0.0.3: icmp_seq=591 ttl=64 time=1.441 ms
```

Here too ping command is used to connect two computers

No.	Time	Source	Destination	Protocol	Length	Info
195	99.066834	10.0.0.3	10.0.0.5	ICMP	98	Echo (ping) request id=0x422f, seq=114/29184, ttl=64 (reply in 196)
196	99.066834	10.0.0.5	10.0.0.3	ICMP	98	Echo (ping) reply id=0x422f, seq=114/29184, ttl=64 (request in 195)
197	100.086771	10.0.0.3	10.0.0.5	ICMP	98	Echo (ping) request id=0x432f, seq=115/29440, ttl=64 (reply in 198)
198	100.087792	10.0.0.5	10.0.0.3	ICMP	98	Echo (ping) reply id=0x432f, seq=115/29440, ttl=64 (request in 197)
199	101.107524	10.0.0.3	10.0.0.5	ICMP	98	Echo (ping) request id=0x442f, seq=116/29696, ttl=64 (reply in 200)
200	101.108526	10.0.0.5	10.0.0.3	ICMP	98	Echo (ping) reply id=0x442f, seq=116/29696, ttl=64 (request in 199)
201	102.132538	10.0.0.3	10.0.0.5	ICMP	98	Echo (ping) request id=0x452f, seq=117/29952, ttl=64 (reply in 202)
202	102.133570	10.0.0.5	10.0.0.3	ICMP	98	Echo (ping) reply id=0x452f, seq=117/29952, ttl=64 (request in 201)
203	103.156327	10.0.0.3	10.0.0.5	ICMP	98	Echo (ping) request id=0x462f, seq=118/30208, ttl=64 (reply in 204)
204	103.157042	10.0.0.5	10.0.0.3	ICMP	98	Echo (ping) reply id=0x462f, seq=118/30208, ttl=64 (request in 203)
205	104.171786	10.0.0.3	10.0.0.5	ICMP	98	Echo (ping) request id=0x472f, seq=119/30464, ttl=64 (reply in 206)
206	104.171786	10.0.0.5	10.0.0.3	ICMP	98	Echo (ping) reply id=0x472f, seq=119/30464, ttl=64 (request in 205)
207	105.108899	10.0.0.3	10.0.0.5	ICMP	98	Echo (ping) request id=0x482f, seq=120/30720, ttl=64 (reply in 208)
208	105.108898	10.0.0.5	10.0.0.3	ICMP	98	Echo (ping) reply id=0x482f, seq=120/30720, ttl=64 (request in 207)
209	106.213369	10.0.0.3	10.0.0.5	ICMP	98	Echo (ping) request id=0x492f, seq=121/30976, ttl=64 (reply in 210)
210	106.214152	10.0.0.5	10.0.0.3	ICMP	98	Echo (ping) reply id=0x492f, seq=121/30976, ttl=64 (request in 209)
211	107.235329	10.0.0.3	10.0.0.5	ICMP	98	Echo (ping) request id=0x4a2f, seq=122/31232, ttl=64 (reply in 212)
212	107.236339	10.0.0.5	10.0.0.3	ICMP	98	Echo (ping) reply id=0x4a2f, seq=122/31232, ttl=64 (request in 211)
213	108.254684	10.0.0.3	10.0.0.5	ICMP	98	Echo (ping) request id=0x4b2f, seq=123/31488, ttl=64 (reply in 214)
214	108.255606	10.0.0.5	10.0.0.3	ICMP	98	Echo (ping) reply id=0x4b2f, seq=123/31488, ttl=64 (request in 213)

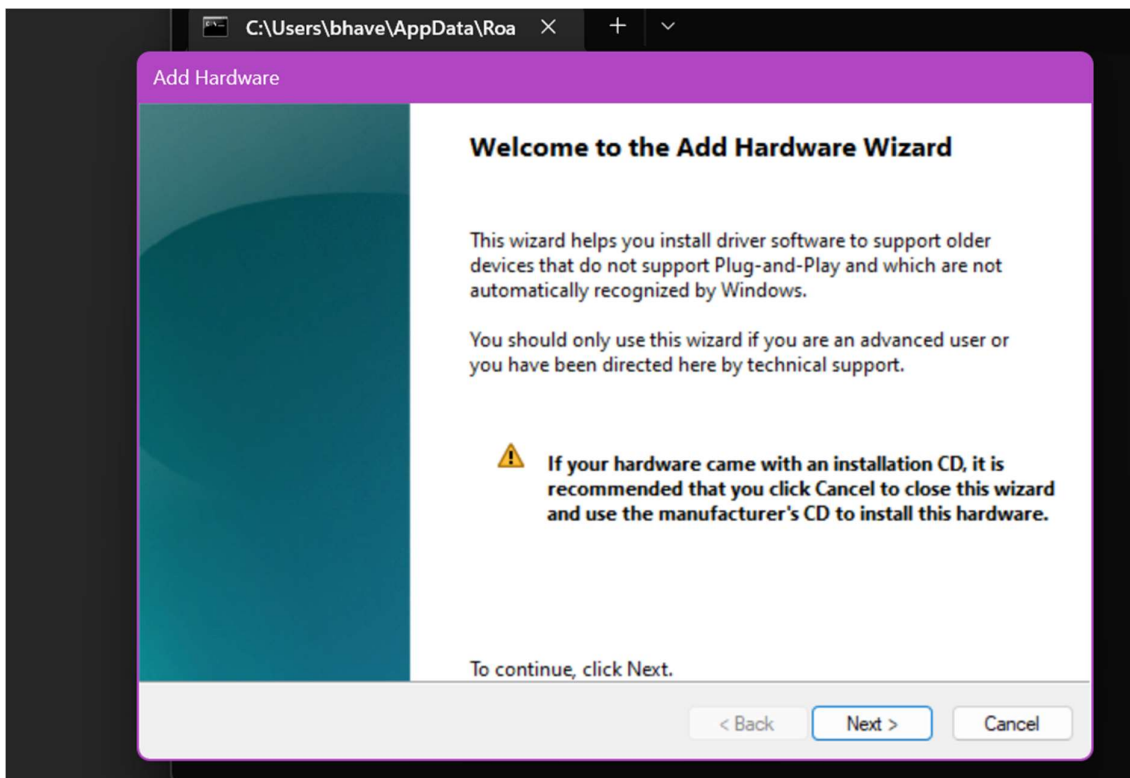
The Wireshark helps us to find check the requests and replies made



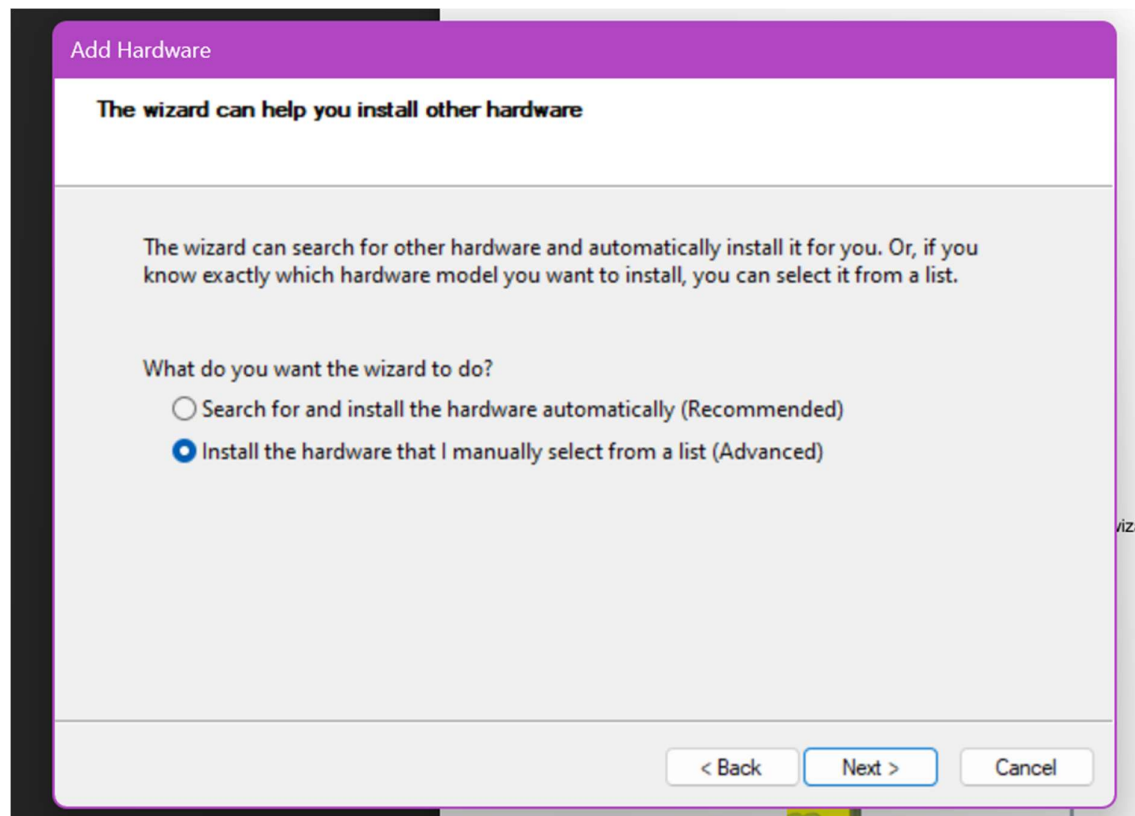
```
C:\Users\bhave\AppData\Roa x + v
Microsoft Windows [Version 10.0.22623.730]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bhave>hddwwiz.exe
```

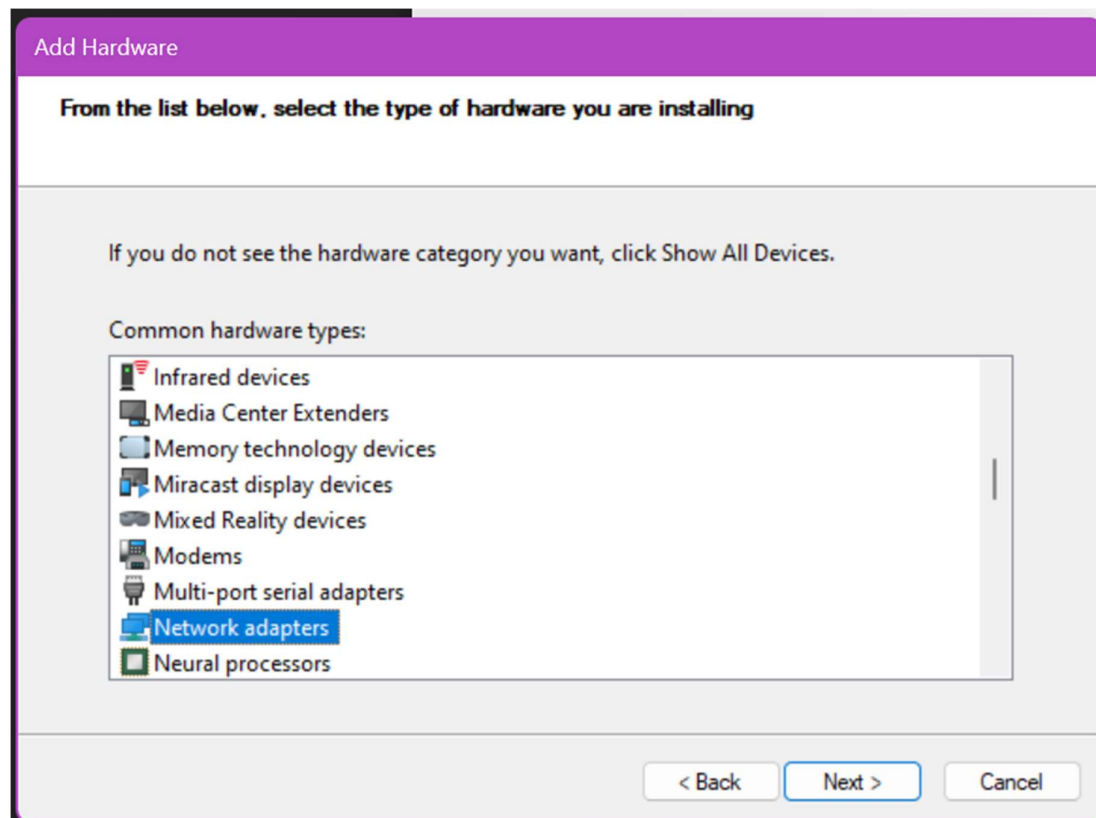
The command is used to access the hardware devices of the personal computers



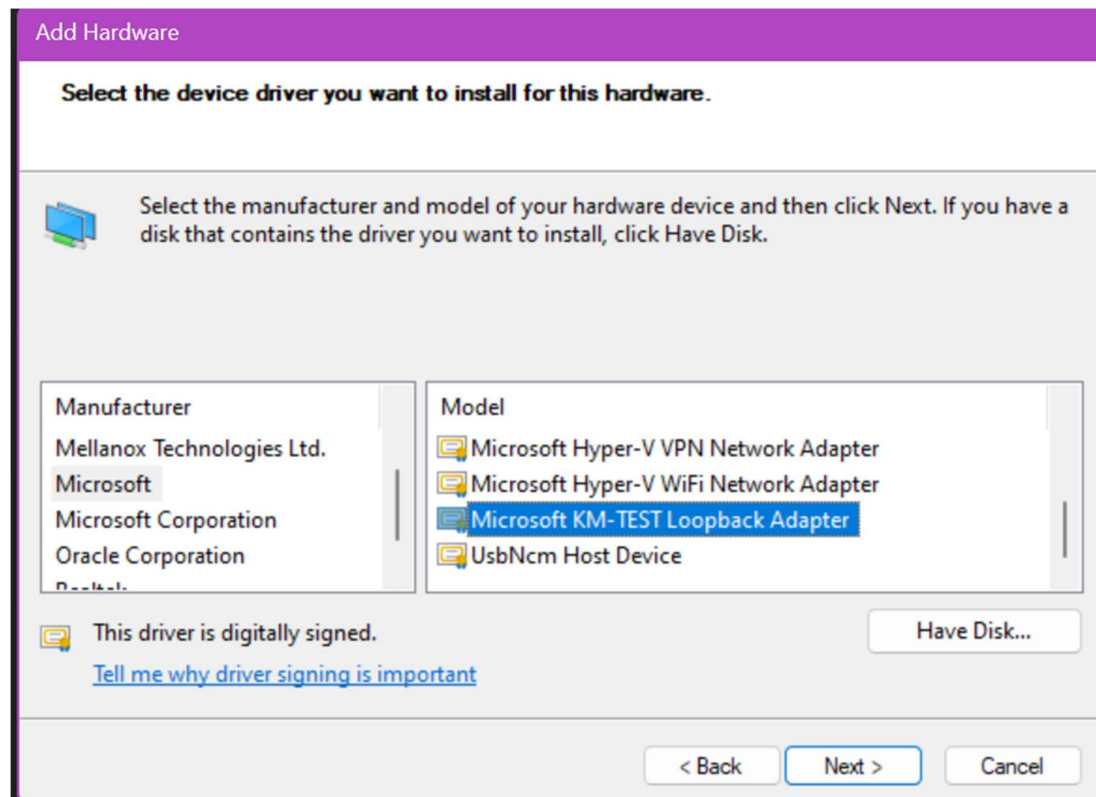
After the command is executed, it opens a hardware wizard



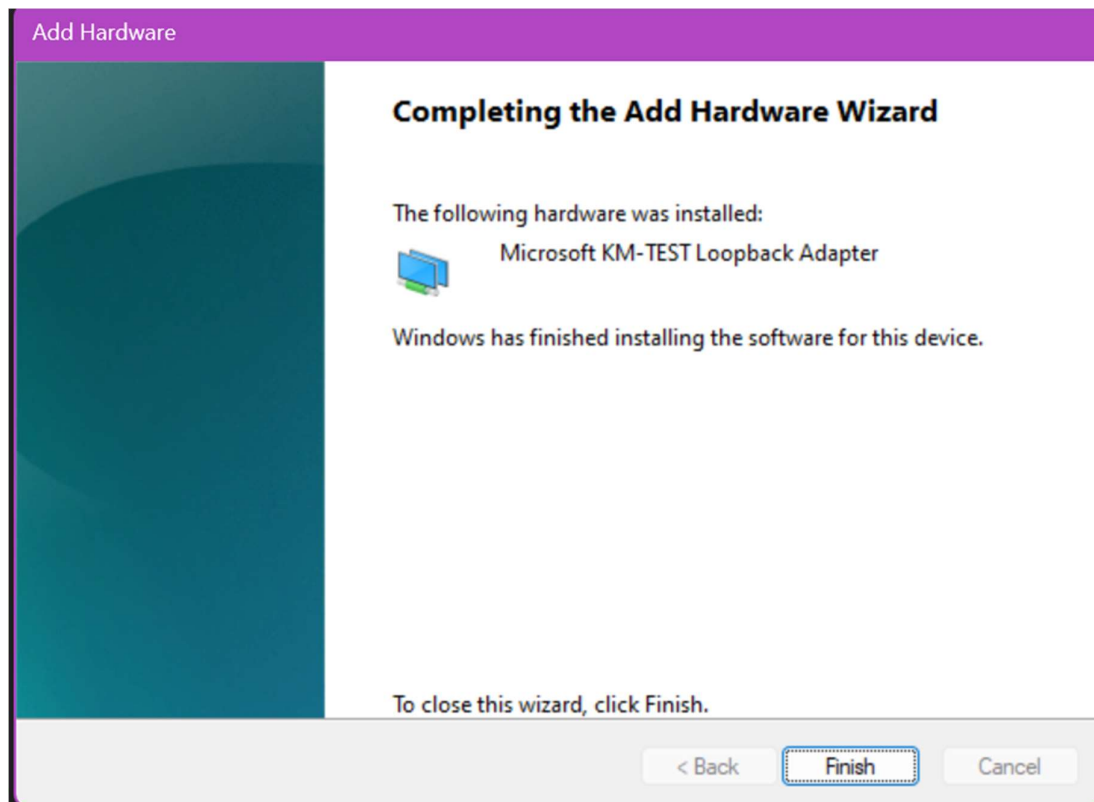
We want to install an external hardware so we choose the option to install it manually and press next



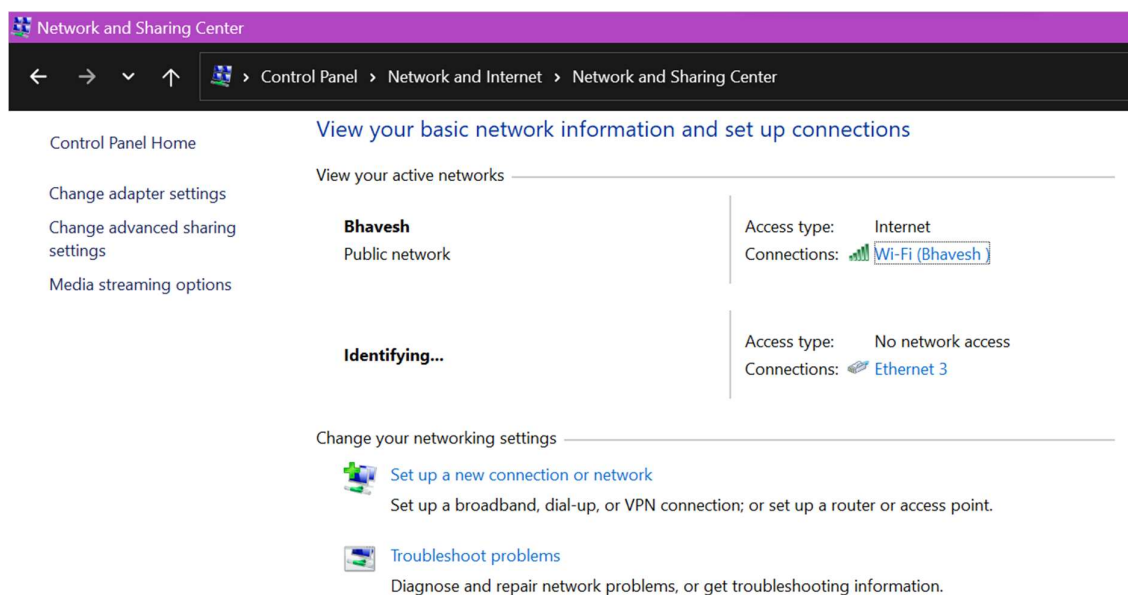
We then select network adapters options and click next



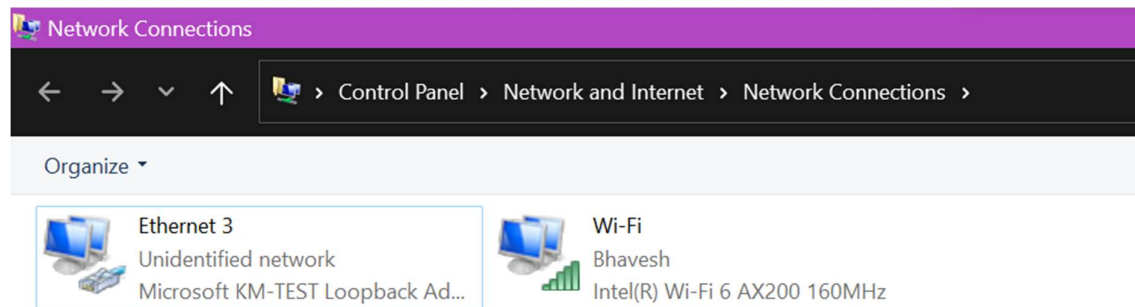
After that we select the manufacturer and then select Microsoft KM Test Loopback Adapter



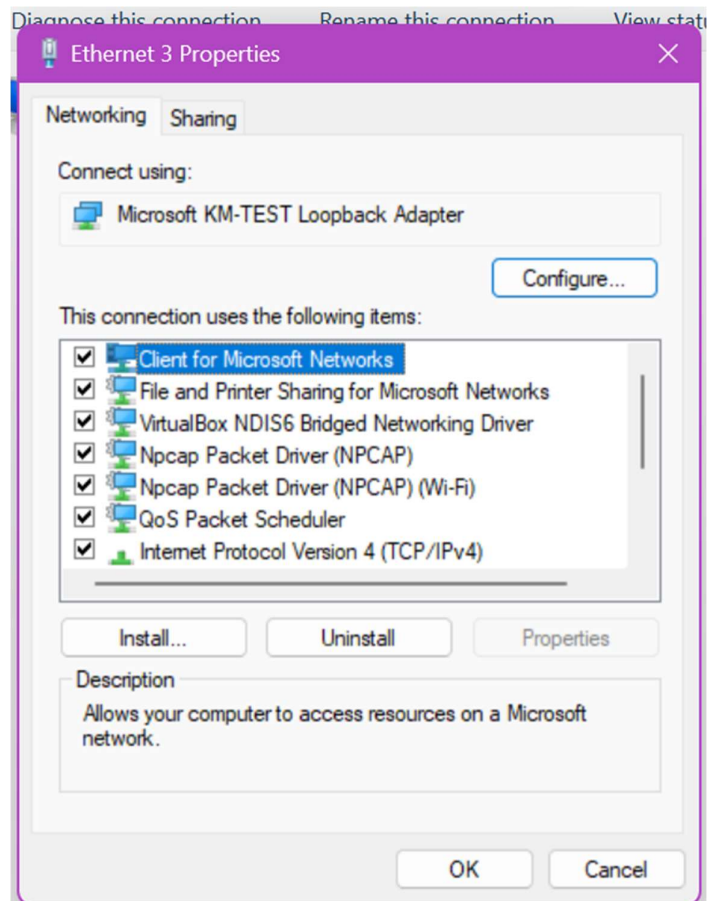
We then finish installing it on our computer



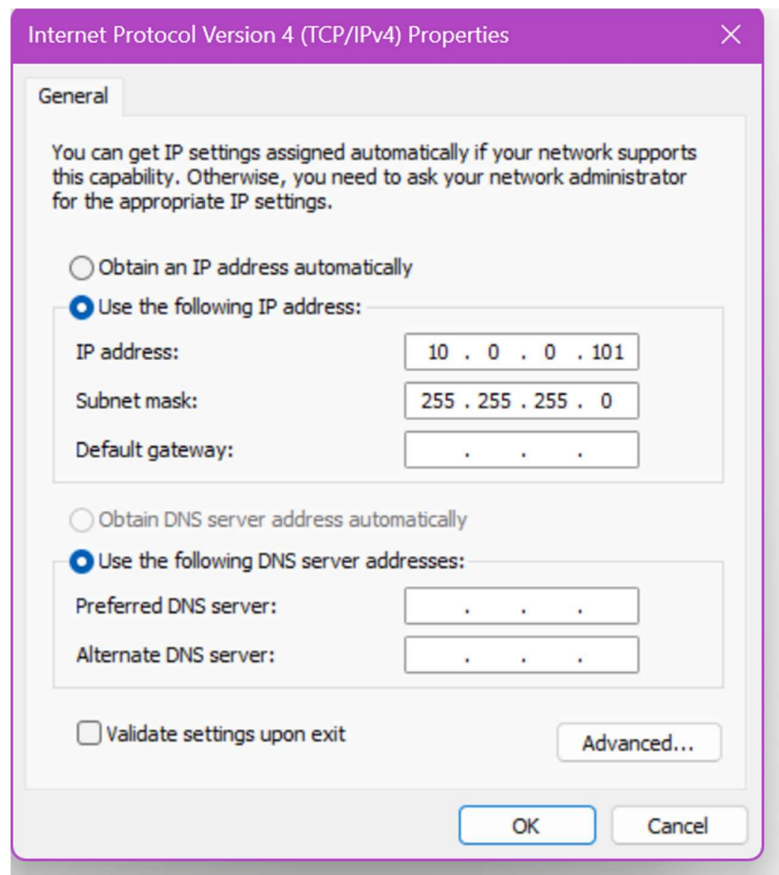
In the control panel, we navigate to Network and Sharing Center. There we can find this adapter



We then assign it an IP address to use it in GSN3



We can see the options provided then find the option of IP address



We then choose that option and then assign the IP address with all other required details

```
C:\Users\bhave\AppData\Roaming>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet 3:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::9d8:4d1d:6ff0:e4c7%54
    IPv4 Address. . . . . : 10.0.0.101
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :
```

Using ipconfig command we can check the connections made by the computer to other networks

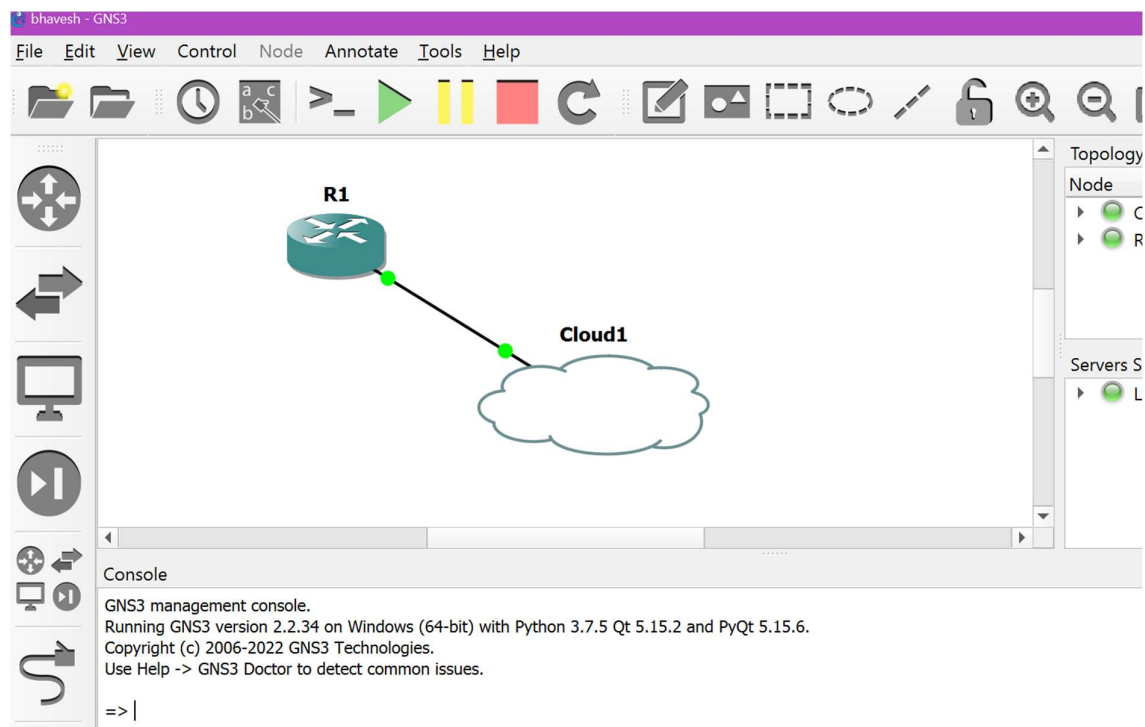
```
C:\Users\bhave>ping 10.0.0.101

Pinging 10.0.0.101 with 32 bytes of data:
Reply from 10.0.0.101: bytes=32 time<1ms TTL=128
Reply from 10.0.0.101: bytes=32 time<1ms TTL=128
Reply from 10.0.0.101: bytes=32 time<1ms TTL=128
Reply from 10.0.0.101: bytes=32 time<1ms TTL=128

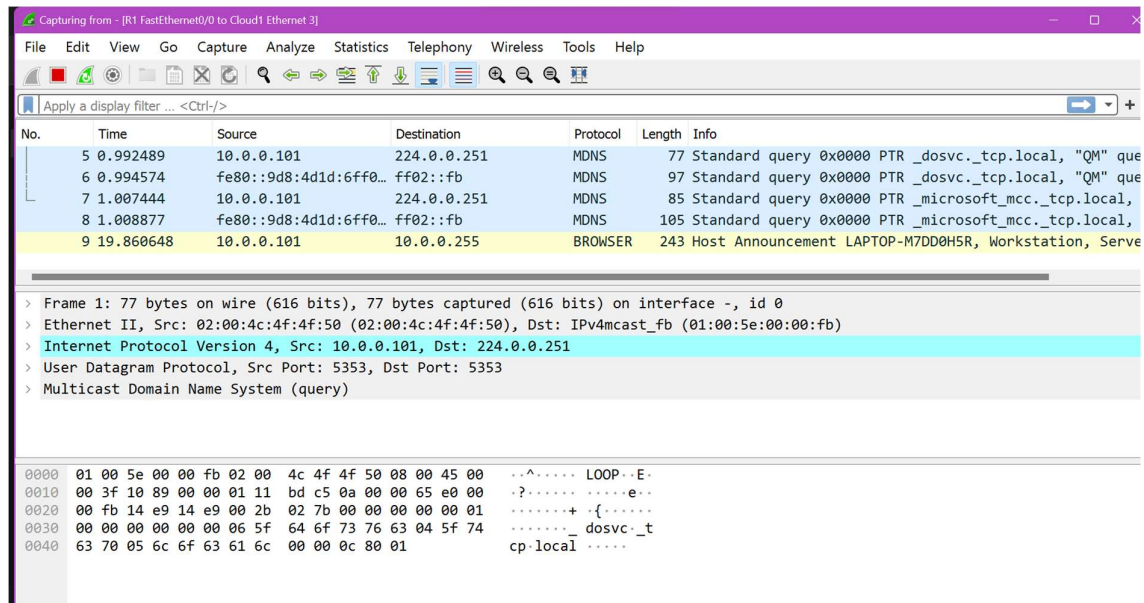
Ping statistics for 10.0.0.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\bhave>|
```

We ping to that network to see if it is reachable or not



In GSN3, we now implement a cloud using that Ethernet with help of Router



Using Wireshark we can check the requests and replies and all other information

Conclusion:

In this practical, we implemented various network topologies like Tree and Ring using the GSN3 Software. We also saw their various applications in real life. We also added our own local PC in GSN3 through Microsoft Loopback Adapter.